

Henning Christiansen Guy De Tré
Adnan Yazici Slawomir Zadrozny
Troels Andreasen
Henrik Legind Larsen (Eds.)

LNAI 7022

Flexible Query Answering Systems

9th International Conference, FQAS 2011
Ghent, Belgium, October 2011
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 7022

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Henning Christiansen Guy De Tré
Adnan Yazici Slawomir Zadrozny
Troels Andreasen Henrik Legind Larsen (Eds.)

Flexible Query Answering Systems

9th International Conference, FQAS 2011
Ghent, Belgium, October 26-28, 2011
Proceedings

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Henning Christiansen
Roskilde University, Denmark, E-mail: henning@ruc.dk

Guy De Tré
Ghent University, Belgium, E-mail: Guy.DeTre@UGent.be

Adnan Yazici
Middle East Technical University (METU), Ankara, Turkey
E-mail: yazici@ceng.metu.edu.tr

Slawomir Zadrozny
Polish Academy of Science, Warsaw, Poland
E-mail: Slawomir.Zadrozny@ibspan.waw.pl

Troels Andreassen
Roskilde University, Denmark, E-mail: troels@ruc.dk

Henrik Legind Larsen
Aalborg University, Esbjerg, Denmark, E-mail: hll@es.aau.dk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-24763-7 e-ISBN 978-3-642-24764-4
DOI 10.1007/978-3-642-24764-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011938238

CR Subject Classification (1998): I.2, H.3, H.2, H.4, H.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at the 9th International Conference on Flexible Query Answering Systems, FQAS 2011, held in Ghent, Belgium, October 26–28, 2011. This biennial conference series has been running since 1994, starting in Roskilde, Denmark, where it was held also in 1996, 1998, and 2009; in 2000 it was held in Warsaw, Poland; in 2002 in Copenhagen, Denmark; in 2004 in Lyon, France; and in 2006 in Milan, Italy.

FQAS has become the premier conference concerned with the very important issue of providing users of information systems with flexible querying capabilities, and with easy and intuitive access to information. More specifically, the overall theme of the FQAS conferences is the modelling and design of innovative and flexible modalities for accessing information systems. The main objective is to achieve more expressive, informative, cooperative, and productive systems that facilitate retrieval from information repositories such as databases, libraries, heterogeneous archives, and the World-Wide Web.

In targeting this objective, the FQAS conferences represent a multidisciplinary approach that draws upon several research areas, including information retrieval, database management, information filtering, knowledge representation, computational linguistics and natural language processing, artificial intelligence, soft computing, classical and non-classical logics, and human computer interaction.

We wish to thank all authors who contributed with their excellent papers, as well as our Program Committee and additional reviewers for their efforts. We thank also our invited speakers Janusz Kacprzyk, Jozo Dujmovic, Gabriella Pasi, and Michael Brands, and the organizers of special sessions, Bich-Lien Doan, Juan Miguel Medina, and Carlos D. Barranco. Finally, we extend our gratitude to the members of the Advisory Board, several of whom have supported the FQAS conferences from the very beginning, and also to the Local Organizing Committee without whom FQAS 2011 would not have been possible.

August 2011

Henning Christiansen
Guy De Tré
Adnan Yazıcı
Sławomir Zadrozny
Troels Andreasen
Henrik Legind Larsen

Organization

FQAS 2011 was co-organized by the Department of Communication, Business and Information Technologies at Roskilde University, Denmark; the Department of Telecommunications and Information Processing at Ghent University, Belgium; the Systems Research Institute, Polish Academy of Sciences, Poland; and the Department of Computer Engineering of Middle East Technical University, Turkey.

General and Program Co-chairs

Henning Christiansen	Roskilde University, Denmark
Guy De Tré	Ghent University, Belgium
Adnan Yazıcı	Middle East Technical University, Turkey
Sławomir Zadrozny	Systems Research Institute, Polish Academy of Sciences, Poland

Local Organizing Committee

Christophe Billiet	Ghent University, Belgium
Antoon Bronselaer	Ghent University, Belgium
Bernard De Baets	Ghent University, Belgium
Guy De Tré	Ghent University, Belgium (Chair)
Gert De Cooman	Ghent University, Belgium
Tom Matthé	Ghent University, Belgium
Joachim Nielandt	Ghent University, Belgium
Daan Van Britsom	Ghent University, Belgium
Jef Wijzen	Université de Mons, Belgium

Steering Committee

Troels Andreassen	Roskilde University, Denmark
Henning Christiansen	Roskilde University, Denmark
Henrik Legind Larsen	Aalborg University, Denmark

International Advisory Board

Patrick Bosc	Norbert Fuhr
Jesús Cardeñosa	Christian S. Jensen
Guy De Tré	Janusz Kacprzyk
Jørgen Fischer Nilsson	Amihai Motro

Gabriella Pasi
Fred Petry
Olivier Pivert
Henri Prade

Zbigniew W. Ras
Ronald R. Yager
Adnan Yazıcı
Sławomir Zadrozny

Program Committee

Troels Andreasen, Denmark
Carlos D. Barranco, Spain
María José Martín Bautista, Spain
Leo Bertossi, Canada
Gloria Bordogna, Italy
Alexander Borgida, USA
Pia Borlund, Denmark
Patrick Bosc, France
Mohand Boughanem, France
Patrice Buche, France
Henrik Bulskov, Denmark
Jesús Cardeñosa, Spain
Paola Carrara, Italy
Panagiotis Chountas, UK
Henning Christiansen, Denmark
Bruce Croft, USA
Juan Carlos Cubero, Spain
Alfredo Cuzzocrea, Italy
Ernesto Damiani, Italy
Agnieszka Dardzinska, Poland
Bernard De Baets, Belgium
Guy De Tré, Belgium
Bich-Lien Doan, France
Peter Dolog, Denmark
Didier Dubois, France
Jørgen Fischer Nilsson, Denmark
Alexander Gelbukh, Mexico
Allel Hadjali, France
Sven Helmer, UK
Enrique Herrera-Viedma, Spain
Eyke Hüllermeier, Germany
Gareth Jones, UK
Janusz Kacprzyk, Poland
Etienne Kerre, Belgium
Werner Kiessling, Germany

Don Kraft, USA
Anne Laurent, France
Henrik Legind Larsen, Denmark
Marie-Jeanne Lesot, France
Marianne Lykke Nielsen, Denmark
Christophe Marsala, France
Davide Martinenghi, Italy
Andrea Maurino, Italy
Juan Miguel Medina, Spain
Jose Olivas, Spain
Daniel Ortiz-Arroyo, Denmark
Stefano Paraboschi, Italy
Gabriella Pasi, Italy
Fred Petry, USA
Olivier Pivert, France
Olga Pons, Spain
Henri Prade, France
Giuseppe Psaila, Italy
Zbigniew W. Ras, USA
Guillaume Raschia, France
Maria Rifqi, France
Andrzej Skowron, Poland
Nicolas Sproyat, France
Heiner Stuckenschmidt, Germany
Lynda Tamine-Lechani, France
Letizia Tanca, Italy
Vicenc Torra, Spain
Farouk Toumani, France
Maria-Amparo Vila, Spain
Peter Vojtas, Slovakia
Jef Wijsen, Belgium
Ronald R. Yager, USA
Adnan Yazıcı, Turkey
Sławomir Zadrozny, Poland

External Reviewers

Rafal Angrik

Ignacio Blanco Medina

Ourdia Boudighaghen

Antoon Bronselaer

Silvia Calegari

Tolga Can

Jesús Roque Campaña

Michel De Rougemont

Markus Endres

Gilles Falquet

Jose Galindo

Maria Grineva

Nicolas Hernandez

Joemon Jose

Olga Kolesnikova

Dilek Küçük

Ivo Lasek

Yulia Ledeneva

Yannick Loiseau

Stefan Mandl

Nicolás Marín

Tom Matthé

Viet Phan-Luong

Mathieu Roche

Fatiha Sais

Jean-Paul Sansonnet

Pinar Senkul

Daan Van Britsom

Sławomir Wierzchoń

Sponsoring Institutions

Faculty of Engineering, Ghent University, Belgium

Research Foundation – Flanders, Belgium

InterSystems Benelux, Belgium

Table of Contents

Logical Approaches to Flexible Querying

Generalizing Conjunctive Queries for Informative Answers	1
<i>Katsumi Inoue and Lena Wiese</i>	
ReAction: Personalized Minimal Repair Adaptations for Customer Requests	13
<i>Monika Schubert, Alexander Felfernig, and Florian Reinfrank</i>	
Uncertainty That Counts	25
<i>Dany Maslowski and Jef Wijsen</i>	
Structured Data-Based Q&A System Using Surface Patterns	37
<i>Nicolas Kuchmann-Beauger and Marie-Aude Aufaure</i>	

Fuzzy Logic in Spatial and Temporal Data Modeling and Querying

Higher Reasoning with Level-2 Fuzzy Regions	49
<i>Jörg Verstraete</i>	
Bipolar Fuzzy Querying of Temporal Databases	60
<i>Christophe Billiet, Jose Enrique Pons, Tom Matthé, Guy De Tré, and Olga Pons Capote</i>	
Implementation of X-Tree with 3D Spatial Index and Fuzzy Secondary Index	72
<i>Sinan Keskin, Adnan Yazıcı, and Halit Oğuztüzün</i>	

Knowledge-Based Approaches

Semantic Processing of Database Textual Attributes Using Wikipedia	84
<i>Jesús R. Campaña, Juan M. Medina, and M. Amparo Vila</i>	
Querying Class-Relationship Logic in a Metalogic Framework	96
<i>Jørgen Fischer Nilsson</i>	
A Semantics-Based Approach to Retrieving Biomedical Information	108
<i>Troels Andreasen, Henrik Bulskov, Sine Zambach, Tine Lassen, Bodil Nistrup Madsen, Per Anker Jensen, Hanne Erdman Thomsen, and Jørgen Fischer Nilsson</i>	

Knowledge Extraction for Question Titling	119
<i>Carolina Gallardo Pérez and Jesús Cardeñosa</i>	

Multimedia

Multilingual Video Indexing and Retrieval Employing an Information Extraction Tool for Turkish News Texts: A Case Study	128
<i>Dilek Küçük and Adnan Yazıcı</i>	

A Search-Engine Concept Based on Multi-feature Vectors and Spatial Relationship	137
<i>Tatiana Jaworska</i>	

Exploiting Class-Specific Features in Multi-feature Dissimilarity Space for Efficient Querying of Images	149
<i>Turgay Yilmaz, Adnan Yazıcı, and Yakup Yildirim</i>	

Data Fuzziness, Reliability and Trust

Generalised Fuzzy Types and Querying: Implementation within the Hibernate Framework	162
<i>Jose Enrique Pons, Ignacio Blanco Medina, and Olga Pons Capote</i>	

Data Reliability Assessment in a Data Warehouse Opened on the Web	174
<i>Sébastien Destercke, Patrice Buche, and Brigitte Charnomordic</i>	

Propagation of Question Waves by Means of Trust in a Social Network	186
<i>Albert Trias Mansilla and Josep Lluís de la Rosa Esteva</i>	

Information Retrieval

Investigating the Statistical Properties of User-Generated Documents . . .	198
<i>Giacomo Inches, Mark James Carman, and Fabio Crestani</i>	

Information Retrieval from Turkish Radiology Reports without Medical Knowledge	210
<i>Kerem Hadımlı and Meltem Turhan Yöndem</i>	

Discovering and Analyzing Multi-granular Web Search Results	221
<i>Gloria Bordogna and Giuseppe Psaila</i>	

Semantically Oriented Sentiment Mining in Location-Based Social Network Spaces	234
<i>Domenico Carlone and Daniel Ortiz-Arroyo</i>	

Preference Queries

Skyline Snippets	246
<i>Markus Endres and Werner Kießling</i>	
Evaluating Top-k Algorithms with Various Sources of Data and User Preferences	258
<i>Alan Eckhardt, Erik Horničák, and Peter Vojtáš</i>	
Efficient and Effective Query Answering for Trajectory Cuboids	270
<i>Elio Masciari</i>	
A Model for Analyzing and Visualizing Tabular Data	282
<i>Ekaterina Simonenko, Nicolas Spyrtatos, and Tsuyoshi Sugibuchi</i>	

Flexible Querying of Graph Data

An Analysis of an Efficient Data Structure for Evaluating Flexible Constraints on XML Documents	294
<i>Stefania Marrara, Emanuele Panzeri, and Gabriella Pasi</i>	
Intuitionistic Fuzzy XML Query Matching	306
<i>Mohammedsharaf Alzebdi, Panagiotis Chountas, and Krassimir Atanassov</i>	
A Cooperative Answering Approach to Fuzzy Preferences Queries in Service Discovery	318
<i>Katia Abbaci, Fernando Lemos, Allel Hadjali, Daniela Grigori, Ludovic Liétard, Daniel Rocacher, and Mokrane Bouzeghoub</i>	

Ranking, Ordering and Statistics

Fuzzy Orderings for Fuzzy Gradual Patterns	330
<i>Malaquias Quintero, Anne Laurent, and Pascal Poncelet</i>	
Spearman's Rank Correlation Coefficient for Vague Preferences	342
<i>Przemysław Grzegorzewski and Paulina Ziemińska</i>	
Premium Based on Mixture Utility	354
<i>Jana Špírková</i>	

Query Recommendation and Interpretation

Which Should We Try First? Ranking Information Resources through Query Classification	364
<i>Joshua Church and Amihai Motro</i>	

Context Modelling for Situation-Sensitive Recommendations	376
<i>Stewart Whiting and Joemon Jose</i>	
Folksonomy Query Suggestion via Users' Search Intent Prediction	388
<i>Chiraz Trabelsi, Bilel Moulahi, and Sadok Ben Yahia</i>	
Factorizing Three-Way Ordinal Data Using Triadic Formal Concepts . . .	400
<i>Radim Belohlavek, Petr Osička, and Vilem Vychodil</i>	

Fuzzy Databases and Applications

On Possibilistic Skyline Queries	412
<i>Patrick Bosc, Allel Hadjali, and Olivier Pivert</i>	
A Fuzzy Valid-Time Model for Relational Databases Within the Hibernate Framework	424
<i>Jose Enrique Pons, Olga Pons Capote, and Ignacio Blanco Medina</i>	
On the Use of a Fuzzy Object-Relational Database for Retrieval of X-rays on the Basis of Spine Curvature Pattern Similarities	436
<i>Sergio Jaime-Castillo, Juan M. Medina, Carlos D. Barranco, and Antonio Garrido</i>	
A Fuzzy-Rule-Based Approach to the Handling of Inferred Fuzzy Predicates in Database Queries	448
<i>Allel Hadjali and Olivier Pivert</i>	
Flexible Content Extraction and Querying for Videos	460
<i>Utku Demir, Murat Koyuncu, Adnan Yazici, Turgay Yilmaz, and Mustafa Sert</i>	
Bipolar SQLf: A Flexible Querying Language for Relational Databases	472
<i>Nouredine Tamani, Ludovic Liétard, and Daniel Rocacher</i>	
On the Behavior of Indexes for Imprecise Numerical Data and Necessity Measured Queries under Skewed Data Sets	485
<i>Carlos D. Barranco, Jesús R. Campaña, and Juan M. Medina</i>	
Fuzzy Domains with Adaptable Semantics in an Object-Relational DBMS	497
<i>José Tomás Cadenas, Nicolás Marín, and M. Amparo Vila</i>	
Author Index	509

Generalizing Conjunctive Queries for Informative Answers

Katsumi Inoue and Lena Wiese*

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{ki,wiese}@nii.ac.jp

Abstract. Deductive generalization of queries is one method to provide informative answers to failing queries. We analyze properties of operators that generalize conjunctive queries consisting of positive as well as negative literals. We show that for the stepwise combination of these operators it suffices to apply the operator in one certain order.

1 Introduction and Related Work

Retrieval of data stored in database systems is a basic use case in the information society. Hence answering a user’s queries is a central issue for database systems (apart from other aspects like for example integrity, availability and efficiency of the system). However, a database system may not always be able to answer queries in a satisfactory manner. In particular, if a database answer is empty the corresponding query is said to be a “failing query” (see for example [15]). The reasons for this can be manifold; for instance, in a selection query, selection conditions may be too strict to yield any result. Some authors (for example [5]) differentiate between “misconceptions” and “false presuppositions” as causes for failure. **Cooperative database systems** search for answers that – although not exactly matching the user’s original query – are **informative answers** for the user: they provide data that are “closely related” to the user’s intention; or they fix misconceptions in a query and return answers to the modified query. Current search engines, web shops or expert systems use similar cooperative techniques to provide users with data that might be close to their interest.

In this article we want to foster the logical foundation of cooperative query answering. We analyze a set of generalization operators that can be applied to failing queries. This logical point of view has some history of related work which we will survey briefly. The term “cooperative database system” was for example used in [2] for a system called “CoBase” that relies on several type abstraction hierarchies (TAH) to relax queries and hence to return a wider range of answers. In a similar manner, Halder and Cortesi [8] employ abstraction of domains and define optimality of answers with respect to some user-defined relevancy constraints. The approach by Pivert et al using fuzzy sets [15] analyzes cooperative

* Lena Wiese gratefully acknowledges a postdoctoral research grant of the German Academic Exchange Service (DAAD).

query answering based on semantic proximity. With what they call “incremental relaxation” they apply generalization operators to single conjuncts in a conjunctive query; hence generalized queries form a lattice structure: queries with the same number of applications generalization operators are in the same level of the lattice. Other related systems are Flex [13], Carmin [6] and Ishmael [5] that introduce and analyze dedicated generalization operators. Hurtado et al [9] relax RDF queries based on an ontology. In a multi-agent negotiation framework, Sakama and Inoue [17] devise procedures for generating “neighborhood proposals” in a negotiation process using the three operators we will also analyze in this paper.

We complement and advance these previous works by

- analyzing the properties of the three generalization operators “dropping conditions”, “anti-instantiation”, and “goal replacement” for deductive generalization of queries.
- combining these operators in a breadth-first search manner while employing a notion of minimality based on the number of applications of operators.

The paper is outlined as follows: Sections 2 and 3 set the basic terminology for cooperative query answering and query generalization. Section 4 introduces three basic generalization operators for conjunctive queries and Section 5 analyzes the combination of these. Section 6 concludes the paper with a brief discussion.

2 Cooperative Query Answering for Knowledge Bases

In this article we will follow a formal approach to achieve informative answers in a cooperative knowledge base system. In particular, we will base query answering on a consequence operator (denoted \models) in a chosen logic. Throughout this article we assume a logical language \mathcal{L} consisting of a finite set of predicate symbols (for example denoted *Ill*, *Treat* or *P*), a possibly infinite set *dom* of constant symbols (for example denoted *Mary* or *a*), and an infinite set of variables (for example denoted *x* or *y*). The capital letter *X* denotes a vector of variables; if the order of variables in *X* does not matter, we identify *X* with the set of its variables and apply set operators – for example we write $y \in X$. We use the standard logical connectors conjunction \wedge , disjunction \vee , negation \neg and material implication \rightarrow and universal \forall as well as existential \exists quantifiers. An atom is a formula consisting of a single predicate symbol only; a literal is an atom (a “positive literal”) or a negation of an atom (a “negative literal”); a clause is a disjunction of atoms; a ground formula is one that contains no variables; the existential (universal) closure of a formula ϕ is written as $\exists\phi$ ($\forall\phi$) and denotes the closed formula obtained by binding all free variables of ϕ with the respective quantifier. In particular, we will later on use single-headed “range-restricted rules” of the form $L_{i_1} \wedge \dots \wedge L_{i_m} \rightarrow L'$ where each L_{i_j} and L' are literals and all free variables of L' are contained in the free variables of the L_{i_j} . On occasion, if a set is a singleton set we will identify the set with the single formula in it. For a set of formulas, a model is an interpretation that makes all formulas in the set *true*.

For two sets of formulas S and S' , logical implication $S \models S'$ means that every model of S is also a model of S' . Two formulas ϕ , ϕ' are equivalent (denoted $\phi \equiv \phi'$) if $\phi \models \phi'$ and $\phi' \models \phi$. A particular case of equivalent formulas are “variants” that are syntactically identical up to variable renaming.

In this article, we assume that data are stored in a **knowledge base**. A knowledge base (denoted Σ) is a set of formulas in the chosen logic; if formulas in Σ contain free variables, these variables are assumed to be universally quantified; in other words, the formulas in Σ are identified with their universal closure. A knowledge base represents a set of “possible worlds”; that is, a set of models of the knowledge base. As an example, consider an information system with medical data on patient’s illnesses and medical treatments. For the knowledge base $\Sigma = \{Ill(\text{Mary}, \text{Cough}) \vee Ill(\text{Mary}, \text{Flu})\}$, a possible world would be $\{Ill(\text{Mary}, \text{Cough})\}$ making only this single ground atom *true* and all other ground atoms *false*. Note that with an infinite underlying domain *dom* there are in general infinitely many worlds (and also infinitely many possible worlds) for a knowledge base. In addition, we implicitly assume that a knowledge base is consistent: it has at least one possible world. In an inconsistent database (that is, one containing a logical contradiction like $\Sigma = \{P(a) \wedge \neg P(a)\}$), any formula can be derived which makes query answering useless. Knowledge bases serve as a quite generic data model for the representation of knowledge in multiagent systems, belief revision or ontology-based reasoning. Note that when restricting a knowledge base to a set of ground atoms and accompanying it with a “closed world assumption” (that makes all ground atoms not contained in the knowledge base *false*), then the concept of a knowledge base can emulate the relational data model; our analysis will however apply to the general case also without this assumption.

A user can issue queries to a knowledge base to retrieve data from it. In the following we analyze operators that can generalize conjunctive queries with positive as well as negative literals.

Definition 1 (Query). *A query is a conjunctive formula $L_1 \wedge \dots \wedge L_n$ where each L_i is a literal. We often abbreviate a query as $Q(X)$, where Q stands for the conjunction of literals and X is an n -tuple of variables appearing in Q .*

For query answering different semantics can be employed. For example, an open query $Q(X)$ can be answered by finding all substitutions for the free variables of $Q(X)$; more formally, for a substitution θ that maps the free variables of $Q(X)$ to arbitrary terms (including variables), $\forall Q(X)\theta$ is a correct answer if $\Sigma \models \forall Q(X)\theta$. [11] analyze minimal disjunctive answers whereas [17] use answer set semantics for extended disjunctive logic programs. In this article, we apply generalization operators to queries. This can be done independent of a specific query answering semantics. We just assume that a dedicated query answering function *ans* represents the set of all correct answers.

Definition 2 (Answer set). *For a query $Q(X)$ and a knowledge base Σ , the set of correct answers (or answer set, for short) $ans(Q(X), \Sigma)$ is a set of closed formulas such that for each $\phi \in ans(Q(X), \Sigma)$ it holds that $\Sigma \models \phi$ and ϕ is derived from $Q(X)$ by some query answering semantics.*

For a given knowledge base, not all queries can be answered correctly. If no correct answer for a query exists, the query is said to fail.

Definition 3 (Failing query). *Let Σ be a knowledge base, $Q(X)$ be a query. If $\text{ans}(Q(X), \Sigma) = \emptyset$, the query $Q(X)$ fails (in Σ).*

3 Query Generalization

Our purpose is to devise strategies for **cooperative query answering with generalization**: if for some query $Q(X)$ the knowledge base answer is the empty set, $Q(X)$ is transformed into a more general query $Q^{gen}(X, Y)$ that has a non-empty answer in the knowledge base. This idea is visualized in Figure 1. Properties of inductive and deductive generalization have already been discussed by de Raedt [3]. He argues that specialization and generalization are inverse operators and can be applied for both inductive and deductive reasoning. He also shows that an inductive operator applied to the negation of a formula behaves as a deductive operator (when again the result of the induction is negated). For query generalization we will apply deduction. Generalization can take place either in a way *independent* of knowledge base contents: it acts solely on the query formula $Q(X)$ like the generalization operator “dropping conditions”. On the other hand, generalization can *depend* on the knowledge base by applying rules contained in the knowledge base: with the operator of “goal replacement” the query $Q(X)$ is modified by substituting a part of the query according to some rule in the knowledge base. In this sense, we will speak of generalization “with respect to a knowledge base”. This notion has been used for “relaxation” in [4]. As it can be applied to open formulas we borrow it here; however, we combine operators in a different manner and employ an operator-based distance.

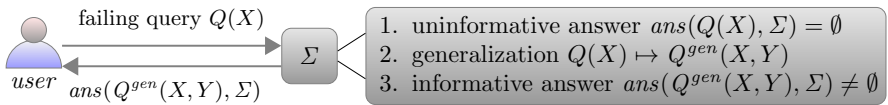


Fig. 1. Query generalization for informative answers

Definition 4 (Deductive generalization wrt. knowledge base [4]). *Let Σ be a knowledge base, $\phi(X)$ be a formula with a tuple X of free variables, and $\psi(X, Y)$ be a formula with an additional tuple Y of free variables disjoint from X . The formula $\psi(X, Y)$ is a deductive generalization of $\phi(X)$, if it holds in Σ that the less general ϕ implies the more general ψ where for the free variables X (the ones that occur in ϕ and possibly in ψ) the universal closure and for free variables Y (the ones that occur in ψ only) the existential closure is taken:*

$$\Sigma \models \forall X \exists Y (\phi(X) \rightarrow \psi(X, Y))$$

As a simple example, assuming classical first-order logic, for the formula $\phi_1 = Ill(\text{Mary}, \text{Cough})$ the formula $Ill(y, \text{Cough})$ is a deductive generalization because it holds in any first-order model that $\exists y(Ill(\text{Mary}, \text{Cough}) \rightarrow Ill(y, \text{Cough}))$. Analogously, $Ill(\text{Mary}, y')$ and $Ill(y, y')$ are generalizations for the same formula ϕ_1 . But note that $Ill(y, y')$ is also a generalization of $Ill(y, \text{Cough})$ because it holds in any first-order model that $\forall y \exists y' (Ill(y, \text{Cough}) \rightarrow Ill(y, y'))$. Whereas for $Ill(y, \text{Cough})$ and $Ill(\text{Mary}, y')$ neither is a generalization of the other.

We can now give a formal definition of generalized queries.

Definition 5 (Generalized query with informative answer). *A query formula $Q^{gen}(X, Y)$ is a generalized query with informative answer for a query $Q(X)$ (with respect to a knowledge base Σ) if the following properties hold:*

1. **Failing query:** $ans(Q(X), \Sigma) = \emptyset$
2. **Generalized query:** $Q^{gen}(X, Y)$ is a deductive generalization of $Q(X)$
3. **Informative answer:** $ans(Q^{gen}(X, Y), \Sigma) \neq \emptyset$

To capture a notion of closeness between the original query and the generalized query, the generalized query should have some property of **minimal** generalization. There are several definitions of minimality available. For example, Plotkin [16] devised subsumption-based “least general generalization”. For generalization based on an abstraction hierarchy, minimal distance can be defined by a shortest path in the hierarchy [18]. Another established method is a model-based distance as for example used for dilation operators [7]. The number of applications of generalization operators is counted in [1]. We also employ a notion of minimality that counts the number of individual generalization steps based on a set $GenOp$ of generalization operators. In contrast to [1], the operators we analyze (in particular “goal replacement”) need not apply only to single conjuncts in a query but can also apply to several conjuncts at the same time; thus our operators lead to a generalization behavior different from [1].

Definition 6 (Generalization operator). *For sets S and S' of formulas, an operator o is a generalization operator if $o(S) = S'$ and for each $\psi \in S'$ there is a $\phi \in S$ such that ψ is a deductive generalization of ϕ .*

Note that o might not be applicable to all formulas in S and hence the mapping is only surjective. Such a generalization operator defines one single atomic generalization step – for example, as we will use later, dropping one condition, or replacing one goal, or anti-instantiating one constant in a query. We now assume that a set $GenOp$ has been specified and define the operator-based distance; however we defer the description of generalization operators until Section 4.

Definition 7 (Operator-based distance). *Let Σ be a knowledge base, $\phi(X)$ be a formula, and $GenOp$ be a set of generalization operators. Let \mathcal{G}_i be the set of formulas obtained by applying i operators from $GenOp$ to $\phi(X)$ in any possible way without allowing formulas that are equivalent to any formula in \mathcal{G}_j ($j \leq i$):*

$$\mathcal{G}_0 := \{\phi(X)\}$$

$$\mathcal{G}_i := \{\psi(X, Y) \mid \psi(X, Y) \in o(\mathcal{G}_{i-1}) \text{ where } o \in GenOp \text{ and for every } j \leq i \text{ there is no } \psi'(X, Y') \in \mathcal{G}_j \text{ such that } \psi'(X, Y') \equiv \psi(X, Y)\}$$

A formula $\psi(X, Y)$ has distance l to $\phi(X)$ (with respect to Σ and $GenOp$) if $\psi(X, Y)$ can be obtained by applying at least l generalization operators to $\phi(X)$; in other words, if $\psi(X, Y) \in \mathcal{G}_l$.

4 Cooperative Query Answering for Conjunctive Queries

In the following subsections, we analyze three generalization operators that modify a given query in order to obtain a generalized query with informative answer (if there is any at all); they are called Dropping condition (DC), Anti-instantiation (AI), and Goal replacement (GR).

4.1 Dropping Condition

As we only consider conjunctions in a query, ignoring one of the conjuncts makes the query more general. For a given conjunctive query, we will call a “subquery” a conjunctive query that contains a subset of the conjuncts of the original query. Assume the original query consists of n conjuncts (we will say that the query has length n). The generalization operator dropping condition (see Operator [1](#)) returns a subquery of length $n - 1$. Note that in this case the free variables of Q^{gen} are a subset of the variables of Q and hence Y (from Definition [5](#)) is empty because no new variables are introduced; this is why Y is left out of Q^{gen} .

Operator 1. Dropping condition (DC)

Input: Query $Q(X) = L_1 \wedge \dots \wedge L_n$ of length n

Output: Generalized query $Q^{gen}(X)$ as a subquery of length $n - 1$

1: Choose literal L_j from L_1, \dots, L_n

2: **return** $L_1 \wedge \dots \wedge L_{j-1} \wedge L_{j+1} \wedge \dots \wedge L_n$

There are $\binom{n}{n-1}$ such subqueries of length $n - 1$. For example, $P(x) \wedge Q(x)$ is a subquery of length 2 of $P(x) \wedge Q(x) \wedge R(x)$ generated by dropping the condition $R(x)$. The operator DC complies with Definition [4](#) because it holds tautologically (that is, in any knowledge base) that a conjunctive query implies all its subqueries. In particular, when $Q(X)$ is a conjunctive query of length n and $Q^{gen}(X)$ is a subquery of length $n - 1$, then $\models \forall X (Q(X) \rightarrow Q^{gen}(X))$.

Proposition 1. *DC is a deductive generalization operator.*

4.2 Anti-instantiation

With anti-instantiation (see Operator [2](#)) a new variable is introduced in the query: the free variables of Q^{gen} are equal to the free variables of Q plus one new variable y ; hence $Y = \{y\}$. Thus, some conditions in the query are relaxed and the resulting query also covers answers with different values for y .

Note that here AI not only applies to constants but also to variables. In particular, AI covers these special cases:

Operator 2. Anti-instantiation (AI)

Input: Query $Q(X) = L_1 \wedge \dots \wedge L_n$ of length n

Output: Generalized query $Q^{gen}(X, Y)$ with Y containing one new variable

- 1: From $Q(X)$ choose a term t such that t is
 - either a variable occurring in $Q(X)$ at least twice
 - or a constant
 - 2: Choose one literal L_j where t occurs
 - 3: Let L'_j be the literal with one occurrence of t replaced with a new variable
 - 4: **return** $L_1 \wedge \dots \wedge L_{j-1} \wedge L'_j \wedge L_{j+1} \wedge \dots \wedge L_n$
-

- turning constants into variables: $P(a)$ is converted to $P(x)$ (see [12])
- breaking joins: $P(x) \wedge S(x)$ is converted to $P(x) \wedge S(y)$ (introduced in [4])
- naming apart variables inside atoms: $P(x, x)$ is converted to $P(x, y)$

For each constant a all occurrences must be anti-instantiated (that is, there are $\sum_a |\text{occ}(a, Q(X))|$ possible anti-instantiations); the same applies to variables v (that is, there are $\sum_v (|\text{occ}(v, Q(X))|)$ possible anti-instantiations) – however, with the exception that if v only occurs twice ($|\text{occ}(v, Q(X))| = 2$), one occurrence of v need not be anti-instantiated due to equivalence. AI is a deductive generalization operator according to Definition 4 because it holds tautologically that the literal containing term t implies the literal where t is replaced by a new variable y ; thus, $\models \forall X \exists y (L_j \rightarrow L'_j)$, because the term t can be taken as a witness for the existence of a value for y . The same applies to the whole query: $\models \forall X \exists y (Q(X) \rightarrow Q^{gen}(X, Y))$.

Proposition 2. *AI is a deductive generalization operator.*

4.3 Goal Replacement

Goal replacement (see Operator 3) checks if the body of a rule in the knowledge base can be mapped (via a substitution) to a subquery. If so, the head of the rule replaces the subquery (with the substitution applied). In this way goal replacement potentially introduces new constants and new predicate symbols in the generalized query and possibly some of the original query variables disappear. The resulting query may cover a greater range of values and hence lead to an informative answer for the user. With the GR operator, a single-headed range-restricted rule from the knowledge base Σ is involved and the property of being a deductive generalization operator now indeed depends on the knowledge base. Due to the range-restrictedness of rules, the GR operator does not introduce new variables. This is why in this case Y (from Definition 5) is the empty set and dropped from the notation.

For example, the query $P(a, y) \wedge P(b, y)$ can be converted to $Q(y) \wedge P(b, y)$ given the rule $P(x, y) \rightarrow Q(y)$ and the substitution $\theta = \{a/x, y/y\}$. The difficulty with GR is that Σ has to be checked for rules with a matching body. Showing that GR complies with Definition 4 amounts to showing that $\{\forall X (L_{i_1} \wedge \dots \wedge L_{i_m}) \rightarrow L'\} \models \forall X (Q(X) \rightarrow L'\theta)$; and this statement holds because $L_{i_1}\theta \wedge \dots \wedge L_{i_m}\theta$ is a subquery of $Q(X)$. Hence, it holds that: $\Sigma \models \forall X (Q(X) \rightarrow Q^{gen}(X'))$.

Operator 3. Goal replacement (GR)

Input: Query $Q(X) = L_1 \wedge \dots \wedge L_n$ of length n

Output: Generalized query $Q^{gen}(X)$ containing a replacement literal

- 1: From Σ choose a single-headed range-restricted rule $L_{i_1} \wedge \dots \wedge L_{i_m} \rightarrow L'$ such that there is a substitution θ for which all literals $L_{i_1}\theta, \dots, L_{i_m}\theta$ occur in $Q(X)$
 - 2: Let $L_{i_{m+1}}, \dots, L_{i_n}$ denote the literals of $Q(X)$ apart from $L_{i_1}\theta, \dots, L_{i_m}\theta$
 - 3: **return** $L_{i_{m+1}} \wedge \dots \wedge L_{i_n} \wedge L'\theta$
-

Proposition 3. *GR is a deductive generalization operator.*

We remark that [4] introduce a more versatile goal replacement operator using so-called reciprocal clauses that may introduce several new conjuncts (and hence some further new predicate symbols) in the query; but for sake of simplicity we do not follow this approach further in this paper.

5 Combining Generalization Operators

We analyze the combination of the three operators DC, AI and GR. To the best of our knowledge a generalization method that combines these three operators has not been analyzed so far. In [17] generalization operators are combined with so-called conditional answers; but the generalization operators themselves are not combined with each other. Lattice properties of queries have already been analyzed by [5,11] for operators that can be uniformly applied to any conjunct of a query (like for example dropping conditions). In contrast to this, goal replacement deletes some conjuncts and introduces a new conjunct while AI introduces a new variable; hence the generalization behavior highly depends on applicability of GR and AI. The analysis of a combination of DC, AI and GR is indeed worthwhile: the main issue is that behavior of operators can have a greater impact when used in combination with other operators. For instance, maximal succeeding subqueries (MSSs) are important when using dropping conditions alone; see the in-depth discussion in [5]. Yet, in combination with other operators, the identification of MSSs may not be the only option to get informative answers.

Moreover, for finding MSSs, [5] shows that a depth-first and top-down search procedure can efficiently find *some* of the minimal succeeding subqueries. However, a disadvantage of this depth-first search might be that the resulting query is far away from the user's original intention. Hence when searching for generalized queries that are close to the original one, an exhaustive search (either depth-first or breadth-first) up to a user-defined bound of generalization steps can be a more viable option and was also proposed in [11].

For the combination of DC, AI and GR, we iteratively apply the three generalization operators in all possible ways. In other words, we compute the sets \mathcal{G}_i from Definition 7. In order to avoid unnecessary generation of duplicate queries, we show in the following some properties of the sets of queries that are obtained by combining any two of the generalization operators DC, AI and GR. First

we study the combination of AI and DC. As AI introduces a new variable, set-containment \subseteq is meant up to variable renaming. When AI is followed by DC, the resulting queries can indeed be found by either dropping conditions alone or by commuting the operators.

Lemma 1 (DC following AI). *Given a query $Q(X)$, let the set of queries $S_1 = AI(DC(Q(X)))$ be obtained by first dropping conditions and then anti-instantiating, let $S'_1 = DC(Q(X))$ be the set of queries obtained by dropping conditions, and let the set of queries $S_2 = DC(AI(Q(X)))$ be obtained by first anti-instantiating and then dropping conditions. Then $S_2 \subseteq S_1 \cup S'_1$ (up to variable renaming).*

When anti-instantiating a term and then dropping the conjunct that contains the term, the two operations coincide with dropping the conjunct alone (and hence the query belongs to set S'_1). When however after anti-instantiating a term, a conjunct different from the one containing the term is dropped, the operators can be applied in reverse order (and hence the query belongs to set S_1). It follows that it suffices to apply AI after DC. In the reverse direction, it can also be shown that $S_1 \subseteq S_2$.

Lemma 2 (GR following DC). *Given a query $Q(X)$, let the set of queries $S_1 = GR(DC(Q(X)))$ be obtained by first dropping conditions and then replacing goals and let the set of queries $S_2 = DC(GR(Q(X)))$ be obtained by first replacing goals and then dropping conditions. Then $S_1 \subseteq S_2$.*

This lemma holds due to the fact that when a conjunct is dropped from a query and then a goal replacement with respect to some rule from the knowledge base is executed on the remaining query, the same can be achieved by first applying the rule and then dropping the conjunct. Hence we can avoid the application of the GR operator after the DC operator. The reverse however does not hold: in some cases DC results in a query to which GR cannot be applied anymore; for example, while $\{S(x), C(x)\}$ is the set of generalized queries obtained from the query $Q(x) = A(x) \wedge B(x) \wedge C(x)$ by first applying the rule $A(x) \wedge B(x) \rightarrow S(x)$ in a goal replacement step and then dropping conditions, the generalized query $C(x)$ can never be obtained by reversing the operators.

Lemma 3 (GR following AI). *Given a query $Q(X)$, let the set of queries $S_1 = AI(GR(Q(X)))$ be obtained by first replacing goals and then anti-instantiating, let the set of queries $S'_1 = GR(Q(X))$ be obtained by goal replacement alone, and let the set of queries $S_2 = GR(AI(Q(X)))$ be obtained by first anti-instantiating and then replacing goals. Then $S_2 \subseteq S_1 \cup S'_1$ (up to variable renaming).*

This is due to the fact that rules that can be applied to anti-instantiated queries, can also be applied to the original query first and then applying the anti-instantiation: Assume that some term t in the original query is anti-instantiated to the new variable y ; then the substitution θ used in goal replacement maps some variable of the rule to y . However the same can be achieved by first applying goal replacement by letting θ map the variable to t and then (if it is still contained in the replacement literal) anti-instantiating it to y . That is, we only

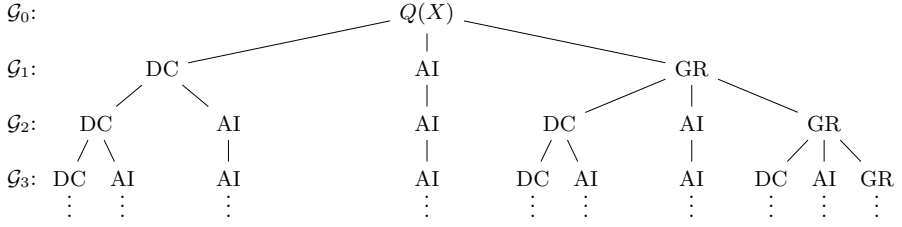


Fig. 2. Operator tree

Table 1. Generalization sets \mathcal{G}_1 to \mathcal{G}_4 for query $Q(X) = Ill(x, Flu) \wedge Ill(x, Cough)$

$DC(Q(X))$	$\{Ill(x, Cough), Ill(x, Flu)\}$
$\cup AI(Q(X))$	$\cup \{Ill(y, Flu) \wedge Ill(x, Cough), Ill(x, y) \wedge Ill(x, Cough), Ill(x, Flu) \wedge Ill(x, y)\}$
$\cup GR(Q(X))$	$\cup \{Treat(x, Medi) \wedge Ill(x, Cough)\}$
$DC(DC(Q(X)))$	$\{\varepsilon\}$
$\cup AI(DC(Q(X)))$	$\cup \{Ill(x, y)\}$
$\cup AI(AI(Q(X)))$	$\cup \{Ill(y, y') \wedge Ill(x, Cough), Ill(y, Flu) \wedge Ill(x, y'), Ill(x, y) \wedge Ill(x, y')\}$
$\cup DC(GR(Q(X)))$	$\cup \{Treat(x, Medi)\}$
$\cup AI(GR(Q(X)))$	$\cup \{Treat(y, Medi) \wedge Ill(x, Cough), Treat(x, y) \wedge Ill(x, Cough), Treat(x, Medi) \wedge Ill(x, y)\}$
$GR(GR(Q(X))) = \emptyset$	
$AI(AI(AI(Q(X))))$	$\{Ill(y, y') \wedge Ill(x, y'')\}$
$\cup AI(DC(GR(Q(X))))$	$\cup \{Treat(x, y)\}$
$\cup AI(AI(GR(Q(X))))$	$\cup \{Treat(y, y') \wedge Ill(x, Cough), Treat(y, Medi) \wedge Ill(x, y'), Treat(x, y) \wedge Ill(x, y')\}$
$DC(DC(DC(Q(X)))) = AI(DC(DC(Q(X)))) = AI(AI(DC(Q(X)))) = DC(DC(GR(Q(X)))) = \emptyset$	
$AI(AI(AI(GR(Q(X)))))$	$\{Treat(y, y') \wedge Ill(x, y'')\}$
$AI(AI(AI(AI(Q(X)))) = AI(AI(DC(GR(Q(X)))) = \emptyset$	

have to apply AI to queries obtained by GR. The reverse direction does not hold because after AI some rule used for GR may not be applicable anymore.

Based on the above lemmata, we can now conclude that in order to compute the sets of generalized queries \mathcal{G}_i the three operators under scrutiny need only be applied in a certain order. This is illustrated in Figure 2.

Theorem 1 (Operator ordering). *When combining the generalization operators DC, AI and GR, the following computations can be avoided: GR following DC, DC following AI, and GR following AI.*

With this result, the search for generalized queries can be much more efficient. However the actual efficiency gain depends on the applicability of GR and AI: when GR is applicable, it suffices to apply it in the first generalization steps before any other operators, followed by DC steps and lastly AI steps. But even

if GR is not applicable we never even have to check for its applicability after DC or AI (this check includes finding matching rules and is hence quite expensive). As AI generates a lot of generalized queries, with our result we can apply it only in the last generalization steps without missing out any generalized query.

Example 1. We now give a comprehensive example. Assume knowledge base $\Sigma = \{Ill(\text{Pete}, \text{Flu}), Ill(\text{Mary}, \text{Cough}), Treat(\text{Mary}, \text{Medi}), Ill(x, \text{Flu}) \rightarrow Treat(x, \text{Medi})\}$. The user asks the query $Q(X) = Ill(x, \text{Flu}) \wedge Ill(x, \text{Cough})$ which fails in Σ (under both exact and disjunctive answer semantics). The generalization sets up to \mathcal{G}_4 are shown in Table 1. These four sets comprise all possible generalizations of $Q(X)$. Already \mathcal{G}_1 can give informative answers (like for example $Ill(\text{Pete}, \text{Flu})$ or $Treat(\text{Mary}, \text{Medi}) \wedge Ill(\text{Mary}, \text{Cough})$) and hence might satisfy the user’s needs.

6 Discussion and Conclusion

Although a given query fails, a knowledge base might still be able to return informative answers to a slightly modified query. We provided a profound analysis of the combination of the three operators DC, AI and GR that stepwise generalize conjunctive queries with positive and negative literals. A prototype implementation using a current theorem proving system (SOLAR [14]) is under development. One open issue is to avoid *overgeneralization* that leads to queries far from the user’s original intent; e.g., the AI operator can be prevented from generating unrestricted predicates (like $Ill(x, y)$) by reinstantiating and hence computing neighborhood queries in the sense of [17]. Moreover, generalization operators could for example be restricted by defining user preferences on the literals in a query. An extension of our approach might consider other logical settings (e.g., queries of a more general form, nonmonotonic reasoning or “conditional answers” [10, 17]). Another topic of future work may be the development of an algorithm that directly and incrementally returns informative answers in our setting (as done by [9] for RDF queries and ontologies) without computing the sets of generalized queries.

References

1. Bosc, P., HadjAli, A., Pivert, O.: Incremental controlled relaxation of failing flexible queries. *JIIS* 33(3), 261–283 (2009)
2. Chu, W.W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C.: CoBase: A scalable and extensible cooperative information system. *JIIS* 6(2/3), 223–259 (1996)
3. de Raedt, L.: Induction in logic. In: *MSL-1996*, pp. 29–38. AAAI Press, Menlo Park (1996)
4. Gaasterland, T., Godfrey, P., Minker, J.: Relaxation as a platform for cooperative answering. *JIIS* 1(3/4), 293–321 (1992)
5. Godfrey, P.: Minimization in cooperative response to failing database queries. *IJCS* 6(2), 95–149 (1997)

6. Godfrey, P., Minker, J., Novik, L.: An architecture for a cooperative database system. In: Risch, T., Litwin, W. (eds.) ADB 1994. LNCS, vol. 819, pp. 3–24. Springer, Heidelberg (1994)
7. Gorogiannis, N., Hunter, A.: Merging first-order knowledge using dilation operators. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 132–150. Springer, Heidelberg (2008)
8. Halder, R., Cortesi, A.: Cooperative query answering by abstract interpretation. In: Černá, I., Gyimóthy, T., Hromkovič, J., Jefferey, K., Královič, R., Vukolić, M., Wolf, S. (eds.) SOFSEM 2011. LNCS, vol. 6543, pp. 284–296. Springer, Heidelberg (2011)
9. Hurtado, C.A., Poulouvassilis, A., Wood, P.T.: Query relaxation in RDF. In: Spaccapietra, S. (ed.) Journal on Data Semantics X. LNCS, vol. 4900, pp. 31–61. Springer, Heidelberg (2008)
10. Inoue, K., Iwanuma, K., Nabeshima, H.: Consequence finding and computing answers with defaults. *J. Intell. Inf. Syst.* 26(1), 41–58 (2006)
11. Iwanuma, K., Inoue, K.: Minimal answer computation and SOL. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 245–258. Springer, Heidelberg (2002)
12. Michalski, R.S.: A theory and methodology of inductive learning. *Artificial Intelligence* 20(2), 111–161 (1983)
13. Motro, A.: Flex: A tolerant and cooperative user interface to databases. *IEEE Transactions on Knowledge & Data Engineering* 2(2), 231–246 (1990)
14. Nabeshima, H., Iwanuma, K., Inoue, K., Ray, O.: SOLAR: An automated deduction system for consequence finding. *AI Communications* 23(2-3), 183–203 (2010)
15. Pivert, O., Jaudoin, H., Brando, C., Hadjali, A.: A method based on query caching and predicate substitution for the treatment of failing database queries. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 436–450. Springer, Heidelberg (2010)
16. Plotkin, G.: Automatic methods of inductive inference. PhD thesis, University of Edinburgh (1971)
17. Sakama, C., Inoue, K.: Negotiation by abduction and relaxation. In: AAMAS 2007. IFAAMAS, pp. 1010–1025 (2007)
18. Shin, M.K., Huh, S.-Y., Lee, W.: Providing ranked cooperative query answers using the metricized knowledge abstraction hierarchy. *Expert Systems with Applications* 32(2), 469–484 (2007)

ReAction: Personalized Minimal Repair Adaptations for Customer Requests

Monika Schubert, Alexander Felfernig, and Florian Reinfrank

Applied Software Engineering, Institute of Software Technology,
Graz University of Technology,
Inffeldgasse 16b/II, 8010 Graz, Austria
{monika.schubert,alexander.felfernig,florian.reinfrank}@ist.tugraz.at

Abstract. Knowledge-based recommender systems support users in finding interesting products from large and potentially complex product assortments. In such systems users continuously refine their specifications which the product has to satisfy. If the specifications are too narrow no product can be retrieved from the product assortment. Instead of just notifying the customer that no product could be found we introduce an approach called *ReAction* to support customers with minimal repair adaptations. In this paper we give a detailed explanation of our algorithm. Besides that we present the results of a detailed empirical evaluation focussing on the quality as well as on the runtime performance. The work presented is relevant for designers and developers of database systems as well as knowledge-based recommender systems interested in (i) identifying relaxations for database queries, (ii) applying and dealing with user utilities, and (iii) improving the system usability through suggesting minimal repair adaptations for inconsistent queries.

1 Introduction

Knowledge-based recommender systems [1] are interactive software applications that support users in effective identification of interesting items from large and potentially complex product assortments. The main advantage of knowledge based recommender systems is that they incorporate explicit knowledge about the items and their properties as well as an explicit formulation of customer requirements. This deep knowledge is the foundation for intelligent explanations and repair actions [8].

A customer normally interacts with a knowledge-based recommender in four steps: (1) the *requirement elicitation phase*, where the system assists the customer in specifying their their requirements. (2) the *suitability check*, where the recommender checks if there exists a product that satisfies the requirements. (3) the *result presentation* phase in which the system presents the recommended product(s) to the customer and finally in phase (4) the customer can ask the system for an explanation why this product was recommended.

Although there are different methods for elicitation of customer requirements as well as for finding and ranking products, all systems have in common that they

treat - at least in the beginning - some or all customer requirements as constraints which the product has to satisfy [11]. If customers continuously refine their requirements, situations can occur where no product of the assortment fulfils all these requirements. This situation is called the *no solution could be found dilemma* [10]. If such a situation occurs the system has to notify the customer that there is no product available. It is rather disappointing for the customer if no further support is available to get out of this situation.

A question that has recently attracted the interest is how to deal with the no solution could be found dilemma. A basic approach to tackle this challenge is an incrementally elimination of one or more constraints of the user requirements. Reiter [14] introduced the hitting set directed acyclic graph (HSDAG) which can be used to systematically identify and resolve conflict sets. There are a couple of algorithms that calculate such conflict sets (see QuickXplain [6], FastXplain [16]). *QuickXplain* [6] uses a divide and conquer approach to split the customer requirements and to identify the conflicting ones. In comparison to this *FastXplain* [16] exploits the structural properties of a database table derived from the product assortment and the customer requirements. Another algorithm which uses this table was introduced by Jannach [5] which builds a lattice to identify query relaxations in content-based recommenders. Felfernig et al. [8] introduced an approach to identify minimal repair actions for inconsistent user requirements. This approach integrates the calculation of explanations with collaborative problem solving techniques on the basis of QuickXplain [6]. In this paper we are introducing *ReAction* - an approach that identifies minimal repair adaptations. *ReAction* performs better compared to state-of-the-art algorithms (CorrectiveRelax [12] and QuickXplain [6]) in terms of prediction quality and runtime performance.

The remainder of the paper is organised as follows: in Section 2 we introduce an example which we will use throughout the paper to illustrate our approach. In Section 3 we demonstrate our approach and describe the *ReAction* algorithm in detail. An empirical evaluation that focuses on prediction quality and runtime performance was performed and the results are presented in Section 4. In Section 5 we evaluate our approach against related work in the field and with Section 6 we conclude our paper and give an outlook on future work.

2 Example: Digital Camera

The following simplified assortment of digital cameras will serve as a working example throughout this paper. The set of digital cameras $P = \{p_1, p_2, \dots, p_9\}$ is stored in the product table (see Table 1). For each item in the table the following attributes $A = \{a_1, a_2, \dots, a_8\}$ are specified: a_1 : *mpix* specifies the maximum number of megapixel a camera can take; a_2 : *display* enumerates the inches of the display at the backside of the camera; a_3 : *zoom* describes the optical zoom; a_4 : *stabilization* informs about the availability of a stabilisation system; a_5 : *waterproof* specifies if the camera is waterproof or not; a_6 : *colour* indicates the main colour of the camera; the a_7 : *weight* is given in pounds and a_8 : *price* specifies the costs (in dollar) of the camera.

Let us assume that the following requirements are specified by our current customer: $R = \{ r_1: mpix > 10.0, r_2: display \geq 3.0, r_3: zoom \geq 4x, r_4: stabilization = yes, r_5: waterproof = yes, r_6: weight \leq 2.0, r_7: price < 250 \}$. The feasibility of these requirements can simply be checked by a relational query $\sigma_{[R]}P$ where $\sigma_{[R]}$ represents the selection criteria of the query. For example $\sigma_{[mpix > 10.0]}P$ would result in $\{p_1, p_2, p_3, p_5, p_7, p_8\}$.

Table 1. Example assortment of digital cameras $P = \{p_1, p_2, \dots, p_9\}$

id	mpix	display	zoom	stabilization	waterproof	colour	weight	price
p_1	12.0	3.0	3.6x	yes	no	red	1.4	129
p_2	14.1	3.5	5x	yes	no	pink	2.0	329
p_3	12.1	2.7	4x	yes	yes	blue	2.3	107
p_4	7.5	2.5	4x	no	yes	silver	1.2	59
p_5	12.0	2.7	4x	yes	no	silver	4.7	170
p_6	10.0	3.0	3.8x	no	no	blue	0.7	349
p_7	14.0	2.7	3.6x	yes	yes	green	0.63	240
p_8	12.3	2.7	8x	yes	no	black	2.95	450
p_9	10.0	2.9	3.8x	yes	yes	red	0.9	221

For our current customer requirements R the query $\sigma_{[r_1, r_2, r_3, r_4, r_5, r_6, r_7]}P = \emptyset$. This means that no solution could be found for these requirements. In such a situation customers ask for repair actions which help them to restore consistency between their request R and the underlying product assortment P . One possible minimal repair action that can be suggested to the customer would be $repair(r_3, r_5) = \{zoom = 3.6x, waterproof = no\}$. If the customer accepts this repair action, the recommended product which satisfies all (adapted) customer requirements is p_1 . A repair action $repair$ is minimal if there does not exist a repair action $repair'$ which is a subset of $repair$.

3 Identifying Minimal Repair Sets

In this section we present the main contributions of our paper. Our approach to identify minimal repair sets is founded on the concepts of Model-based Diagnosis (MBD) [2, 14]. These concepts allow the automated identification of minimal sets of faulty constraints in the customer requirements [7]. In our setting of knowledge-based recommender systems we can consider the product assortment as a model. The intended behaviour of our system would be that based on the customer request a recommendation can be found. If this is not the case (like in the example) the diagnosis task is to determine those requirements that need to be relaxed or deleted in order to find a recommendation.

A *Customer Request (CR) Diagnosis Problem* can be defined as:

Definition 1 (CR Diagnosis Problem): A CR Diagnosis Problem is defined as a tuple $\langle R, P \rangle$ where $R = \{r_1, r_2, \dots, r_n\}$ is a set of requirements (the request from the customer) and $P = \{p_1, p_2, \dots, p_m\}$ is a set of products.

Founded on this definition of a CR Diagnosis Problem, a CR Diagnosis (Customer Request Diagnosis) can be defined as:

Definition 2 (CR Diagnosis): A CR Diagnosis for $\langle R, P \rangle$ is a set $d = \{r_1, r_2, \dots, r_q\} \subseteq R$ s.t. $\sigma_{[R-d]}P \neq \emptyset$. A diagnosis is minimal iff there does not exist a diagnosis d' with $d' \subset d$.

As already mentioned the customer requirements of our working example are inconsistent with the product assortment in Table 1. Thus there is no product that satisfies all customer requirements $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$. The corresponding diagnoses for this problem are: $D = \{d_1 : \{r_3, r_5\}, d_2 : \{r_5, r_7\}, d_3 : \{r_2, r_6\}, d_4 : \{r_2, r_3\}, d_5 : \{r_1, r_2, r_4\}\}$ since there is always at least one product for $\sigma_{[R-d_i]}P$. For example for the diagnosis $d = \{r_3, r_5\}$ this is $\sigma_{[R-\{r_3, r_5\}]}P = \{p_1\}$.

After having identified at least one diagnosis we can propose repair adaptations for each diagnosis. Thus we are identifying possible adaptations for customer requirements in a way that customers are able to find at least one item that satisfies their requirements. A possible repair adaptation for the diagnosis $d = \{r_3, r_5\}$ would be $\Pi_{[attributes(d)]}(\sigma_{[R-d]}P) = \Pi_{[zoom, waterproof]}(\sigma_{[r_1, r_2, r_4, r_6, r_7]}P) = \Pi_{[zoom, waterproof]}(\{p_1\}) = \{zoom = 3.6x, waterproof = no\}$

3.1 Algorithm

In order to systematically identify repair actions we are introducing the algorithm *ReAction* (see Algorithm 1). This algorithm is based on the algorithm *ReDiagnoses* (see Algorithm 2). *ReAction* (Algorithm 1) takes the customer request $R = \{r_1, r_2, \dots, r_n\}$ and the product assortment $P = \{p_1, p_2, \dots, p_m\}$ as inputs. Besides that it takes the sorting criteria c .

Algorithm 1. ReAction(R, P, c): repairs

```

{Input: R - customer request (requirements)}
{Input: P - table of all products}
{Input: c - sort criteria (i.e. utility)}
{Input: n - number of diagnoses that should be calculated}
{Output: repairs - sorted list of repair actions}
if  $\sigma_{[R]}P \neq \emptyset$  then
  return  $\emptyset$ 
end if
 $R \leftarrow sort(R, c)$ 
 $d \leftarrow ReDiagnosis(\emptyset, R, P)$ 
 $repairs \leftarrow \Pi_{[attributes(d)]}(\sigma_{[R-d]}P)$ 
return  $repairs$ 

```

First the algorithm sorts the user requirements according to the sorting criteria c . This sorting criteria can be similar to the total ordering used by [6,9]. For evaluation purposes we are applying two different utility values: First, the *user*

utility which has been specified by the customer when interacting with the recommender system. Second, the *log utility* values are based on past interactions of all customers with the system. For each attribute a_i we sum up all occurrences of each selected repair adaptation of the past. The log utility is defined as:

$$\text{logutility}(a_i) = \text{occurrence}(a_i) / \text{all occurrences}$$

Note that there are different personalisation strategies besides using utility values (i.e. similarity, probability, hybrids) that can be used as sort criteria. For our working example we used utility values specified by the customer these are shown in Table 2.

Table 2. Utility values specified by the customer through an interaction with the system (in percentage)

	mpix	display	zoom	stabilization	waterproof	weight	price
user utility	18.0	5.0	22.0	7.0	8.0	15.0	25.0
log utility	12.0	7.0	20.0	8.0	10.0	17.0	26.0

The algorithm *ReAction* sorts the user requirements according to an increasing utility value. This results in the following ordering: $\{r_2, r_4, r_5, r_6, r_1, r_3, r_7\}$ for our working example. If there exists a conflicting situation (no product can be found in the product assortment P for the customer request R) the algorithm *ReDiagnoses* (see Algorithm 2) is called with this ordered set of constraints. This algorithm returns one preferred minimal diagnosis. This is the diagnosis including the attributes with the lowest overall utility as these are the most probable ones to be changed by the customer [6]. Based on this diagnosis all repair actions - those assignments to the faulty constraints that lead to at least one product - are calculated and returned.

The basic approach of the algorithm *ReDiagnosis* is to follow a divide and conquer strategy (see Figure 1). *ReDiagnosis* takes a sorted set of requirements, for example, $R = \{r_2, r_4, r_5, r_6, r_1, r_3, r_7\}$ as well as the product table as parameters. This set is splitted into two subsets namely $R_1 = \{r_2, r_4, r_5, r_6\}$ and $R_2 = \{r_1, r_3, r_7\}$. Then it is checked whether the query $\sigma_{[AR]}P$ returns any product. The fact that $\sigma_{[AR]}P = \sigma_{[r_1, r_3, r_7]}P = \{p_3, p_4, p_5\} \neq \emptyset$ returns at least one product means that a subset of $R_1 = \{r_2, r_4, r_5, r_6\}$ constitutes a diagnosis. Now we have to investigate R_1 further and identify those elements that are part of the minimal diagnosis. Therefore we construct (from $R_1 = \{r_2, r_4, r_5, r_6\}$) the sets $R'_1 = \{r_2, r_4\}$ and $R'_2 = \{r_5, r_6\}$. The consistency of AR' is checked by $\sigma_{[AR']}P = \sigma_{[r_1, r_3, r_5, r_6, r_7]}P = \emptyset$. This means that the current set of requirements AR' is inconsistent and we have to further investigate in it. We split the current set of requirements up into $\{r_5\}$ and $\{r_6\}$. $AR'' = \{r_1, r_3, r_6, r_7\}$ is inconsistent which means that r_6 is an element of the diagnosis. The other part of the diagnosis comes from the set $R'_1 = \{r_2, r_4\}$. We explore the sets $\{r_2\}$ and $\{r_4\}$ which results in the set of $\{r_2\}$. Summarizing the preferred minimal

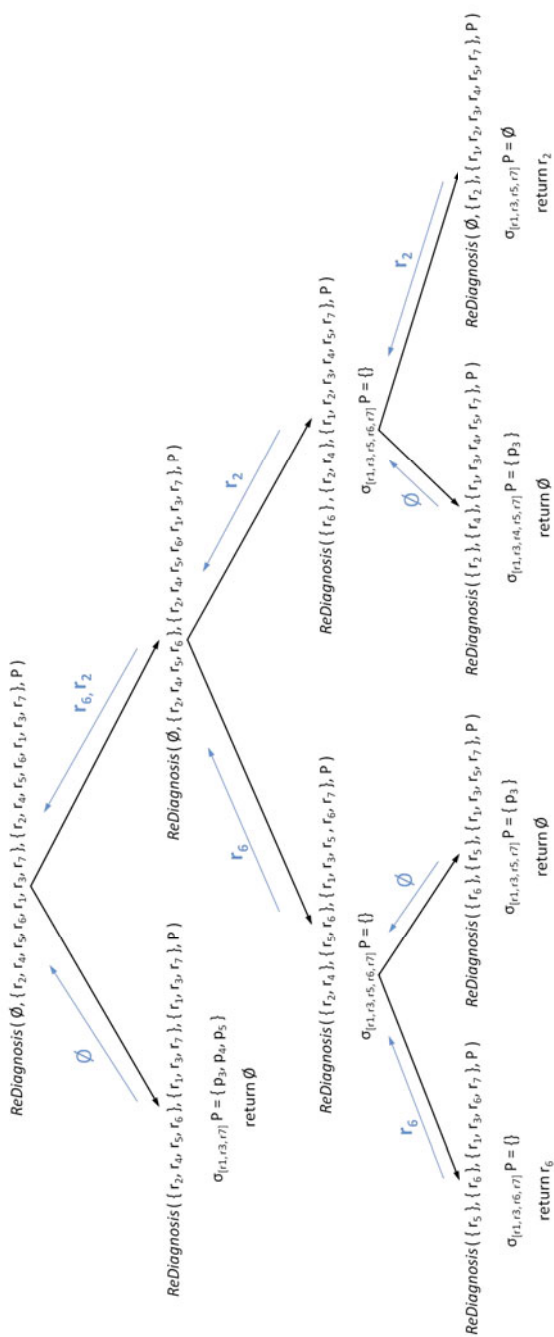


Fig. 1. Execution trace for *ReDiagnosis* to calculate one diagnosis for the CR diagnosis problem $R = \{r_2, r_4, r_5, r_6, r_1, r_3, r_7\}$ and the product table P (see Table [D](#))

Algorithm 2. ReDiagnosis(D, R, AR, P)

```

{Input: D - delta set, initially empty}
{Input: R - customer request (requirements)}
{Input: AR - all customer requirements (initial same as R)}
{Input: P - table of all products}
{Output: diagnosis - set of faulty requirements}
if  $D \neq \emptyset$  and  $\sigma_{[AR]}P \neq \emptyset$  then
  return  $\emptyset$ 
end if
if singleton(R) then
  return R
end if
 $k \leftarrow \lceil \frac{n}{2} \rceil$ 
 $R_1 \leftarrow \{r_1, \dots, r_k\}$ 
 $R_2 \leftarrow \{r_{k+1}, \dots, r_n\}$ 
 $\delta_1 \leftarrow \text{ReDiagnosis}(R_1, R_2, AR - R_1, P)$ 
 $\delta_2 \leftarrow \text{ReDiagnosis}(\delta_1, R_1, AR - \delta_1, P)$ 
return  $(\delta_1 \cup \delta_2)$ 

```

diagnosis for the CR diagnosis problem $\langle R, P \rangle$ where $R = \{r_1, r_2, \dots, r_7\}$ and $P = \{p_1, p_2, \dots, p_9\}$ is $d_1 = \{r_6, r_2\}$. The complete trace of the algorithm for the example of Section 2 is shown in Figure 1.

3.2 More Repair Sets

The *ReDiagnosis* algorithm (Algorithm 2) can be adapted in order to calculate more than one diagnosis. Inspired by the hitting set directed acyclic graph (HS-DAG) [14] the adapted algorithm *ReDiagnoses-Tree* builds up a tree based on diagnoses. Basically in each step of the algorithm we identify one diagnosis using *ReDiagnosis* which we add as edges to the current node. In Figure 2 you can see the tree for our working example of Section 2. The first diagnosis we retrieve from *ReDiagnosis* is $d_1(R) = \{r_6, r_2\}$. This diagnosis is added to the root node and for each element of the diagnosis we are adding one edge. In the next step we calculate the next diagnosis for the left path - namely for the requirements except the path $(R - r_6)$. When calling *ReDiagnosis* with $R - r_6$ we receive the diagnosis $d_2(R - r_6) = d_2(r_2, r_4, r_5, r_1, r_3, r_7) = \{r_1, r_4, r_2\}$ as a result. If we cannot find a diagnosis for a leaf anymore ($R - path$ is consistent) we close this leaf. The whole algorithm continues until all leaves are closed or a given number of diagnoses is reached.

4 Evaluation

Our empirical evaluation is based on two parts - first the quality evaluation and second the performance evaluation. In order to show the quality of our algorithm we looked at the prediction quality of *ReAction*. We compared these results to the

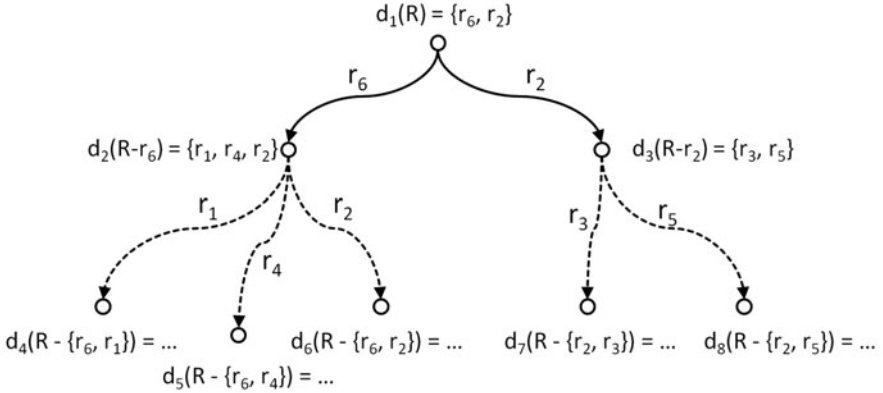


Fig. 2. Tree that is built for the example introduced in Section 2. The first diagnosis of the requirements sorted according to the utility returned by the *ReDiagnosis* is $d_1 = \{r_6, r_2\}$. This is added to the root node.

algorithm *QuickXplain* introduced by Junker [6] as well as the *CorrectiveRelax* algorithm introduced by O’Callaghan et al [12]. *QuickXplain* only calculates minimal conflict sets. Thus we are combining it with the hitting set directed acyclic graph (HSDAG) by Reiter [14] to systematically identifying all minimal diagnoses which are again the foundation for repair actions. The *CorrectiveRelax* algorithm uses a divide and conquer approach to determine minimal preferred diagnoses by constructing the complement of a maximal relaxation.

In order to evaluate the prediction quality of *ReAction* we used the interaction log of a financial service application that has been developed for one of the largest financial service providers in Austria¹. The interaction log contains 1703 sessions and each session holds information about the specific value of each attribute. In 418 sessions the requirements of the customers became inconsistent. For these sessions the log incorporates information on which repair action has been selected. On the basis of this dataset we compared the prediction quality of *ReAction* with two approaches (*CorrectiveRelax* [12] and *QuickXplain* [6]).

For the evaluation all algorithms had to predict the repair action that led to the selected item. The *ReAction* algorithm used once the user utility values and once the utility values retrieved from the interaction log (see Table 2). After applying the algorithms (*ReAction* with user utility, *ReAction* with log utility, *QuickXplain* using the user utilities as a preference criteria and *CorrectiveRelax* with user utility) to the dataset we measured the prediction quality. The prediction quality is defined as:

$$precision(d, n) = \begin{cases} 1, & \text{iff } d \text{ is among the top } n, \\ 0, & \text{otherwise.} \end{cases}$$

¹ www.hypo-alpe-adria.at

We evaluated the precision for the top $n = \{1, 3, 5, 10\}$ repair actions and measured the precision quality. The average number of repair alternatives was 20.42 (std.dev. 4.51). The results of the evaluation are presented in Table 3. As we can see from the results the *ReAction* algorithm using the log utility performs nearly similar compared to the one using the user utility. This is a really interesting observation as the log utility is based on the selected repairs retrieved from a lot of customers and the user utility is based on the customers’ preferences. This means that in this application there is no need to ask customers explicitly for their preferences but the past interaction of all customers are sufficient. Although this is depending on the characteristics of the domain, the application as well as on the selection criteria used. The precision quality of the *CorrectiveRelax* is the same as the precision quality *ReAction*. This is based on the fact that both are using the same preference criteria and both are keeping these criteria throughout the calculation.

Table 3. Precision values for the top-n ranked repair actions

	n=1	n=3	n=5	n=10
ReAction (user utility)	0.179	0.534	0.700	0.935
ReAction (log utility)	0.175	0.501	0.878	0.921
QuickXplain	0.166	0.481	0.740	0.875
CorrectiveRelax	0.179	0.534	0.700	0.935

The second aspect that we evaluated is the runtime performance for determining all repair actions. All algorithms were implemented in Java that operates with a standard relational database system realized in MySQL. All tests and measurements have been performed on a standard desktop PC (*Intel® Core™2 Quad CPU Q9400* CPU with *2.66GHz* and *2GB* RAM). We generated random settings for our evaluation that represent typical knowledge-based recommender sessions. The only condition that needs to hold for all generated settings is that the generated customer request needs to be inconsistent with the generated product assortment. We used product assortments with an increasing number of items ($I = \{100, 200, 300, 400, 500, 1000, 5000, 10000\}$) and 15 constraints. 15 constraints is already a high number of constraints when it comes to an user interaction, but in order to get a clearer understanding about the runtime quality we have chosen a high number. We compared the runtime of all algorithms calculating $\{1, 3, 10, all\}$ repair actions. The results are show in Figure 3. As we can see in Figure 3 the *ReAction* algorithm performs slightly better compared to the *CorrectiveRelax* [12] and *QuickXplain* [6]. And although the *CorrectiveRelax* has the same prediction quality as the *ReAction* algorithm, the *ReAction* algorithm is faster.

Summarizing we can say that our empirical findings confirmed our assumption that on a quality level the algorithm *ReAction* is better compared to the *QuickXplain* algorithm introduced by Junker [6]. Besides that it is also faster in calculating repair actions compared to the *CorrectiveRelax* [12] as well as to the *QuickXplain* [6].

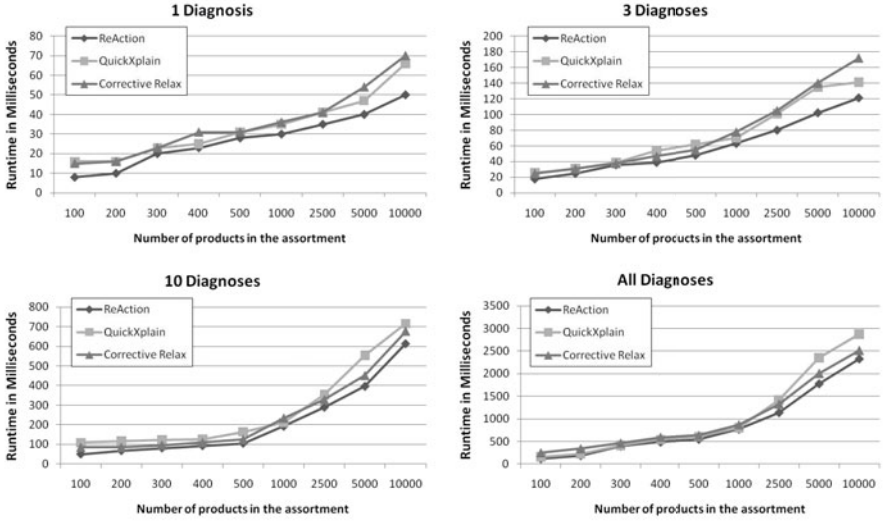


Fig. 3. Performance evaluation of the algorithms *ReAction*, *QuickXplain* [6] and *Corrective Relax* [12] to calculate $\{1, 3, 10, all\}$ repair actions for an increasing number of items and 15 constraints

5 Related Work

Junker [6] introduced the algorithm *QuickXplain* for identifying minimal conflict sets. A conflict set is a set $CS \subseteq R$ s.t. $P \cup CS$ is inconsistent ($\sigma_{[CS]}P = \emptyset$). These minimal conflict sets can be used to identify minimal diagnoses. The actual calculation of the diagnoses is done by a combination of the *Hitting Set Directed Acyclic Graph* (HSDAG) by Reiter [14] and the *QuickXplain* [6]. Similar to *QuickXplain*, *ReDiagnosis* relies on a divide and conquer strategy but focuses on the determination of a minimal diagnosis (in contrast to a minimal conflict set). As shown by Felfernig et al. [9] the direct identification of diagnoses - similar to our approach - is much more effective compared to the calculation using minimal conflict sets and the HSDAG.

Relevant research on supporting users in interactive systems (e.g. knowledge-based recommender systems) was introduced by Felfernig et al. [8]. The authors proposed an approach that calculates leading diagnoses on the basis of similarity measures used for determining n-nearest neighbours.

A general approach to the identification of preferred diagnoses was introduced by DeKleer in [3] where probability estimates are used to determine the leading diagnoses with the overall goal to minimize the number of measurements needed for identifying a malfunctioning device.

O’Sullivan et al. [13] introduced minimal exclusion sets which correspond to minimal diagnoses [4, 14]. The authors propose an approach to determine of representative explanations that can help to increase the probability of finding

an acceptable diagnosis. Compared to O’Sullivan et al. [13] our approach focuses more on user utility than on the degree of representativeness.

Schlobach et al. [15] introduced an approach that uses pinpointing for the identification of repairs for incoherent terminologies. These pinpoints prevent the algorithm from calculating minimal hitting sets by using the superset to approximate minimal diagnoses. To compute the pinpoints themselves, all minimal conflict sets are needed. This leads to an overhead because compared to the approach introduced by Reiter [14] these are computed on demand.

Relevant research of the field of database systems focuses on the consistency of the database itself. Zhou et al. [17] proposed an approach to query relaxations based on schemes. This approach focuses on exploiting schemes to effectively query vaguely structured information, whereas our approach assumes that the products in the database are complete.

6 Conclusion and Future Work

In this paper we introduced the algorithm *ReAction* to identify minimal repair adaptations for a customer request on a product table. The *ReAction* algorithm is based on the algorithm *ReDiagnosis* which calculates the preferred minimal diagnosis. The preference is given by an increasing user utility. For our evaluation we used two types of utilities: the user utility given through an interaction with the user and the log utility retrieved from past interactions of all customers. Based on the preferred minimal diagnosis the *ReAction* algorithm identifies all repair actions i. e. assignments to the faulty constraints that lead to at least one product. Using our working example we showed how to calculate repair adaptations (algorithm *ReAction*) for the preferred minimal diagnoses. After proposing a general approach we systematically evaluated the suitability by applying this algorithm to different datasets. We compared our approach to the algorithms *CorrectiveRelax* [12] and *QuickXplain* [6].

The results of this paper suggest using diagnoses based on user utilities instead of low cardinality diagnoses for calculating repair sets. Our approach can be applied easily to other domains than knowledge-based recommender systems, like any other system that uses queries that can get inconsistent. We are currently extending the approach in order to include different personalization strategies and the preliminary results are encouraging.

References

1. Burke, R.: Knowledge-based recommender systems. In: Library and Information Systems, vol. 69(32), pp. 180–200. Marcel Dekker, New York (2000)
2. DeKleer, J., Williams, B.: Diagnosing Multiple Faults. *Artificial Intelligence* 32(1), 97–130 (1987)
3. DeKleer, J.: Using crude probability estimates to guide diagnosis. *Artificial Intelligence* 45(3), 381–391 (1990)
4. DeKleer, J., Mackworth, A., Reiter, R.: Characterizing diagnoses and systems. *Artificial Intelligence* 56(2-3), 197–222 (1992)

5. Jannach, D.: Finding preferred query relaxations in content-based recommenders. *Intelligent Techniques and Tools for Novel System Architectures* 109, 81–97 (2008)
6. Junker, U.: QuickXplain: Preferred Explanations and Relaxations for Over-Constrained Problems. In: *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 167–172. The AAAI Press, California (2004)
7. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based Diagnosis of Configuration Knowledge Bases. *Artificial Intelligence* 152(2), 213–234 (2004)
8. Felfernig, A., Friedrich, G., Schubert, M., Mandl, M., Mairitsch, M., Teppan, E.: Plausible Repairs for Inconsistent Requirements. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 791–796 (2009)
9. Felfernig, A., Schubert, M., Zehentner, C.: An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 25(2), 1–10 (2011)
10. Pu, P., Chen, L.: User-Involved Preference Elicitation for Product Search and Recommender Systems. *AI Magazine* 29(4), 93–103 (2008)
11. McSherry, D.: Incremental Relaxation of Unsuccessful Queries. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004. LNCS (LNAI)*, vol. 3155, pp. 331–345. Springer, Heidelberg (2004)
12. O’Callaghan, B., O’Sullivan, B., Freuder, E.C.: Generating corrective explanations for interactive constraint satisfaction. In: van Beek, P. (ed.) *CP 2005. LNCS*, vol. 3709, pp. 445–459. Springer, Heidelberg (2005)
13. O’Sullivan, B., Papdopoulos, A., Faltings, B., Pu, P.: Representative explanations for over-constrained problems. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 323–328 (2007)
14. Reiter, R.: A theory of diagnosis from first principles. *Artificial Intelligence* 23(1), 57–95 (1987)
15. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. *Journal of Automated Reasoning* 39(3), 317–349 (2007)
16. Schubert, M., Felfernig, A., Mandl, M.: FastXplain: Conflict Detection for Constraint-Based Recommendation Problems. In: *Trends in Applied Intelligent Systems: 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Cordoba, Spain*, pp. 621–630 (2010)
17. Zhou, X., Gaugaz, J., Balke, W., Nejd, W.: Query Relaxation Using Malleable Schemas. In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing, China*, pp. 545–556 (2007)

Uncertainty That Counts

Dany Maslowski and Jef Wijsen

Université de Mons, Mons, Belgium
{dany.maslowski,jef.wijsen}@umons.ac.be

Abstract. Uncertainty is modeled by a multibase (\mathbf{db}, μ) where \mathbf{db} is a database with zero or more primary key violations, and μ associates a multiplicity (a positive integer) to each fact of \mathbf{db} . In data integration, the multiplicity of a fact g can indicate the number of data sources in which g was found. In planning databases, facts with the same primary key value are alternatives for each other, and the multiplicity of a fact g can denote the number of people in favor of g .

A repair of \mathbf{db} is obtained by selecting a maximal number of facts without ever selecting two distinct facts of the same relation that agree on their primary key. Every repair has a support count, which is the product of the multiplicities of its facts.

For a fixed Boolean query q , we define $\sigma\text{CERTAINTY}(q)$ as the following counting problem: Given a multibase (\mathbf{db}, μ) , determine the weighted number of repairs of \mathbf{db} that satisfy q . Here, every repair is weighted by its support count. We illustrate the practical significance of this problem by means of examples.

For conjunctive queries q without self-join, we provide a syntactic characterization of the class of queries q such that $\sigma\text{CERTAINTY}(q)$ is in \mathbf{P} ; for queries not in this class, $\sigma\text{CERTAINTY}(q)$ is $\sharp\mathbf{P}$ -hard (and hence highly intractable).

1 Motivation

Many database applications require integrating data from multiple sources. Some data sources may store inaccurate or outdated values for the same entity [7]. Users of the database often do not know which value is the correct one. This leads to uncertainty.

For example, Figure 1 shows a database resulting from the integration of three distinct source databases. Primary keys are underlined: the business rules impose that every department has a single budget and manager; every employee has a single first name, last name, and address. Unfortunately, the three source databases contained conflicting information. The *multiplicity* column μ indicates the number of source databases recording a given fact: two source databases recorded that the Toys department is managed by employee 456, while one database recorded that Toys is managed by employee 123. Two source databases recorded that Shoes is managed by employee 123; the remaining database did not store this department. Also, the source databases did not agree on the address of Ann Smith: the address 8 Corn St. was found in two databases, while

6 Main St. was found once. Luckily, the three source databases agreed on John Kipling’s address.

A *repair* of a database **db** is a maximal subset of **db** that satisfies the primary key constraints. The integrated database of Figure 1 has four repairs, because there are two choices for the manager of Toys, and two choices for the address of Ann Smith. Figure 2 shows the four repairs. Significantly, if we have equal trust (or suspicion) in each source database, then these four repairs do not have the same likelihood. The repair r_1 stores the most agreed upon values for the Toys’ manager and for Ann Smith’s address, while r_4 stores values for these items that were found only once. This can be quantified by the *support count* of a repair, which is obtained by multiplying the multiplicities of its facts. The repairs r_1 and r_4 have support counts 24 and 6, respectively. Alternatively, these numbers could be conveniently represented as a fraction of the total sum of support counts of all repairs; in this example, the support counts sum up to $54 = 24 + 12 + 12 + 6$.

DEPT	DName	Budget	Mgr	μ	EMP	E#	FName	LName	Address	μ
Toys	10K	456	2		123	Ann	Smith	8 Corn St.	2	
Toys	10K	123	1		123	Ann	Smith	6 Main St.	1	
Shoes	12K	123	2		456	John	Kipling	7 River St.	3	

Fig. 1. Integrated company database

Given a Boolean query q , we want to know the weighted number of repairs in which q evaluates to true; here, every repair is weighted by its support count. For example, the following query q_1 asks whether the manager of the Toys department lives in ‘6 Main St.’ The query q_1 is only true in r_4 with support count 6.

$$q_1 = \exists x \exists y \exists z \exists u (\text{DEPT}(\text{‘Toys’}, x, y) \wedge \text{EMP}(y, z, u, \text{‘6 Main St.’}))$$

The following query q_2 asks whether the Toys’ manager lives in ‘7 River St.’ The query q_2 is true in r_1 and r_2 , whose support counts sum up to $36 = 24 + 12$.

$$q_2 = \exists x \exists y \exists z \exists u (\text{DEPT}(\text{‘Toys’}, x, y) \wedge \text{EMP}(y, z, u, \text{‘7 River St.’}))$$

These figures could be conveniently presented as fractions of the total sum of support counts: the support fractions of q_1 and q_2 are $\frac{6}{54}$ and $\frac{36}{54}$, respectively.

Multiplicities can arise in many applications. In a conference planning database, for example, the multiplicities may indicate the number of steering committee members that are in favor of a given conference location. In the database of Figure 3, eight members are favorable to organizing FQAS 2015 in Seattle. The support counts of all repairs sum up to 100. The following query q_3 asks the support for organizing FQAS in North-America in some year:

$$q_3 = \exists x \exists y (\text{CONF}(\text{‘FQAS’}, x, y, \text{‘North-America’}))$$

DEPT	<u>DName</u>	Budget	Mgr	μ	EMP	<u>SS#</u>	FName	LName	Address	μ
	Toys	10K	456	2		123	Ann	Smith	8 Corn St.	2
	Shoes	12K	123	2		456	John	Kipling	7 River St.	3

Repair r_1 with support count $24 = 2 \times 2 \times 2 \times 3$

DEPT	<u>DName</u>	Budget	Mgr	μ	EMP	<u>SS#</u>	FName	LName	Address	μ
	Toys	10K	456	2		123	Ann	Smith	6 Main St.	1
	Shoes	12K	123	2		456	John	Kipling	7 River St.	3

Repair r_2 with support count $12 = 2 \times 2 \times 1 \times 3$

DEPT	<u>DName</u>	Budget	Mgr	μ	EMP	<u>SS#</u>	FName	LName	Address	μ
	Toys	10K	123	1		123	Ann	Smith	8 Corn St.	2
	Shoes	12K	123	2		456	John	Kipling	7 River St.	3

Repair r_3 with support count $12 = 1 \times 2 \times 2 \times 3$

DEPT	<u>DName</u>	Budget	Mgr	μ	EMP	<u>SS#</u>	FName	LName	Address	μ
	Toys	10K	123	1		123	Ann	Smith	6 Main St.	1
	Shoes	12K	123	2		456	John	Kipling	7 River St.	3

Repair r_4 with support count $6 = 1 \times 2 \times 1 \times 3$

Fig. 2. Four repairs r_1 , r_2 , r_3 , r_4 with their support counts

For the example table, there is only one repair in which FQAS is *not* organized in North-America; the support count of this repair is $18 = 2 \times 9$. Consequently, the support count for organizing FQAS in North-America is 82; the support fraction of q_3 is thus $\frac{82}{100}$.

CONF	<u>Conf</u>	<u>Year</u>	<u>Town</u>	<u>Continent</u>	μ
	FQAS	2015	Seattle	North-America	8
	FQAS	2015	Berlin	Europe	2
	FQAS	2016	Paris	Europe	9
	FQAS	2016	Dallas	North-America	1

Fig. 3. A conference planning table

Formally, for a given Boolean query q , we define $\sigma\text{CERTAINTY}(q)$ as the following problem: given a database \mathbf{db} with primary key violations and multiplicities for all facts, determine the weighted number of repairs in which q evaluates to true. In this article, we study $\sigma\text{CERTAINTY}(q)$ for queries q that belong to the class SJFCQ, which is the class of Boolean conjunctive queries that are self-join-free (that is, in which no relation name occurs more than once). Unfortunately, there are queries q in this class for which the data complexity of $\sigma\text{CERTAINTY}(q)$ is highly intractable (in particular, $\sharp\mathbf{P}$ -hard). In this article, we provide a syn-

tactic characterization of the SJFCQ queries q such that $\sigma\text{CERTAINTY}(q)$ is tractable (that is, in \mathbf{P}).

Recall that the class $\sharp\mathbf{P}$ contains the counting variant of problems in \mathbf{NP} . By Toda's theorem [12], every problem in the polynomial-time hierarchy can be solved in polynomial time given an oracle that solves a $\sharp\mathbf{P}$ -complete problem. Thus, $\sharp\mathbf{P}$ -hardness suggests a higher level of intractability than \mathbf{NP} -hardness, insofar decision problems and counting problems can be compared.

In summary, the contribution made by this article is twofold:

1. We propose to model uncertainty by primary key violations plus multiplicities. Multiplicities may be useful in practice, because they allow to model the support for a given database fact. They arise, for example, as the result of a voting process.
2. We give a sound and complete syntactic characterization of the self-join-free conjunctive queries whose data complexity is tractable.

The remainder of this article is organized as follows. Section 2 formally defines our data model and the problem of interest. We focus on Boolean conjunctive queries in which each relation name is used at most once. Section 3 discusses related work. Section 4 determines a sufficient and necessary condition, called safety \sharp , on queries q under which $\sigma\text{CERTAINTY}(q)$ is tractable. Section 5 concludes the article. Several proofs are available in a separate appendix.

2 Preliminaries

2.1 Basic Notions

Data Model We define $\mathbb{N} = \{0, 1, 2, \dots\}$. Each relation name R of arity n , $n \geq 1$, has a unique *primary key* which is a set $\{1, 2, \dots, k\}$ where $1 \leq k \leq n$. We say that R has *signature* $[n, k]$ if R has arity n and primary key $\{1, 2, \dots, k\}$. Elements of the primary key are called *primary-key positions*, while $k + 1, k + 2, \dots, n$ are *non-primary-key positions*. For all positive integers n, k such that $1 \leq k \leq n$, we assume denumerably many relation names with signature $[n, k]$.

We assume a denumerable set **dom** of *constants*, disjoint from a denumerable set **vars** of *variables*. If R is a relation name of signature $[n, k]$, and s_1, \dots, s_n are variables or constants, then $R(\underline{s_1, \dots, s_k}, s_{k+1}, \dots, s_n)$ is an R -*goal* (or simply *goal* if R is understood). Notice that primary key positions are underlined. An R -*fact* (or simply *fact*) is a goal in which no variable occurs. Two R -facts g and h are *key-equal* if they agree on all primary-key positions. Every fact is key-equal to itself.

A *database* **db** is a finite set of facts. Such database may violate primary keys, and so capture uncertainty. Given a database **db**, we write **adom**(**db**) for the set of constants that occur in **db**.

¹ This notion is unrelated to the notion of safety that guarantees domain independence in relational calculus [11, page 75].

A database \mathbf{db} is called *consistent* if it contains no two distinct, key-equal facts. A *repair* \mathbf{r} of a database \mathbf{db} is a maximal subset of \mathbf{db} that is consistent. If g is a fact of \mathbf{db} , then $\mathbf{block}(g, \mathbf{db})$ is the subset of \mathbf{db} containing each fact that is key-equal to g . The sets $\mathbf{block}(g, \mathbf{db})$ are also called *blocks*. Intuitively, repairs are obtained by choosing exactly one fact from each block.

A *multibase* is a pair (\mathbf{db}, μ) where \mathbf{db} is a database and μ is a total function $\mu : \mathbf{db} \rightarrow \mathbb{N} \setminus \{0\}$. The *support count* of a repair \mathbf{r} , denoted $\sigma(\mathbf{r}, \mathbf{db}, \mu)$, is defined by:

$$\sigma(\mathbf{r}, \mathbf{db}, \mu) = \prod \{\mu(g) \mid g \in \mathbf{r}\} .$$

For every $g \in \mathbf{db}$, we define the support count of its block as follows:

$$\sigma\mathbf{block}(g, \mathbf{db}, \mu) = \sum \{\mu(h) \mid h \in \mathbf{block}(g, \mathbf{db})\} .$$

Notice the use of \prod in the case of repairs, and \sum in the case of blocks: the facts in a repair are mutually independent, while the facts in a block are mutually exclusive.

Queries. A *Boolean conjunctive query* q is a finite set of goals. A Boolean conjunctive query $q = \{g_1, g_2, \dots, g_n\}$ represents the first-order logic sentence $\exists x_1 \dots \exists x_m (g_1 \wedge g_2 \dots \wedge g_n)$, where x_1, \dots, x_m are all variables occurring in q . Such query is *self-join-free* if it contains no two distinct goals with the same relation name. Thus, every relation name occurs at most once in a self-join-free query. The class of self-join-free Boolean conjunctive queries is denoted by SJFCQ.

Let V be a finite set of variables. A *valuation* over V is a mapping $\theta : \mathbf{vars} \cup \mathbf{dom} \rightarrow \mathbf{vars} \cup \mathbf{dom}$ such that for every $x \in V$, $\theta(x) \in \mathbf{dom}$, and for every $s \notin V$, $\theta(s) = s$. Valuations extend to goals and queries in the straightforward way.

If \mathbf{s} is a sequence of variables and constants, then $\mathbf{Vars}(\mathbf{s})$ denotes the set of variables that occur in \mathbf{s} . If g is a goal, then $\mathbf{Vars}(g)$ denotes the set of variables that occur in g , and $\mathbf{KVars}(g)$ denotes the subset of $\mathbf{Vars}(g)$ containing each variable that occurs at a primary-key position. If q is a query, then $\mathbf{Vars}(q)$ denotes the set of variables that occur in q .

If q is a conjunctive query, $x \in \mathbf{Vars}(q)$, and $a \in \mathbf{dom}$, then $q_{x \rightarrow a}$ is the query obtained from q by replacing all occurrences of x with a .

A database \mathbf{db} is said to *satisfy* Boolean conjunctive query q , denoted $\mathbf{db} \models q$, if there exists a valuation θ over $\mathbf{Vars}(q)$ such that $\theta(q) \subseteq \mathbf{db}$.

Counting Repairs. For some fixed $q \in \text{SJFCQ}$, $\sigma\text{CERTAINTY}(q)$ is the following problem: Given a multibase (\mathbf{db}, μ) , determine the total sum of support counts of repairs of \mathbf{db} that satisfy q .

Let (\mathbf{db}, μ) be a multibase and q a Boolean query. We write $\mathbf{rset}(\mathbf{db})$ for the set of repairs of \mathbf{db} , and $\mathbf{rset}(\mathbf{db}, q)$ for the subset of $\mathbf{rset}(\mathbf{db})$ containing each repair that satisfies q . Furthermore, we define:

$$\sigma\mathbf{rset}(\mathbf{db}, \mu) = \sum \{\sigma(\mathbf{r}, \mathbf{db}, \mu) \mid \mathbf{r} \in \mathbf{rset}(\mathbf{db}, q)\}$$

CONF	Conf	Year	Town	Continent	P
	FQAS	2015	Seattle	North-America	0.8
	FQAS	2015	Berlin	Europe	0.2
	FQAS	2016	Paris	Europe	0.9
	FQAS	2016	Dallas	North-America	0.1

Fig. 4. The conference planning table as a probabilistic table

$$\sigma\text{rset}(\mathbf{db}, \mu, q) = \sum \{ \sigma(\mathbf{r}, \mathbf{db}, \mu) \mid \mathbf{r} \in \text{rset}(\mathbf{db}, q) \}$$

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = \frac{\sigma\text{rset}(\mathbf{db}, \mu, q)}{\sigma\text{rset}(\mathbf{db}, \mu)}$$

Thus, for a fixed query $q \in \text{SJFCQ}$, $\sigma\text{CERTAINTY}(q)$ is the problem that takes as input a multibase (\mathbf{db}, μ) and asks to determine $\sigma\text{rset}(\mathbf{db}, \mu, q)$.

2.2 Counts Versus Fractions

We show that the choice to work with counts or fractions throughout this article is not fundamental. In [3], it is illustrated that a database with $2n$ facts can have 2^n repairs. That is, the number of repairs can be exponential in the size of the database. Nevertheless, the following lemma implies that the total sum of support counts of all repairs can be computed in polynomial time.

Lemma 1. *Let (\mathbf{db}, μ) be a multibase. Let \mathbf{r} be a repair of \mathbf{db} . Then,*

$$\sigma\text{rset}(\mathbf{db}, \mu) = \prod_{g \in \mathbf{r}} \sigma\text{block}(g, \mathbf{db}, \mu) ,$$

where, as usual, the empty product is defined to be equal to 1.

It follows that $\sigma\text{rset}(\mathbf{db}, \mu)$ can be determined in time $\mathcal{O}(n \log n)$ where n is the cardinality of \mathbf{db} : sort each relation of \mathbf{db} on its primary key values; for each block, determine the support count of that block, and multiply these numbers. Since $\sigma\text{rset}(\mathbf{db}, \mu, q) = \sigma\text{frac}(\mathbf{db}, \mu, q) \times \sigma\text{rset}(\mathbf{db}, \mu)$, if we can determine $\sigma\text{frac}(\mathbf{db}, \mu, q)$ in time $\mathcal{O}(f(n))$, then we can determine $\sigma\text{rset}(\mathbf{db}, \mu, q)$ in time $\mathcal{O}(f(n) + n \log n)$. In particular, $\sigma\text{CERTAINTY}(q)$ is in \mathbf{P} if for each multibase (\mathbf{db}, μ) , $\sigma\text{frac}(\mathbf{db}, \mu, q)$ can be determined in polynomial time in the size of (\mathbf{db}, μ) . For that reason, we can focus on determining fractions $\sigma\text{frac}(\mathbf{db}, \mu, q)$ instead of counts $\sigma\text{rset}(\mathbf{db}, \mu, q)$.

3 Related Work

The current article generalizes [10] by allowing multiplicities. The data model in [10] has no multiplicities, which is tantamount to setting $\mu(g) = 1$ for every database fact g . We believe that multiplicities are useful in many applications, as illustrated in Section [1].

Block-independent-disjoint probabilistic databases [4,5] use probabilities instead of multiplicities, as illustrated by Figure 4. If one requires that the probabilities within each block sum up to 1, then the difference between multiplicities and probabilities turns out to be irrelevant. It should be noted, however, that the authors of [4,5] do not require that the probabilities in a block sum up to 1, in which case a nonempty database can have an empty repair. This is different from our data model, in which a repair cannot be empty unless the original database is empty. This difference is significant. For example, Dalvi et al. [6,5] obtain an intractability result for the query $q = \{R(\underline{x}, y), S(\underline{y})\}$, whereas σ CERTAINTY(q) is tractable in our setting.

Our work can also be viewed as a variant of consistent query answering [2], which deals with the problem of computing answers to queries on databases that violate integrity constraints. In our data model, the only constraints are primary keys. Fuxman and Miller [8] were the first ones to focus on consistent conjunctive query answering under primary key violations. Their results have been extended and improved in recent works [13,14,15,11]. These works on primary key violations have focused on the question whether a query is true in every repair; in the current article, we ask to determine the weighted number of repairs in which the query is true.

Counting the fraction of repairs that satisfy a query is also studied by Greco et al. in [9]. The constraints in that work are functional dependencies, and the repairs are obtained by updates. Greco et al. present an approach for computing approximate probabilistic answers in polynomial time. We, on the other hand, characterize queries for which exact fractions can be obtained in polynomial time.

4 The Boundary of Tractability

We define a syntactically restricted class of SJFCQ queries, called *safe* queries, and show that for every safe SJFCQ query q , the problem σ CERTAINTY(q) is in **P**. Moreover, we show that for every unsafe SJFCQ query q , it is the case that σ CERTAINTY(q) is \sharp **P**-hard. The outline is as follows: Lemmas 2–6 first provide arithmetic expressions for $\sigma\text{frac}(\mathbf{db}, \mu, q)$ under particular syntactic conditions on q ; these lemmas are then combined in an algorithm that defines the safe queries.

The following lemma implies that we can compute in polynomial time the total sum of support counts of repairs that contain a given fact.

Lemma 2 (SE0a). *Let g be a fact. Let $q = \{g\}$, an SJFCQ query. Then, for every multibase (\mathbf{db}, μ) ,*

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = \begin{cases} 0 & \text{if } g \notin \mathbf{db} \\ \frac{\mu(g)}{\sigma\text{block}(g, \mathbf{db}, \mu)} & \text{if } g \in \mathbf{db} \end{cases}$$

Lemma 3 deals with queries that are either satisfied or falsified by all repairs. An example of such query is $q_0 = \{R(\underline{x}, x, y, z)\}$. Trivially, a database \mathbf{db} satisfies

Algorithm *IsSafe*(q)**Input:** SJFCQ query q **Output:** Boolean in $\{\mathbf{true}, \mathbf{false}\}$

1. (* SE0a *)
2. **if** $q = \{g\}$ with $\text{Vars}(g) = \emptyset$ **then return true**
- 3.
4. (* SE0b *)
5. **if** $\llbracket q \rrbracket = \emptyset$ **then return true**
- 6.
7. (* SE1 *)
8. **if** $q = q_1 \uplus q_2$ **and** $q_1 \neq \emptyset \neq q_2$ **and** $\text{Vars}(q_1) \cap \text{Vars}(q_2) = \emptyset$
9. **then return** $(\text{IsSafe}(q_1) \wedge \text{IsSafe}(q_2))$
- 10.
11. (* SE2 *)
12. **if** $\llbracket q \rrbracket \neq \emptyset$ **and** $\exists x \forall g \in \llbracket q \rrbracket (x \in \text{KVars}(g))$ **then return** $\text{IsSafe}(q_{x \mapsto a})$
13. (* a is any fixed constant *)
- 14.
15. (* SE3 *)
16. **if** $\exists x \exists g \in q (\text{KVars}(g) = \emptyset, x \in \text{Vars}(g))$ **then return** $\text{IsSafe}(q_{x \mapsto a})$
- 17.
18. (* Otherwise *)
19. **if** none of the above **then return false**

Fig. 5. Algorithm *IsSafe*

q_0 if and only if \mathbf{db} contains an R -fact of the form $R(a, a, b, c)$, for some constants a, b, c . Moreover, if \mathbf{db} contains an atom of this form, then every repair of \mathbf{db} will contain an atom of this form, and thus satisfy q_0 . Inversely, if \mathbf{db} contains no atom of this form, then no repair of \mathbf{db} will satisfy q_0 .

Definition 1. Let q be an SJFCQ query. A variable $x \in \text{Vars}(q)$ is called a liaison variable if x has at least two occurrences in q .² A variable $y \in \text{Vars}(q)$ is called an orphan variable if y occurs only once in q and the only occurrence of y is at a non-primary-key position.

The complex part of an SJFCQ query q , denoted $\llbracket q \rrbracket$, contains every goal $g \in q$ such that some non-primary-key position in g contains a liaison variable or a constant.

Example 1. Let $q = \{R(\underline{x}, y), S(\underline{y}, z), T(\underline{y}, u, a)\}$. Then, y is a liaison variable because y occurs more than once in q . The variables u and z are orphan, because they occur only once in q at a non-primary-key position. The variable x is neither liaison nor orphan. The complex part of q is $\llbracket q \rrbracket = \{R(\underline{x}, y), T(\underline{y}, u, a)\}$.

The complex part of a query q can be empty, as is the case for $q = \{R(\underline{x}, y), S(\underline{x}, u), T(\underline{y}, w)\}$, in which each position of R is a primary-key position. Lemma 3 implies that if the complex part of an SJFCQ query q is empty, then the problem $\sigma\text{CERTAINTY}(q)$ is tractable.

² Liaison variables are sometimes called join variables in the literature.

Algorithm $Eval(\mathbf{db}, \mu, q)$

Input: multibase (\mathbf{db}, μ) , safe SJFCQ query q

Output: $\sigma\text{frac}(\mathbf{db}, \mu, q)$

1. (* Evaluation SE0a *)
2. **if** $q = \{g\}$ with $\text{Vars}(g) = \emptyset$
3. **then if** $g \in \mathbf{db}$
4. **then return** $\mu(g)/\sigma\text{block}(g, \mathbf{db}, \mu)$
5. **else return** 0
- 6.
7. (* Evaluation SE0b *)
8. **if** $\llbracket q \rrbracket = \emptyset$
9. **then if** $\mathbf{db} \models q$
10. **then return** 1
11. **else return** 0
- 12.
13. (* Evaluation SE1 *)
14. **if** $q = q_1 \uplus q_2$ **and** $q_1 \neq \emptyset \neq q_2$ **and** $\text{Vars}(q_1) \cap \text{Vars}(q_2) = \emptyset$
15. **then return** $Eval(\mathbf{db}, \mu, q_1) \times Eval(\mathbf{db}, \mu, q_2)$
- 16.
17. (* Evaluation SE2 *)
18. **if** $\llbracket q \rrbracket \neq \emptyset$ **and** $\exists x \forall g \in \llbracket q \rrbracket (x \in \text{KVars}(g))$
19. **then return** $1 - \left(\prod_{a \in \text{adom}(\mathbf{db})} (1 - Eval(\mathbf{db}, \mu, q_{x \mapsto a})) \right)$
- 20.
21. (* Evaluation SE3 *)
22. **if** $\exists x \exists g \in q (\text{KVars}(g) = \emptyset, x \in \text{Vars}(g))$
23. **then return** $\sum_{a \in \text{adom}(\mathbf{db})} Eval(\mathbf{db}, \mu, q_{x \mapsto a})$

Fig. 6. Algorithm $Eval$

Lemma 3 (SE0b). *Let q be an SJFCQ query. If $\llbracket q \rrbracket = \emptyset$, then for every multibase (\mathbf{db}, μ) ,*

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = \begin{cases} 0 & \text{if } \mathbf{db} \not\models q \\ 1 & \text{if } \mathbf{db} \models q \end{cases}$$

Proof. Straightforward.

Lemma 4 deals with queries q that can be partitioned into two subqueries, say q_1 and q_2 , such that q_1 and q_2 have no variables in common. The lemma implies that if $\sigma\text{CERTAINTY}(q_1)$ and $\sigma\text{CERTAINTY}(q_2)$ are both tractable, then so is $\sigma\text{CERTAINTY}(q)$.

Lemma 4 (SE1). *Let q, q_1, q_2 be SJFCQ queries such that $q = q_1 \cup q_2$, $q_1 \cap q_2 = \emptyset$, and $\text{Vars}(q_1) \cap \text{Vars}(q_2) = \emptyset$. Then, for every multibase (\mathbf{db}, μ) ,*

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = \sigma\text{frac}(\mathbf{db}, \mu, q_1) \times \sigma\text{frac}(\mathbf{db}, \mu, q_2).$$

Lemma 5 treats queries q for which there exists a variable x such that x occurs at a primary-key position in every goal of the complex part of q . An example is the query $\{R(\underline{x}, y), S(\underline{x}, y, z), T(\underline{z}, u)\}$, whose complex part is $\{R(\underline{x}, y)$,

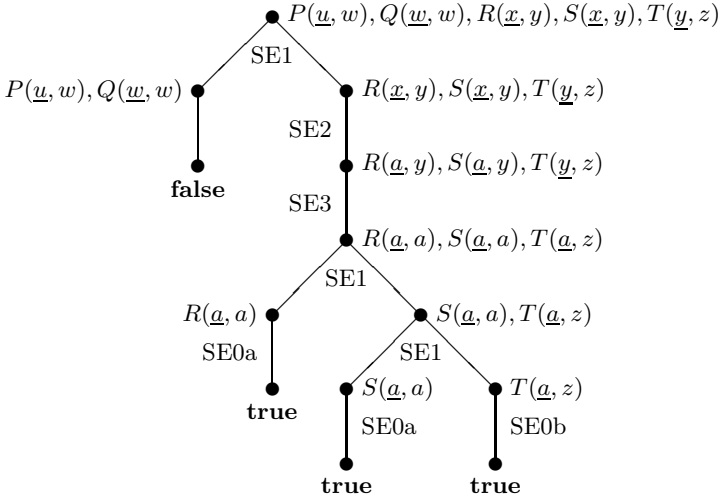


Fig. 7. Tree representation of an execution of `IsSafe`. Vertices are labeled by queries. Each edge label indicates the rule that, applied on the parent, results in the children. Since some leaf vertex is **false**, the query at the root node is unsafe. Notice that the subquery $R(\underline{x}, y), S(\underline{x}, y), T(\underline{y}, z)$ is safe, because all its descendant leaf vertices are **true**.

$S(x, y, z)$ }; the variable x occurs at a primary-key position in each goal of the complex part.

Lemma 5 (SE2). *Let q be an SJFCQ query such that $\llbracket q \rrbracket \neq \emptyset$ and for some variable x , $x \in \bigcap_{g \in \llbracket q \rrbracket} \text{KVars}(g)$. Then, for every multibase (\mathbf{db}, μ) ,*

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = 1 - \left(\prod_{a \in \text{adom}(\mathbf{db})} (1 - \sigma\text{frac}(\mathbf{db}, \mu, q_{x \mapsto a})) \right).$$

Lemma 6 treats queries with a goal g such that all primary-key positions in g are occupied by constants, and at least one non-primary-key position is occupied by a variable.

Lemma 6 (SE3). *Let q be an SJFCQ query such that for some goal $g \in q$, for some variable x , $\text{KVars}(g) = \emptyset$ and $x \in \text{Vars}(g)$. Then, for every multibase (\mathbf{db}, μ) ,*

$$\sigma\text{frac}(\mathbf{db}, \mu, q) = \sum_{a \in \text{adom}(\mathbf{db})} \sigma\text{frac}(\mathbf{db}, \mu, q_{x \mapsto a}).$$

Lemmas 2-6 are now bundled in a recursive algorithm, called `IsSafe`, which takes as input an SJFCQ query q and returns **true** if for every multibase (\mathbf{db}, μ) ,

$\sigma\text{frac}(\mathbf{db}, \mu, q)$ can be obtained by recursive application of Lemmas 2–6; otherwise `IsSafe` returns **false**. Algorithm `IsSafe` is shown in Figure 5. To simplify the notation, we write $A \uplus B$ for the *disjoint union* of sets A and B , where A and B are understood to be disjoint. An execution of `IsSafe` is illustrated in Figure 7. Algorithm `IsSafe` always terminates since every recursive call has an argument query that contains either less goals or less variables.

Definition 2. An SJFCQ query q is called *safe* if the algorithm `IsSafe` returns **true** on input q ; if `IsSafe` returns **false**, then q is called *unsafe*.

Figure 6 shows algorithm `Eval`, which takes as input a multibase (\mathbf{db}, μ) and a safe SJFCQ query q , and returns $\sigma\text{frac}(\mathbf{db}, \mu, q)$ in polynomial time in the size of \mathbf{db} .

Theorem 1. *If q is a safe SJFCQ query, then*

1. *for each multibase (\mathbf{db}, μ) , algorithm `Eval` correctly computes $\sigma\text{frac}(\mathbf{db}, \mu, q)$;*
2. *$\sigma\text{CERTAINTY}(q)$ is in \mathbf{P} .*

Finally, we show that $\sigma\text{CERTAINTY}(q)$ is $\sharp\mathbf{P}$ -hard (and hence highly intractable) for queries q that violate safety.

Theorem 2. *For every unsafe SJFCQ query q , the problem $\sigma\text{CERTAINTY}(q)$ is $\sharp\mathbf{P}$ -hard under polynomial-time Turing reductions.*

Proof. From [10], it follows that $\sigma\text{CERTAINTY}(q)$ is already $\sharp\mathbf{P}$ -hard if all multiplicities are 1.

Consequently, given an SJFCQ query q , we can test by means of algorithm `IsSafe` whether q is safe. If q is not safe, then computing $\sigma\text{frac}(\mathbf{db}, \mu, q)$ for a given database (\mathbf{db}, μ) has highly intractable data complexity, unless $\mathbf{P} = \mathbf{NP}$. On the other hand, if q is safe, then algorithm `Eval` computes $\sigma\text{frac}(\mathbf{db}, \mu, q)$ in polynomial time in the size of (\mathbf{db}, μ) .

5 Conclusion

Uncertainty is modeled in our data model by means of primary key violations. Each database fact g has a multiplicity, indicating how often g occurs in the database. We have argued that multiplicities may be useful in practice.

For a Boolean query q , the problem $\sigma\text{CERTAINTY}(q)$ asks the weighted number of repairs (or possible worlds) in which q evaluates to true. Intuitively, a query q is more certain if it is true in more repairs. We have given a sound and complete syntactic characterization of the self-join-free conjunctive queries q for which $\sigma\text{CERTAINTY}(q)$ can be solved in polynomial time (data complexity).

An open problem is to determine tractability boundaries for richer query languages, like unions of conjunctive queries or conjunctive queries with self-joins.

References

1. Abiteboul, S., Hull, R., Vianu, V.: *Foundations of Databases*. Addison-Wesley, Reading (1995)
2. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: PODS, pp. 68–79. ACM Press, New York (1999)
3. Arenas, M., Bertossi, L.E., Chomicki, J., He, X., Raghavan, V., Spinrad, J.: Scalar aggregation in inconsistent databases. *Theor. Comput. Sci.* 296(3), 405–434 (2003)
4. Dalvi, N.N., Ré, C., Suciu, D.: Probabilistic databases: diamonds in the dirt. *Commun. ACM* 52(7), 86–94 (2009)
5. Dalvi, N.N., Re, C., Suciu, D.: Queries and materialized views on probabilistic databases. *J. Comput. Syst. Sci.* 77(3), 473–490 (2011)
6. Dalvi, N.N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: Libkin, L. (ed.) PODS, pp. 1–12. ACM, New York (2007)
7. Fan, W., Geerts, F., Wijsen, J.: Determining the currency of data. In: Lenzerini, M., Schwentick, T. (eds.) PODS, pp. 71–82. ACM, New York (2011)
8. Fuxman, A., Miller, R.J.: First-order query rewriting for inconsistent databases. *J. Comput. Syst. Sci.* 73(4), 610–635 (2007)
9. Greco, S., Molinaro, C.: Approximate probabilistic query answering over inconsistent databases. In: Li, Q., Spaccapetra, S., Yu, E.S.K., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 311–325. Springer, Heidelberg (2008)
10. Maslowski, D., Wijsen, J.: On counting database repairs. In: *Proceedings of the 4th International Workshop on Logic in Databases, LID 2011*, pp. 15–22. ACM, New York (2011), <http://doi.acm.org/10.1145/1966357.1966361>
11. Pema, E., Kolaitis, P.G., Tan, W.C.: On the tractability and intractability of consistent conjunctive query answering. In: *Proceedings of the 2011 Joint EDBT/ICDT Ph.D. Workshop, PhD 2011*, pp. 38–44. ACM, New York (2011), <http://doi.acm.org/10.1145/1966874.1966881>
12. Toda, S.: PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.* 20(5), 865–877 (1991)
13. Wijsen, J.: On the consistent rewriting of conjunctive queries under primary key constraints. *Inf. Syst.* 34(7), 578–601 (2009)
14. Wijsen, J.: On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In: Paredaens, J., Gucht, D.V. (eds.) PODS, pp. 179–190. ACM, New York (2010)
15. Wijsen, J.: A remark on the complexity of consistent conjunctive query answering under primary key violations. *Inf. Process. Lett.* 110(21), 950–955 (2010)

Structured Data-Based Q&A System Using Surface Patterns

Nicolas Kuchmann-Beauger^{1,2} and Marie-Aude Aufaure²

¹ SAP Research, 157/159 rue Anatole France, 92309 Levallois-Perret, France

² Ecole Centrale Paris, MAS laboratory, 92290 Chatenay-Malabry, France
{marie-aude.aufaure,nicolas.kuchmann-beauger}@ecp.fr

Abstract. Question Answering (Q&A) systems, unlike other Information Retrieval (IR) systems, aim at providing directly the answer to the user, and not a list of documents in which the correct answer may be found. Our system is based on a data warehouse and provides composite answers made of a dataset and the corresponding chart visualizations. The question translation step is based on a new proposal for surface patterns that incorporate business semantic as well as domain-specific knowledge allowing a better coverage of questions.

Keywords: Question Answering, Information Extraction, linguistic patterns.

1 Introduction

Question Answering (Q&A) systems aim at providing one answer to a user's question. The major difference from other Information Retrieval (IR) systems is that the result is not a list of documents where the correct answer has to be found, but the answer itself [7]. Surveys on large-scale search services [9] which represent IR systems show that users struggle to formulate queries: the average query is being reformulated 2,6 times, but this average query is concise: it contains 3,5 tokens. The context of Q&A is quite different, because the user expects from the system the exact answer to appear and in this case keywords are not enough to express complex information needs. The fact that Q&A systems return concise answers and not whole pieces of documents, require from Q&A techniques a deeper understanding of document content [2]. Communities concerned by this field of research are Natural Language Processing (NLP), Machine Learning (ML), and more generally IR and Artificial Intelligence (AI).

Most common systems are based on unstructured data, especially web documents. Our context is quite different, because we focus on structured data in warehouses. Typical users are employees of a company who want to query and analyze those data; these users generally want to have a quick overview of the data, but do not always exactly know how to express such queries, because the syntax of the technical query (e.g. SQL or MDX) is not that easy to employ.

Questions of these users are data-oriented (the expected answer is a table of values and an associated visualization) whereas questions in traditional Q&A

are often factual questions or explanatory questions (the expected answer is a sentence or a phrase expressing the answer).

The answer that we consider is not a sentence as it is the case in most Q&A systems, but rather a composite answer. For example, the question “Detail the sales in the US and compare it with France” would return in our context tables of values, charts showing specific comparisons between data and possibly recommendation of reports composed of relevant queries, as opposed to a well-formed sentence (which would be the case in a traditional Q&A system based on unstructured data). Open questions, like “Why are we not going well?” is not the scope of our work.

Most people familiar with IR tools are used to express queries using keywords, because most popular IR systems like search engines are based on the assumption that queries are composed of keywords. However, there are several concerns about this interaction. Simple queries could be expressed using an ordered list of terms, but not complex ones. Table 1 illustrates examples of complex BI queries. The proposed system allows as input queries expressed in Natural Language, but

Table 1. Examples of complex BI queries from different fields

Field	Question
Acquisition	Can I measure if my marketing campaign was effective?
Attendance	What effect has the campaign had on attendance?
Referral	How many of the referrals were existing customers?
Discount analysis	What is the correct price for my products?

it remains possible to interact with the system using traditional keywords.

We have addressed the core matter of answering BI questions, and existing proposals from the IR or Q&A communities do not satisfy these specific needs. In particular, we hope getting better flexibility on handling BI question using our new linguistic pattern definition that do not rely on the classic hypothesis of syntactic isomorphy (see section 3).

The rest of this paper is structured as follows. Section 2 presents the related work in this field. Section 3 deals with adopted definitions and choices for representing linguistic patterns. Section 4 presents the architecture of the system and give details of the implementation. The evaluation criteria and our experiments are discussed in section 5. The conclusion and future work are presented section 6.

2 Related Work

Q&A is one of the first applications of Artificial Intelligence. Its goal is to answer questions expressed in Natural Language (NL). First Q&A systems were based on data structured in databases [4,17] but the great majority of such systems look

for answers in textual documents, because of the huge availability of unstructured documents, especially on internet.

Q&A systems focus on different strategies in order to map user’s question to one answer, whether extracted from text corpora or retrieved in databases. Andrenucci and Sneider [1] address the main research approaches related to Q&A: NLP,IR and template-based approaches.

Pattern-based approaches (also called template-based approaches) are popular in this field, because they lead to good results (the TREC-10 winner used only a list of surface patterns as external resource). One issue is the representation of such patterns. Sung et al. [16] distinguishes between patterns that do not represent any semantic and patterns that retrieve semantic relationships among terms (called semantic patterns). Question patterns are usually associated to answer patterns (that locate the answer in textual documents). In this case, fine-grained answer typing is important in order to get precise answers. Patterns defined by Soubbotin [15] are rich patterns composed of predefined string sequences as well as unordered combination of strings and definition patterns. Much work has also been done in the pattern learning area: Saiz-Noeda et al. [12] propose a learning approach based on the maximum entropy, and apply this to anaphora resolution. Ravichandran et Hovy [10] propose a method to learn automatically new patterns for unstructured data-based Q&A, and propose to exploit answers from the Web as well (accurate answers are returned by Web search services in the top positions).

3 Linguistic Patterns, Definitions and Hypothesis

Morpho-syntactic patterns are extensively used, but there are very few comments on the definition of such patterns. Such linguistic patterns have been defined in the linguistic theory [5] as “a schematic representation like a mathematical formula using terms or symbols to indicate categories that can be filled by specific morphemes”. Patterns used by Sneider [13] are regular strings of characters where sets of successive tokens are replaced by entity slots (to be filled by corresponding terms in the real text). An innovation in [14] is the definition of a pattern being composed of two subpatterns: one required pattern (regular patterns) and one forbidden patterns, that corresponds to pattern that must not match the message. Finkelstein-Landau et Morin [3] define formally morpho-syntactic patterns related to their Information Extraction (IE) task: they aim at extracting semantic relationship from textual documents. The definition is displayed formula [1].

$$A = A_1 \dots A_i \dots A_j \dots A_n \quad (1)$$

In this formula, A_k $k \in [1, n]$ denotes an *item* of the pattern which is a part of a text (no constraint *a priori* on the sentence boundaries). An *item* is defined as an ordered set of *tokens*, which compose words[1]. In this approach the syntactic isomorphy hypothesis is adopted.

¹ Delimiting tokens is not an easy task in any language.

$$B = B_1 \dots B'_i \dots B'_j \dots B'_n \quad (2)$$

This hypothesis states the following assertion:

$$\left. \begin{array}{l} \exists(i, j) \quad \text{win}(A_1, \dots, A_{i-1}) = \text{win}(B_1, \dots, B_{j-1}) \\ \text{win}(A_{i+1}, \dots, A_{i+1}) = \text{win}(B_{j+1}, \dots, B_{j+1}) \end{array} \right\} \implies A_i \sim B_j \quad (3)$$

which means that if two patterns A and B are *equivalent* (they denote the same string of characters), and if it is possible to split both patterns in identical *windows* composed of the same tokens when applied to string of characters, then the remaining items of both patterns (A_i and B_j) share the same syntactic function.

We propose another formulation of patterns which are composite patterns, and we do not rely on the syntactic isomorphy. The categories used to describe our patterns are tokens themselves (TOKEN), part-of-speech of tokens (POS), wh-question words (WHQ, extensively used in our context) stems of tokens (LEMMA), terms related to a known concept in our domain ontology (ENTITY), objects defined in the data model of the underlying Data Warehouse (SL(DIM), SL(MEA) or SL(MEM)), or references to existing patterns (PATTERN). We also allow the representation of the syntactic relation related to the syntactic hypothesis, which is the underlying syntagmatic order, and semantic relationship defined in the domain ontology. An other feature is the possibility to specify token references, which means that one token may be represented in the same pattern by more than one category, which is not possible in classical morpho-syntactic patterns. In addition, we also use the classical wildcards to specify cardinalities. We have defined a grammar to parse patterns.

The benefit of using our formulation is that the same pattern matches other formulations, such as “For 2011, what are the sales revenue?”. Figure 1 displays an example of the parse tree of this pattern.

4 Architecture and Implementation Details

The proposed system interacts with a Data Warehouse (DW) through an abstraction layer (called Semantic Layer) on which queries are expressed regardless of the data connection. Technical queries are composed of objects from the layer, and aggregations, automatically computed, do not need to be expressed.

We present the architecture of the proposal figure 2.

We will focus on each components, Question Processing, Pattern Matching, Answer Processing and Answer Federation, but we will insist on the second one (Pattern Matching) which presents the main novelty of this paper.

4.1 Question Processing

The Question Processing component aims at analyzing the user’s question. We use shallow NLP techniques to avoid time consuming processing. Our approach

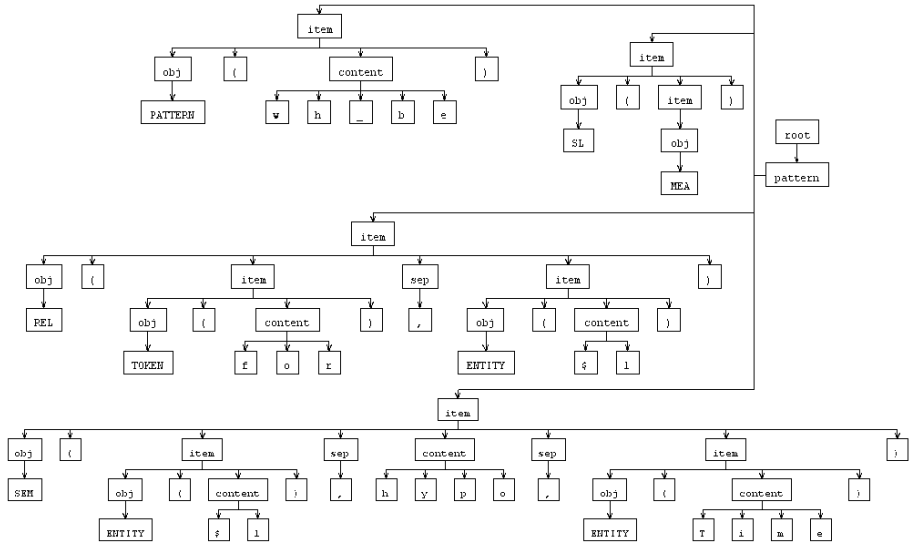


Fig. 1. Parse tree of the pattern corresponding to the question “What are the sales revenue for Q1?” generated by ANTLR [8]

is based on linguistic patterns which compose the general-domain knowledge. Our assumption is that using a few patterns will be sufficient in most cases.

When a new question is submitted to the system and if the user hasn’t specified the question language, it is analyzed. Then, the question is tokenized according to language rules defined in the SAP TextAnalysis language recognition tool. The NER identifies named entities in the user’s question, including business entities. Additional knowledge is composed of a set of English question patterns that are matched against users’ questions.

Technical queries that are associated to these initial patterns are used to produce the graphs representing the queries. These graphs are then used by the Answer Processing component to produce potential candidate answers. The last component of the system will be used to properly display the answers to the user (raw data and/or best associated visualization).

4.2 Pattern Matching

This component analyzes the parsed user’s input and retrieves similar patterns. The Pattern Learning approach is not fully implemented yet, but its goal is to build new patterns from users’ parsed input when no similar existing pattern can be retrieved (this occurs when the most similar existing pattern presents a low similarity according to a custom threshold, in which case these similar patterns are not considered similar enough).

Consider the following pattern, which is part of the predefined patterns:

```
REL(WHQ, LEMMA(be)) SL(MEA) REL(TOKEN(in), ENTITY($1))
SEM(ENTITY($1), hypo, ENTITY(Place))
```

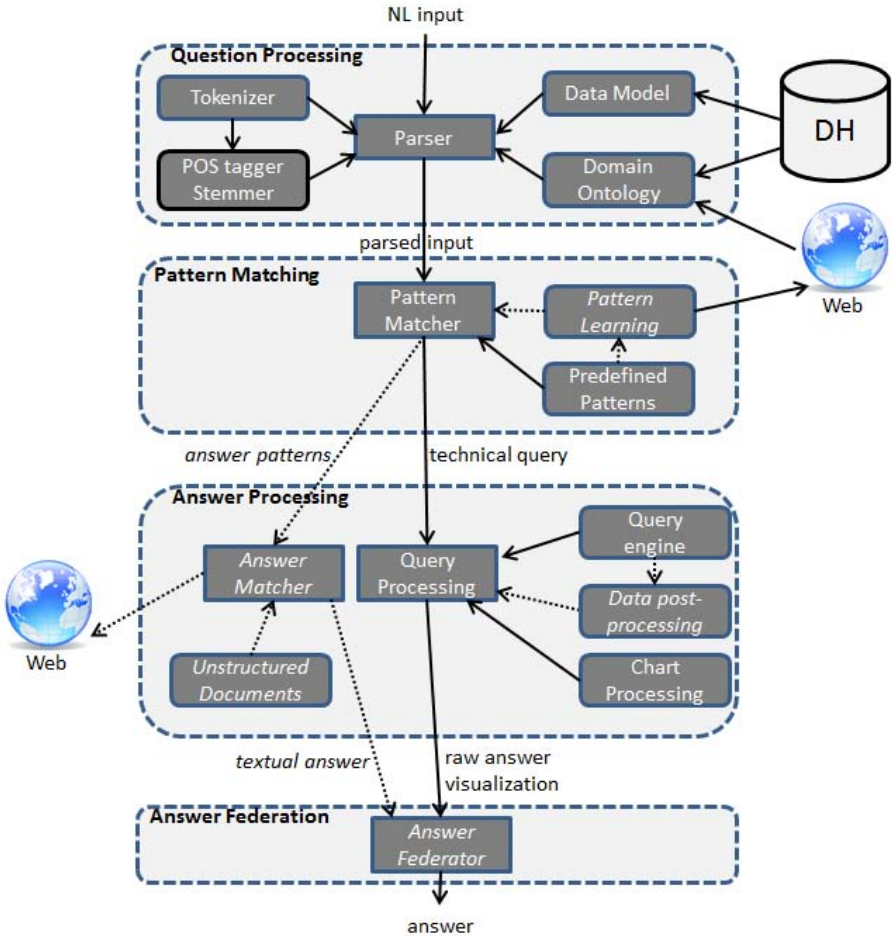


Fig. 2. Architecture of the proposal

One remarks that the order of the tokens in such patterns do not have any impact on the patterns themselves, the syntagmatic order being specified by the keyword REL. The keyword SEM indicates a semantic relationship defined in our domain ontology, and the identity constraint is specified by the keyword \$1 in this example. One associated initial question may be “What are the sales revenue in North America?”. The parser produces a set of items, and the exact matching algorithm [1] instantiates the technical query associated to the pattern, and returns the set of associated answer patterns to be used for searching answers in unstructured documents. The subfunctions are explained below:

- *getReachableItems* retruns user items that appear *after* the considered item according to the position of each item (item position in the user’s question) and the length of each item (the number of tokens that compose the item)

Algorithm 1. Exact pattern matching

```

var potentialQueries : Array = {}
for item ∈ userItems.getItems() do
  reachableItems ← item.getReachableItems()
  item.cardinality ← countSameItems(item, reachableItems)
end for
for pattern ∈ patterns do
  pattern.updatedReferences()
  var found : boolean = true
  for item ∈ pattern.getItems() do
    if ¬userItems.contains(item) then
      found ← false
    end if
  end for
  if found then
    potentialQueries.add(pattern.getQuery())
  end if
end for

```

- *countSameItems* counts the number of identical items that appear after the considered item in the parsed user’s question
- *updateReferences* replaces references of sub-patterns by the sub-patterns themselves, and links the items that make a reference to each other (which concerns items containing one $\$i$ ($i \in \mathbb{N}^+$) argument.
- *contains* is the matcher sub-function itself. It takes into account the type of the item, the name and arguments of the item (depending of the type of the item), the cardinality of both user and pattern items and the reference constraint if applicable

When no exact matching pattern is available, most similar patterns are considered and we made the assumption that the similarity measure should not consider that every token types are equivalent. We proposed the order displayed in table 2. This order corresponds to a weight in the similarity measure, that will lead to an evaluation to validate those weights.

In our context, the most similar pattern selection can be seen as a maximization problem where we try to maximize the number of features (tokens) from the parsed user’s question that also belong to the candidate pattern. We consider the following problem 4:

$$\begin{aligned}
 \max \sum_i w_{t(t_i)} t_i \in t \subset \mathcal{T} \\
 |t_i| < n \quad t_i \in t \\
 \sum_i w_{t(t_i)} \leq 1
 \end{aligned} \tag{4}$$

where $t(t_i)$ denotes the type of the i th token in the candidate pattern, $w(k)$ the weight associated to the token type k , t the set of tokens that forms the candidate pattern, n the length of the user’s question and \mathcal{T} the set of possible tokens in patterns.

Table 2. Weight order of token types when comparing patterns

Order	Token type
1	SL (MEA)
2	SL (DIM)
3	SL (MEM)
4	REL
5	SEM
6	ENTITY
7	LEMMA
8	POS
9	TOKEN

This allows us to match user’s question to predefined patterns, even if exact matching is not possible. Moreover, this formulation seems more accurate than a classic similarity measure based on the distance between tokens of the potential pattern and the tokens of the predefined patterns, because in our context we do not want to rank patterns, we aim at selecting one most similar pattern. An other explanation for this choice, is that such measures as described in [11] rely on the edit distance measure, which is based on the assumption that the considered linguistic patterns share the syntagmatic order, which is not our hypothesis.

4.3 Answer Processing

The picture 3 is an example of the answer provided by the prototype that corresponds to the question “What are the sales revenue in New York and in Texas?”. The language is automatically identified (*English*) and the answer is composed of the raw values (a table) and the visualization of the answer. The “input interpretation” corresponds to the linguistic pattern that has been selected from the user’s input.

Two components are at the moment fully implemented: the incorporation of the results of the query engine, and the chart processing. The former consists in invoking a query service using objects defined in the data model (from the data abstract layer), and interpreting the XML-output, and the latter returns a vector or binary image from the raw data.

The data post-processing has been partly implemented, but we believe this feature is a vital in the context of structured-based Q&A.

4.4 Answer Federation

The Answer Federation component merges answers from different sources: answer from the Data Warehouse, and the answer from other unstructured or semi-structured documents. As an example, BI reports are analyzed to identify content relevant to the user’s query, and the provided information is compared to the answer.

If relevant, those reports are then suggested in a recommender approach: users may be interested in navigating documents (containing more general information

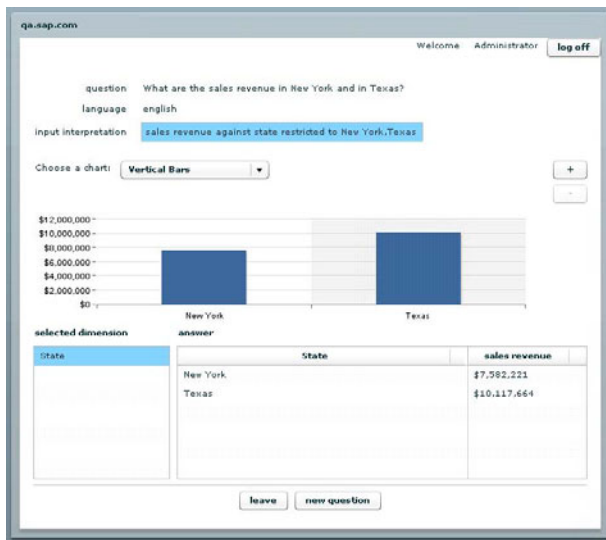


Fig. 3. Answer provided by the prototype

than the answer to the user's initial query), but our approach is to propose those documents in the end, as a complementary source of information. We propose the exact answer first, and encourage then users to explore the data and the information to satisfy better the information need.

5 Evaluation Criteria and Experiments

Q&A systems have been studied for decades, and numerous evaluation scenarios have been proposed. We will discuss the scenarios applicable to our system on the one hand, and the results on the second hand.

5.1 Evaluation Criteria and Scenarios

When evaluating a whole system, different scenarios are possible:

1. evaluation of the system globally (*black box evaluation*): how does the system globally perform compared to an assumed ground truth?
2. evaluation of each sub-component (*white box evaluation*): from one input specifications and output specification, how well does the component proceed?

In the real life, things are not that simple. One huge restriction, is the lack of comparable systems: one should compete the system with other systems based on structured data and dedicated to BI questions; however, to the best of our knowledge, there is no competition comparable to TREC for open-domain questions for example.

This leads to two forms of evaluation, that sound applicable in our context:

- evaluation of the users’ satisfaction
- evaluation of each component

Evaluation users’ satisfaction may be performed by directly asking to the user, but feedback from user experience show that users are not willing to waste time giving their opinion on the usability of the systems they have used. An other option consists in analyzing users’ interaction with the system, which is an entire research area [6] and not the scope of the present paper.

5.2 Experiments

In order to mimit a real use of the system, we selected randomly 100 BI questions written by experts and linked with the DW, which contains data about sales of clothes in different stores.

The results are displayed table 3. The first line “no answer” correspond to

Table 3. Results of the experiment

Kind	Result
No answer	6%
Already existing pattern	20%
New pattern defined	74%

very complex questions that cannot be answered yet, because we do not reach the required analysis level for these questions. Example of such questions are: “Which products have the largest sales changes since last period?” or “What is the total revenue change attributable to the 10 biggest revenue growers and decliners between 2004 and 2005?”. The second category is made of questions that did not require any new pattern definition. The last category is composed of the remaining of questions, that required the definition of new patterns. One assumption presented in section 3 states that if several patterns are applicable to one question, the longest pattern is taken into consideration according to the weight order depending on the item types (see table 2). The situation where this assumption lead to a wrong question analysis has never been met in our experiment.

6 Conclusion and Future Work

We have implemented a Q&A system able to answer BI-questions expressed in NL or using keywords on data warehouses. The original proposal on pattern formulation leads to a better coverage of users’ questions. The system does not need any setup effort. Shallow linguistic techniques that we use allow us to get a better understanding of the users’ need.

We believe one major improvement will be the ability to handle unstructured and semi-structured documents, such as documents present in enterprise intranets, or documents located in users' repositories, such as BI reports.

The approach we are willing to adopt, is case-based reasoning in the context of pattern learning. This approach will learn automatically new linguistic patterns from users' input. Taking into account the context is also a major topic in our work; the considered context is the user-centered context and the global preferences and security roles that will be defined. The follow-up questions feature may also improve our results, since users may want to make reference to previous questions, and because constraints on follow-up questions may be defined. An other interesting improvement will be the generation of a textual summary of the answer using the domain ontology.

References

1. Andrenucci, A., Sneiders, E.: Automated question answering: Review of the main approaches. In: Proceedings of the Third International Conference on Information Technology and Applications (ICITA 2005), vol. 2, pp. 514–519. IEEE Computer Society, Washington, DC, USA (2005), <http://dx.doi.org/10.1109/ICITA.2005.78>
2. Ferrández, A., Peral, J.: The benefits of the interaction between data warehouses and question answering. In: Daniel, F., Delcambre, L.M.L., Fotouhi, F., Garrigós, I., Guerrini, G., Mazón, J.N., Mesiti, M., Müller-Feuerstein, S., Trujillo, J., Truta, T.M., Volz, B., Waller, E., Xiong, L., Zimányi, E. (eds.) EDBT/ICDT Workshops. ACM International Conference Proceeding Series. ACM, New York (2010)
3. Finkelstein-landau, M., Morin, E.: Extracting semantic relationships between terms: Supervised vs. unsupervised methods. In: Proceedings of International Workshop on Ontological Engineering on the Global Information Infrastructure, pp. 71–80 (1999)
4. Green, B., Wolf, A., Chomsky, C., Laughery, K.: Baseball: an automatic question answerer, pp. 545–549 (1986)
5. Hayes, B., Curtiss, S., Szabolcsi, A., Stowell, T., Stabler, E., Sportiche, D., Koopman, H., Keating, P., Munro, P., Hyams, N., Steriade, D.: Linguistics: An Introduction to Linguistic Theory. Wiley-Blackwell (February 2001)
6. Joachims, T.: Optimizing search engines using clickthrough data. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 133–142 (2002)
7. Moldovan, D., Surdeanu, M.: On the role of information retrieval and information extraction in question answering systems. In: Pazienza, M.T. (ed.) SCIE 2003. LNCS (LNAI), vol. 2700, pp. 129–147. Springer, Heidelberg (2003)
8. Parr, T.J., Quong, R.W.: Antlr: A predicated- $ll(k)$ parser generator. *Softw., Pract. Exper.* 25(7), 789–810 (1995)
9. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Jia, X. (ed.) Infocale. ACM International Conference Proceeding Series, vol. 152, p. 1. ACM, New York (2006)
10. Ravichandran, D., Hovy, E.H.: Learning surface text patterns for a question answering system. In: ACL, pp. 41–47 (2002)

11. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Data Knowl. Eng.* 61, 484–499 (2007), <http://portal.acm.org/citation.cfm?id=1238147.1238395>
12. Saiz-Noeda, M., Suárez, A., Palomar, M.: Semantic pattern learning through maximum entropy-based wsd technique. In: *Proceedings of the 2001 Workshop on Computational Natural Language Learning, ConLL 2001*, vol. 7. Association for Computational Linguistics, Stroudsburg (2001), <http://dx.doi.org/10.3115/1117822.1455624>
13. Sneiders, E.: *Automated Question Answering: Template-based Approach*. Ph.D. thesis, Royal Institute of Technology, Sweden (2002)
14. Sneiders, E.: Automated email answering by text pattern matching. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) *IceTAL 2010*. LNCS, vol. 6233, pp. 381–392. Springer, Heidelberg (2010)
15. Soubbotin, M.M.: Patterns of potential answer expressions as clues to the right answers. In: *TREC (2001)*
16. Sung, C.L., Lee, C.W., Yen, H.C., Hsu, W.L.: An alignment-based surface pattern for a question answering system. In: *IRI*, pp. 172–177. IEEE Systems, Man, and Cybernetics Society (2008)
17. Woods, W.A.: Progress in natural language understanding: an application to lunar geology. In: *AFIPS 1973: Proceedings of the National Computer Conference and Exposition, June 4-8*, pp. 441–450. ACM, New York (1973)

Higher Reasoning with Level-2 Fuzzy Regions

Jörg Verstraete^{1,2}

¹ Systems Research Institute - Polish Academy of Sciences
Ul. Newelska 6, 01-447 Warszawa, Poland
jorg.verstraete@ibspan.waw.pl

<http://www.ibspan.waw.pl>

² Database, Document and Content Management - Department of
Telecommunications and Information Processing
Sint Pietersnieuwstraat 41, 9000 Gent, Belgium
jorg.verstraete@telin.ugent.be
<http://telin.ugent.be/ddcm>

Abstract. Spatial data is quite often is prone to uncertainty and imprecision. For this purpose, fuzzy regions have been developed: they basically consist of a fuzzy set over a two dimensional domain, allowing for both fuzzy regions and fuzzy points to be modelled. The model is extended to a level-2 fuzzy region to overcome some limitations, but this has an impact on operations. In this contribution, we will look into the construction of and combination of existing data to yield level-2 fuzzy regions.

Keywords: level-2 fuzzy set, level-2 fuzzy region, fuzzy spatial.

1 Introduction

A lot of data that is currently used involves some spatial content, and unfortunately this data often is prone to uncertainty and or imprecision: the origin for this imperfection can be the data itself (the real life data is inherently imperfect), limitations in measurements (the real life data is perfect, but it is impossible or too expensive to measure it accurately) or the combination of data (data from different sources can contradict or be incompatible). Applying flexible querying on crisp spatial data can also result in uncertain or imprecise results: a user can query for locations *at walking distance*, locations that involve *steep* inclines or locations where it is likely to see a specific animal or plant. Especially in the case where the data in the database is uncertain or imprecise, but even in the case where the data is crisp and flexible querying is permitted on spatial data, it is necessary to consider data structures capable of handling uncertain and imprecise spatial data. This contribution concerns the further development of the mathematical foundation of a spatial data structure capable of handling uncertainty and imprecision in spatial data. The structure is based on a simpler model from which implementable models have been derived.

Traditionally, there are two big types of models used for spatial data: entity based and field based. In an entity based approach, basic geometrical shapes are

used to represent features: lines represent roads (or sides of the road); polygons can represent buildings or pieces of land. The field based approach allows numerical data to be represented over a region over interest, and is commonly used for e.g. pollution data: a grid is overlaid with the map, and with each cell of the grid a value is associated; this value is deemed representative for the area covered by the cell.

When representing fuzzy spatial data in a database, it is not only necessary to have the adequate operators to combine different data as dictated by the theory, but it is also necessary to provide operators to construct data from components. These operations can be necessary when generating the data to be stored in the database, but also when answering queries to generate the results. In this contribution an entity based approach, where features are subject to imperfection: their outline or location (or both) can be uncertain or imprecise. For the representation of uncertain or imprecise spatial features, fuzzy regions have been developed; these fuzzy regions are basically a fuzzy set over a two dimensional domain; a veristic interpretation allows us to represent regions with an imperfect outline, a possibilistic interpretation will allow the representation of an uncertain point. This model has been extended to level-2 fuzzy regions to allow for the modelling of uncertainty and imprecision in a single unified model and to allow both interpretations at the same time. For this level-2 representation, a number of operators have been developed; in this contribution we will consider higher level operations necessary to construct and reason with these models. Section 2 explains the fuzzy regions; in section 3 the extension to level-2 fuzzy regions is explained. Section 4 concerns the main topic of this paper and explains which operations can be used to construct and reason with level-2 fuzzy regions. The conclusion summarizes the findings.

2 Fuzzy Regions

When modelling real world entities, it is frequent for the real world entity not to have an accurate or certain definition. Examples of this can be the spread of a pollutant, with varying degrees of concentration or just simply the edge of a lake. When representing this as a region, there must be some means to indicate that some points belong to a lesser extent to the region than other points. Some authors have used additional boundaries to indicate the points that fully belong and the points that do not belong to the region (broad boundary model [1], egg-yolk model [2]), but such models don't allow for further specification of points in between both boundaries. The models have only been used for topological purposes as well. Other models have been developed, to model these points more accurately (e.g. [3] [4]).

The fuzzy region model requires a different view on regions: rather than defining a region by means of a boundary, the region is considered as a set of point. This set can be augmented to a fuzzy set, thus allowing each point to have a membership grade. An overview of this model was presented in [5].

2.1 Simple Fuzzy Regions

Definition 1 (Fuzzy region)

$$\tilde{R} = \{(p, \mu_{\tilde{R}}(p)) | p \in \mathbb{R}^2\} \tag{1}$$

A fuzzy region essentially is a fuzzy set defined over a two dimensional domain; the concept is illustrated on figure 1. Consequently, the traditional fuzzy operations for intersection and union (t-norms and t-conorms) are immediately applicable.

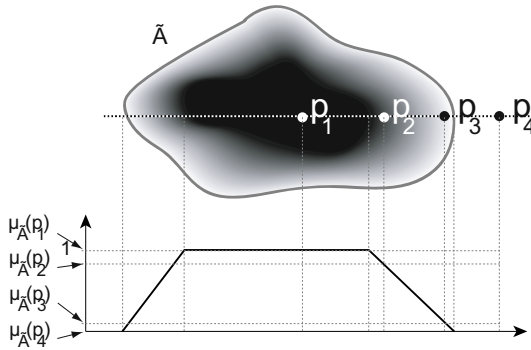


Fig. 1. The concept of a fuzzy region \tilde{A} ; a fuzzy set over a two dimensional domain. All points belong to some extent to the region; indicated by means of the membership grade. The lower half of the figure shows a cross section. The shades of grey relate to the membership grades: darker shades match higher membership grades (the region has a dark outline to indicate its maximal outline).

2.2 Fuzzy Regions Using Powerset

A first criticism to the fuzzy regions is that it is impossible to group point together. In some situations (e.g. the aforementioned example of the lake), it is possible that a user has additional knowledge that some points belong together and should form one basic element of the region. The fuzzy region model was therefore extended, by allowing the basic elements of the fuzzy region to be sets of points. This is illustrated on figure 2. The definition makes use of the powerset¹.

Definition 2 (Fuzzy region with powerset extension)

$$\tilde{R} = \{(P, \mu_{\tilde{R}}(P)) | P \in \wp(\mathbb{R}^2) \wedge \forall P_1, P_2 \in \tilde{R} : P_1 \cap P_2 = \emptyset\} \tag{2}$$

¹ The powerset of a set is the set of all subsets contained in that one set, including the empty set and the set itself.

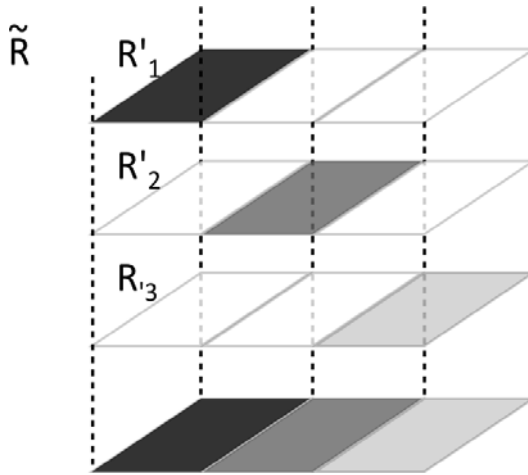


Fig. 2. A fuzzy region defined over the powerset of the two dimensional domain. The region \tilde{R} is comprised of three elements, the regions R'_1 , R'_2 and R'_3). These are crisp regions that are given a membership grade with a veristic interpretation; they are elements or subregions.

This extension allows for individual points to be modelled. The fuzzy region allows for the representation of regions, when the fuzzy set is given a veristic interpretation: all points belong to some extent to the region. Not that the intersection of two basic elements is required to be empty; this limitation was made to facilitate the operations and has no real impact on the usefulness: the model was mainly considered as a stepping stone towards the level-2 fuzzy regions.

3 Level-2 Fuzzy Regions

Consider the example of the pollutant or the lake from section 2. Both of them could be represented using a fuzzy region (with a veristic interpretation), but interpretation may be slightly off. Do we consider a point that is definitely flooded (in the case of the lake) for certain water levels to partly belong? To further illustrate: what if we want to make statements? The water level of the lake is known to vary, can we represent the lake still as a fuzzy region? If we don't know the water level at some point in time, it would require a possibilistic interpretation to describe its boundary. Similarly, if we want to predict where the pollutant could be in the next few days, there may be different possibilities on how it spreads (depending on weather conditions for instance). So this basically yields a number of possible outlines and again requires a possibilistic interpretation. Furthermore, the current model is dependent on the metadata, stating if the fuzzy region has a possibilistic or a veristic interpretation. This knowledge is important in order to apply the correct operators, and it may be confusing and difficult to define operators between fuzzy regions with different interpretations.

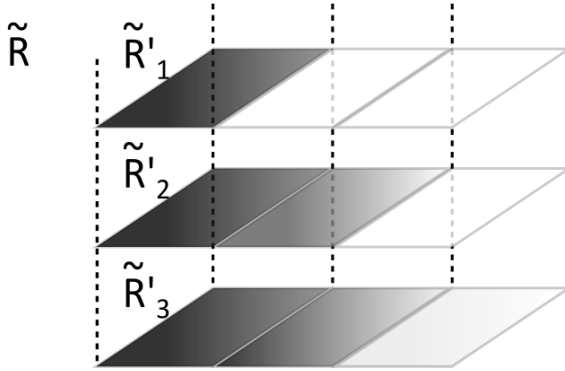


Fig. 3. The concept of a fuzzy region defined over the fuzzy powerset of the two dimensional domain. The region \tilde{R} represents three possible candidates: the regions \tilde{R}'_1 , \tilde{R}'_2 and \tilde{R}'_3). These are fuzzy regions that are given a membership grade with a possibilistic interpretation; they are candidates or possibilities.

These arguments led to the development of the level-2 fuzzy regions. The level-2 fuzzy region uses a fuzzy region (as in definition 11 or 12) with a veristic interpretation as basic element. Several such fuzzy regions are combined in a new fuzzy set, in which they are given possibility degrees (this set thus has a possibilistic interpretation). Each fuzzy region represents a possibility for the (fuzzy) feature to be modelled; this is illustrated on figure 3. The different fuzzy regions \tilde{R}'_1 , \tilde{R}'_2 and \tilde{R}'_3 could be representations for the lake with different water levels; or they could be predictions for the spread of the pollutant in different circumstances (e.g. different surface winds).

3.1 Definition

To achieve the concept formally, we will make use of the fuzzy powerset. The fuzzy powerset of a set A , denoted $\tilde{\varphi}(A)$, is the set of all fuzzy subsets that belong to A . Using this concept, the level-2 fuzzy region can be defined over the $\tilde{\varphi}(\mathbb{R}^2)$, effectively making the level-2 fuzzy region a fuzzy set of fuzzy sets (regions). This concept is known in literature as a level-2 fuzzy set ([6]) and is not to be confused with a type-2 fuzzy set ([7]), where the membership degrees are fuzzy sets. Simply stated, whereas the level-2 fuzzy set is a fuzzy set over a fuzzy domain with crisp membership grades, the type-2 fuzzy set is a fuzzy set over a crisp domain with fuzzy sets as membership grades. For extending the fuzzy regions; the level-2 concept was chosen as it allows us to model the concept of candidate fuzzy regions and keep the existing operations at that lower level. Using a type-2 extension would make it difficult to maintain the spatial relation between points in a concept similar to the candidate regions.

The level-2 fuzzy region can then be defined as:

$$\tilde{R} = \{(\tilde{R}', \mu_{\tilde{R}}(\tilde{R}')) | \tilde{R}' \in \tilde{\varphi}(\mathbb{R}^2)\} \quad (3)$$

with the membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \tilde{\varphi}(\mathbb{R}^2) &\mapsto [0, 1] \\ \tilde{R}' &\rightarrow \mu_{\tilde{R}}(\tilde{R}') \end{aligned}$$

Note that unlike in definition 2, the intersection of the elements is not required to be empty. This is because it concerns quite a different interpretation: in definition 2 the elements were considered to be subregions, groups of points that belong together; in the above definition, the elements are candidate regions. It is very likely for them to overlap in some parts, especially if the outline is not certain, but some central region is.

Operations. In [8], obtaining information regarding points that belong to a level-2 fuzzy region has been considered. It was shown that the membership of a single point in the level-2 fuzzy region can be expressed by means of a type-2 fuzzy set: each candidate region boasts a membership grade for the point; but with each candidate region comes a possibility degree. The union and intersection of level-2 fuzzy regions ([9]) has been presented; whereas other operations (e.g. distance and surface area) are under development.

4 Combination of Level-2 Fuzzy Regions

Due to the nature of level-2 fuzzy regions, it is necessary to consider possible means of constructing them. This is illustrated on figure 4. The first idea would be to simply use the set operations (union, intersection). However, while the union of two level-2 fuzzy regions yields a new level-2 fuzzy region, it does not really allow for an easy creation of a level-2 fuzzy set given a number of possible fuzzy regions. For this purpose, we need to introduce additional operations. Similarly, the intersection of two level-2 fuzzy regions yields a new level-2 fuzzy region, but it does not provide for an easy way of reducing the number of possibilities.

We will first list the set operations, show their shortcomings for constructing level-2 fuzzy regions and then introduce the additional operations.

4.1 Set Operations

In [9], the traditional set operations applied on level-2 fuzzy regions were presented; they require a double application of Zadehs extension principle ([10]), and the definitions will be repeated below.

Union. The union of two level-2 fuzzy regions is defined by means of the extension principle.

Definition 3 (Union of level-2 fuzzy regions)

$$\tilde{R}_1 \cup \tilde{R}_2 = \bigcup_{R'_1: \mu_{\tilde{R}_1}(R'_1) > 0 \wedge R'_2: \mu_{\tilde{R}_2}(R'_2) > 0} \{(\tilde{R}'_1 \cup \tilde{R}'_2, \mu_{\tilde{R}_1 \cup \tilde{R}_2}(\tilde{R}'_1 \cup \tilde{R}'_2))\} \quad (4)$$

The membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \wp(\mathbb{R}^2) &\mapsto [0, 1] \\ \mu_{\tilde{R}_1 \cup \tilde{R}_2}(\tilde{R}'_1 \cup \tilde{R}'_2) &\rightarrow S(\mu_{\tilde{R}_1}(\tilde{R}'_1), \mu_{\tilde{R}_2}(\tilde{R}'_2)) \end{aligned}$$

Conceptually, we see that the union takes all possible combinations of the candidate regions.

Intersection. The intersection of two level-2 fuzzy regions is defined by means of the extension principle.

Definition 4 (Intersection of level-2 fuzzy regions)

$$\tilde{R}_1 \cap \tilde{R}_2 = \bigcup_{R'_1 : \mu_{\tilde{R}_1}(R'_1) > 0 \wedge R'_2 : \mu_{\tilde{R}_2}(R'_2) > 0} \{(\tilde{R}'_1 \cap \tilde{R}'_2, \mu_{\tilde{R}_1 \cap \tilde{R}_2}(\tilde{R}'_1 \cap \tilde{R}'_2))\} \quad (5)$$

The membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \wp(\mathbb{R}^2) &\mapsto [0, 1] \\ \mu_{\tilde{R}_1 \cap \tilde{R}_2}(\tilde{R}'_1 \cap \tilde{R}'_2) &\rightarrow T(\mu_{\tilde{R}_1}(\tilde{R}'_1), \mu_{\tilde{R}_2}(\tilde{R}'_2)) \end{aligned}$$

Example. As an example, we will consider two regions as shown on figure 4a and figure 4b. Region \tilde{A} is a level-2 fuzzy region with two fuzzy candidate regions; Region \tilde{B} is a level-2 fuzzy region with a single candidate region. The result of the intersection operator of regions \tilde{A} and \tilde{B} is shown on figure 4c, the union is illustrated on figure 4d. Note that $\tilde{R}'_4 = \tilde{R}'_1 \cap \tilde{R}'_3$ and $\tilde{R}'_5 = \tilde{R}'_2 \cap \tilde{R}'_3$; whereas $\tilde{R}'_6 = \tilde{R}'_1 \cup \tilde{R}'_3$ and $\tilde{R}'_7 = \tilde{R}'_2 \cup \tilde{R}'_3$. These are the results as obtained from the set-operation defined above.

To reason with level-2 fuzzy regions, it is also necessary to be able to add or remove possibilities to/from a level-2 fuzzy region. Adding the region \tilde{B} to \tilde{A} then yields the result shown on figure 4e. The example shows that - while union and intersection are meaningful operators to combine different data - there also is a need for additional operations to process the level-2 fuzzy regions.

4.2 Construction of and Reasoning with Level-2 Regions

As was explained in the previous section, the union and intersection as presented in [9] are not really suited to add or remove candidate regions, nor to merely alter the possibilities of candidate regions. To increase or decrease the number of possibilities, it is not possible to use the standard union and intersection as defined above (and as suggested in [6]): the double application of the extension principle complicates things in this situation. For the purpose of easily modifying level-2 regions, additional operators will be introduced in this contribution. These operations are even considered by some as valid intersection and union operations, as the second application of the extension principle (as suggested in [6]) can yield counter-intuitive results.

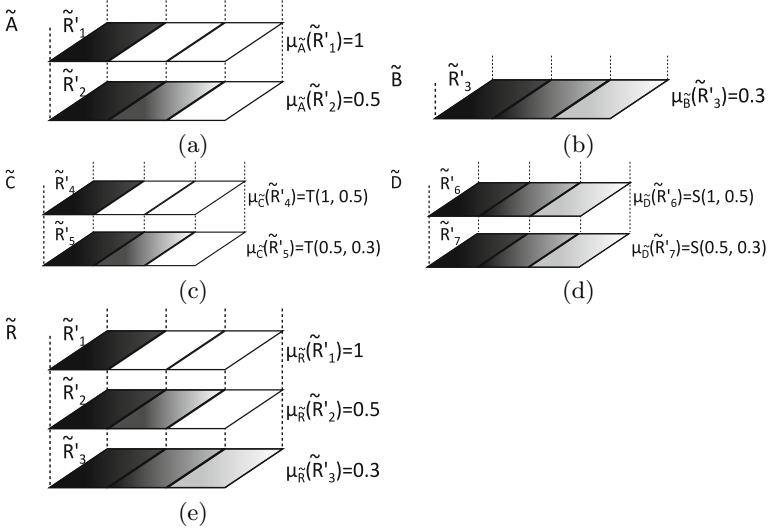


Fig. 4. An example showing the issues with union and intersection. Two 2 level-2 fuzzy regions are shown in (a) and (b), their intersection is shown in (c), their union in (d) and the combinations of both regions into a new level-2 fuzzy region in (e).

Addition/Increase of Possibilities. To add candidate regions, we consider 2 level-2 fuzzy regions and merge all the candidate regions of both level-2 fuzzy regions. This means the union of only the top level needs to be considered. All possible candidate regions from both arguments are grouped in a single fuzzy set.

Definition 5 (Adding/increasing possibilities in level-2 fuzzy regions)

$$\tilde{R}_1 \oplus \tilde{R}_2 = \bigcup_{R': \mu_{\tilde{R}_1}(\tilde{R}') > 0 \vee \mu_{\tilde{R}_2}(\tilde{R}') > 0} \{(\tilde{R}', \mu_{\tilde{R}_1 \oplus \tilde{R}_2}(\tilde{R}'))\} \quad (6)$$

The membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \wp(\mathbb{R}^2) &\mapsto [0, 1] \\ \mu_{\tilde{R}_1 \oplus \tilde{R}_2}(\tilde{R}') &\rightarrow S(\mu_{\tilde{R}_1}(\tilde{R}'), \mu_{\tilde{R}_2}(\tilde{R}')) \end{aligned}$$

If the region to be added occurs twice in the result, an appropriate T-conorm is used to determine the membership grade of the region - commonly, this will be the maximum. This allows for multiple information about different candidates for the same region to be combined. The result of this is illustrated on figure 4.

Reduction of Possibilities. A similar approach can be applied to reduce the number of possibilities. Rather than consider all candidate regions that occur either fuzzy region, we only consider those that only belong to both candidate regions, thus limiting the number of possibilities and possibly decreasing the membership grades.

Definition 6 (Reduction of the possibilities in a level-2 fuzzy region)

$$\tilde{R}_1 \ominus \tilde{R}_2 = \bigcup_{R': \mu_{\tilde{R}_1}(\tilde{R}') > 0 \wedge R': \mu_{\tilde{R}_2}(\tilde{R}') > 0} \{(\tilde{R}', \mu_{\tilde{R}_1 \ominus \tilde{R}_2}(\tilde{R}'))\} \quad (7)$$

The membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \wp(\mathbb{R}^2) &\mapsto [0, 1] \\ \mu_{\tilde{R}_1 \ominus \tilde{R}_2}(\tilde{R}') &\rightarrow T(\mu_{\tilde{R}_1}(\tilde{R}'), \mu_{\tilde{R}_2}(\tilde{R}')) \end{aligned}$$

If a candidate region \tilde{R}' exists in both level-2 fuzzy regions, a T-norm is used to determine its membership grade; this commonly will be the minimum. Note that to remove a candidate region, it should not belong to one of the arguments.

To remove a candidate region, given the candidate region to remove as an argument, one would need an operation such as the one defined below. Not only does it allow for the complete removal, but it also allows for decreasing the membership grades.

Definition 7 (Removal of possibilities from a level-2 fuzzy region)

$$\tilde{R}_1 \tilde{R}_2 = \bigcup_{R': \mu_{\tilde{R}_1}(\tilde{R}') > 0} \{(\tilde{R}', \mu_{\tilde{R}_1 \setminus \tilde{R}_2}(\tilde{R}'))\} \quad (8)$$

The membership function is defined as:

$$\begin{aligned} \mu_{\tilde{R}} : \wp(\mathbb{R}^2) &\mapsto [0, 1] \\ \mu_{\tilde{R}_1 \tilde{R}_2}(\tilde{R}') &\rightarrow \min(0, \mu_{\tilde{R}_1}(\tilde{R}') - \mu_{\tilde{R}_2}(\tilde{R}')) \end{aligned}$$

Complexity. The model (and thus also the operations) at this point are purely theoretical to built the foundations for the representation models. To consider the complexity of the operators, first more practical models should be derived. This is currently in progress, and we are working on similar approaches as for the simple fuzzy regions: these were approximated using triangular networks or bitmaps ([5], [11], [12]). The same representation can be used for the candidate fuzzy regions, in which case the complexity depends on both the representation methods uses and the number of candidate regions involved. As only a single application of the extension principle is performed, the operations will be faster than the set operations defined that apply the extension principle twice ([9]).

5 Application Examples

5.1 Feature Representation

Consider a lake for which the water level can vary, based on the amount of rainfall of the last couple of weeks, melting snow at the end of winter, etc. Traditionally,

a lake will be modelled by a single region; at best there can be flood regions associated with it. Using the level-2 fuzzy region model, it is possible to represent this lake as a single object in the database. The model can store as many different outlines for the lake as desired; these can be added to the model using the operator defined in Section 4.2. Any queries related to the lake will take into account all possible (represented) water levels in the results.

5.2 Representation of Query Results

As mentioned in the introduction, fuzzy queries on crisp data can yield fuzzy results. Querying a spatial database for locations that are *close* to an existing location is the simplest example. Similar results can be obtained for other queries. In an interface where the user indicates locations on a map, imprecision can be introduced to take into account the scale at which the user is working: if the user indicates a point on a map with a small scale, the point is likely to be less accurate than if a large scale would have been used.

5.3 Usage in Querying

Suppose an area that matches specific criteria needs to be pinpointed: this can for instance be necessary to trace the last whereabouts of a missing person, or to determine the best place to build some facilities. Suppose some of the criteria are: close to a river, close to a highway, far from a forest, good connections with public transport, ... For every criteria, a level-2 fuzzy region can be constructed: close to a river will yield a number of possible locations; these can be combined with the results of the other queries using the intersection. Querying for a location that satisfies these criteria is then achieved by combining the level-2 fuzzy regions.

6 Conclusion

In this contribution, modification of level-2 fuzzy regions was considered. The level-2 fuzzy regions are a novel model to represent uncertainty and imprecision of spatial data represented by regions or points. The creation and modification of level-2 fuzzy regions is made more difficult by the fact that set operations mathematically need to be defined using a double application of the extension principle; this yields undesired results when intending to merely add or remove possibilities. For this purpose, new operations have been introduced, which allow a more intuitive approach to add possibilities to or remove possibilities from a level-2 fuzzy set.

References

1. Clementini, E., Di Felice, P.: An algebraic model for spatial objects with undetermined boundaries. In: GISDATA Specialist Meeting - Revised Version (1994)
2. Cohn, A., Gotts, N.M.: Spatial regions with undetermined boundaries. In: Proceedings of the Second ACM Workshop on Advances in GIS, pp. 52–59 (1994)

3. Schneider, M.: Modelling Spatial Objects with Undetermined Boundaries Using the Realm/ROSE Approach. In: Geographic Objects with Indeterminate Boundaries. GISDATA Series, vol. 2, pp. 141–152. Taylor & Francis, Abington (1996)
4. Kanjilal, V., Liu, H., Schneider, M.: Plateau regions: An implementation concept for fuzzy regions in spatial databases and GIS. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS(LNAI), vol. 6178, pp. 624–633. Springer, Heidelberg (2010)
5. Verstraete, J., De Tré, G., De Caluwe, R., Hallez, A.: Field based methods for the modelling of fuzzy spatial data. In: Fred, P., Vince, R., Maria, C. (eds.) Fuzzy Modeling with Spatial Information for Geographic Problems, pp. 41–69. Springer, Heidelberg (2005)
6. Gottwald, S.: Set theory for fuzzy sets of higher level. *Fuzzy Sets and Systems* 2(2), 125–151 (1979)
7. Mendel, J.M.: Uncertain rule-based fuzzy logic systems, Introduction and new directions. Prentice Hall, Englewood Cliffs (2001)
8. Jörg, V.: Using level-2 fuzzy sets to combine uncertainty and imprecision in fuzzy regions. In: Mugellini, E., Szczepaniak, P.S., Pettenati, M.C., Sokhn, M. (eds.) AWIC 2011. AISC, vol. 86, pp. 163–172. Springer, Heidelberg (2011)
9. Verstraete, J.: Union and intersection of level-2 fuzzy regions. In: World Conference on Soft Computing (2011)
10. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
11. Verstraete, J., De Tré, G., Hallez, A., De Caluwe, R.: Bitmap based structures for the modelling of fuzzy entities. *Control and Cybernetics* 35, 147–164 (2006)
12. Verstraete, J., De Tré, G., Hallez, A., De Caluwe, R.: Using tin-based structures for the modelling of fuzzy gis objects in a database. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 15, 1–20 (2007)

Bipolar Fuzzy Querying of Temporal Databases

Christophe Billiet², Jose Enrique Pons¹, Tom Matthé²,
Guy De Tré², and Olga Pons Capote¹

¹ Department of Computer Science and Artificial Intelligence
University of Granada
C/Periodista Daniel Saucedo Aranda s/n E-18071, Granada-Spain
jpons,opc@decsai.ugr.es

² Department of Telecommunications and Information Processing,
Ghent University,
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
{Christophe.Billiet,Tom.Matthe,Guy.DeTre}@UGent.be

Abstract. Temporal databases handle temporal aspects of the objects they describe with an eye to maintaining consistency regarding these temporal aspects. Several techniques have allowed these temporal aspects, along with the regular aspects of the objects, to be defined and queried in an imprecise way. In this paper, a new technique is proposed, which allows using both positive and negative -possibly imprecise- information in querying relational temporal databases. The technique is discussed and the issues which arise are dealt with in a consistent way.

1 Introduction

Temporal databases are databases that handle certain temporal aspects of the data they contain [9]. In temporal databases, time-related attributes are generally not treated like the other attributes, though both describe properties of the same objects. This is because the time-related attributes are considered to have an impact on the consistency of the object set modelled by the database. The database will thus enforce a consistent behaviour towards time.

In the presented paper, the focus is on relational temporal databases. The necessity of the mentioned consistency is explained in the following example. Consider a car rental service with a database at its disposal, which contains the properties of every car the service owns, including usual properties like car color, but also time-related properties like the starting and ending day of a car rental. If time is not handled specifically by the database, the rental office employees could insert several different records for the same car, each containing a different starting day and the same ending day. This would suggest that that same car was rented multiple times to possibly different clients at the same time, which is physically impossible, resulting in loss of consistency. A temporal database system would implicitly prevent this situation from occurring.

Now consider clients expressing their preferences for cars in both positive and negative statements and describing these preferences in an imperfect way.

E.g. a client could request a car which is ‘about’ 2 years old, preferably black, but certainly not yellow, and will be available during approximately all of next month.

In the previous example, as in almost every real-life application, human preferences are at the basis of every query. However, humans express their preferences in both positive and negative statements. Positive preferences express what is desired, acceptable or satisfactory, while negative statements express what is undesired, unacceptable or unsatisfactory. Depending on the situation, both can be used. This introduces the need for bipolar querying, a querying technique which allows introducing both positive and negative user preferences in a database query.

The combination of bipolar querying and the use of imprecise query preferences is well discussed in existing literature [5], [8], but not in the context of temporal databases. This paper will present an approach to query temporal databases using both positive and negative imprecise -and possibly temporal- preferences. In existing bipolar querying techniques, query results are ranked before presenting to the user, to provide the user with the best fitting results. In some existing techniques for querying temporal databases, records are either accepted or rejected as query results based on whether the queried time indication is totally covered by the record’s time specification or not. The approach presented here will integrate a bipolar ranking technique with a technique to determine the degree of satisfaction of records to the query’s temporal preference, to determine a record ranking best fitting for the user.

The rest of the paper is structured as follows. In section 2, some general concepts and issues of both temporal databases and bipolar querying are explained, both in the context of crisp and imprecise databases. In section 3, the presented approach is described and discussed and finally illustrated with an example. Section 4 contains the conclusions.

2 Preliminaries

2.1 Bipolar Satisfaction Degrees

As stated above, humans sometimes express their preferences using both positive and negative statements. Moreover, the positive and negative statements do not necessarily have to be each others inverse. E.g., when a user states that he doesn’t want a blue car, it is not necessarily the case that he/she will be equally satisfied with any other color. This is what is called *heterogeneous* bipolarity [7], [8].

In fuzzy querying of regular databases, query satisfaction modeling is a matter of degree. Usually, criteria satisfaction is modeled by means of a satisfaction degree $s \in [0, 1]$, expressing the degree to which a particular database record is satisfactory according to a given criterion. It is implicitly assumed that the degree on which the database record is unsatisfactory according to the given criterion, is exactly the inverse of the satisfaction degree, i.e., $d = 1 - s$ with $d \in [0, 1]$ the dissatisfaction degree. This assumption leads to the concept of ‘symmetric bipolarity’. However, as explained above, this assumption does not

always hold, leading to the concept of ‘heterogeneous bipolarity’ where satisfaction and dissatisfaction are not necessarily each others inverses. This can happen in two ways. First, there can be some indifference about whether some attribute values are satisfactory or not. In the extreme case the attribute value will be neither satisfactory nor unsatisfactory. E.g., in searching for a new car, a user might be totally indifferent about the satisfaction of a car regarding the color attribute when the car has the color ‘green’ (i.e., a green car will be neither satisfactory, nor unsatisfactory). In that case the overall satisfaction depends on the other attributes. Secondly, there might be some conflict in the user’s query specifications. In the extreme case, the attribute value will be satisfactory as well as unsatisfactory according to the user’s query specifications. Of course, these conflicting specifications are not desirable in query handling, but they can never be ruled out when trying to model human reasoning. To explicitly handle and model this ‘heterogeneous bipolarity’, the concept of a bipolar satisfaction degree has been introduced [15].

Definition. A *bipolar satisfaction degree* (BSD) is a pair

$$(s, d), \quad s, d \in [0, 1]$$

with s the *satisfaction degree* and d the *dissatisfaction degree*. Both s and d take their values in the unit interval $[0, 1]$ and are independent of each other. They reflect to what extent the bipolar representation corresponds to the concepts ‘satisfied’, respectively ‘dissatisfied’. The extreme values are 0 (‘not at all’) and 1 (‘fully’).

This definition is very closely related to Atanassov intuitionistic fuzzy sets (AFS) [2], except that the consistency condition ($0 \leq s + d \leq 1$) is missing. Because s and d are considered to be completely independent of each other, it is allowed that $s + d > 1$. The reason for this is that BSDs try to model heterogeneous bipolarity in human reasoning, and that human reasoning by definition isn’t always consistent.

Dealing with BSD’s in Bipolar ‘Fuzzy’ Querying. In this paper, we only study one specific kind of ‘fuzzy’ queries, namely those that are specified by two composed query conditions Q^{pos} and Q^{neg} . The case where one (bipolar) query condition can specify both positive and negative preferences at the same time, falls outside the scope of this paper. The condition Q^{pos} reflects all the positive (possibly ‘fuzzy’) preferences of the user, whereas the condition Q^{neg} expresses all the negative (possibly ‘fuzzy’) preferences of the user. As such, the composed conditions Q^{pos} and Q^{neg} can respectively be considered as poles of positive and negative conditions, where the conditions themselves can possibly be fuzzy.

A bipolar ‘fuzzy’ query \tilde{Q} with positive and negative criteria is then formally specified by

$$\tilde{Q} = (Q^{pos}, Q^{neg}).$$

where Q^{pos} represents the logical expression of positive query conditions and Q^{neg} represents the logical expression of negative query conditions.

Ranking. Processing a bipolar ‘fuzzy’ query as described above will, in the presented framework, result in an associated BSD for each database tuple. In order to rank these tuples in accordance with their global query satisfaction, a ranking function for BSDs is required. Different ranking functions can be used [15]. One of them, which gives equal importance to the satisfaction degree and the dissatisfaction degree, and which will be used in this paper, is

$$\text{Rank}_{BSD} = s - d \in [-1, 1]$$

Three special cases can be distinguished:

- $s - d = 1$: in this case it must be that $s = 1$ and $d = 0$, so this is the case of *full satisfaction* (without any indifference or conflict).
- $s - d = -1$: in this case it must be that $s = 0$ and $d = 1$, so this is the case of *full dissatisfaction* (without any indifference or conflict).
- $s - d = 0$: in this case the ranking is *neutral*. The criterion is as satisfied as it is dissatisfied.

2.2 Time in Databases

The concept of time has been studied in databases for a long time. A true standard for adding temporal aspects to relational databases does not exist, but there is a consensus in the literature [9] on what is called a *temporal database*: a temporal database is a database dealing with some aspects of time in its schema. In a temporal DBMS, a **chronon** is the shortest duration of time supported by the system. In temporal databases, some temporal attributes can be managed without treating the attribute differently from non-temporal attributes. The time described by such an attribute is called **user defined time** (*UDT*). In addition to UDT, the following types of time can be discerned in a temporal database, all of which are handled exceptionally by the DBMS:

- **Transaction time.** (*TT*) [20], [13] denotes the time when the fact (object) is stored in the database. It is usually append-only: as the past can not be changed, TT can not be changed neither. Furthermore, at the moment of insertion, a TT can be neither in the past nor in the future.
- **Valid time.** (*VT*) [14], [21] denotes the time when the fact (object) is true in the modelled reality. A fuzzy extension has been proposed by [11].
- **Decision time.** (*DT*, proposed in [17]) denotes the time when an event was decided to happen.

E.g., consider a database containing employee contract descriptions. The time when the employee’s contract is valid, represented by an interval, is VT. The time when the employee’s contract is stored in the database is the TT. The time when the decision for hiring this employee was made is the DT.

When working with these time concepts, the Data Manipulation Language (*DML*, which is part of the standard database querying language SQL) is extended to deal with possible temporal inconsistencies within the data and to

handle more complex (temporal) queries. Depending on the time managed, a database is classified as either a **Valid Time Database (VTDB)**, a **Transaction Time Database (TTDB)**, a **bi-temporal database** (both valid and transaction time are managed) or a **tri-temporal database** (valid time, transaction time and decision time are managed).

Imperfection and Time. Representing imprecision and its semantics when dealing with time has been studied for a long time. Several proposals for representing and computing imprecise time indications can be found in [3] and [4]. Also, the changes between several granularities can be seen as a source of imprecision [6].

In the proposal section we will consider two kinds of imprecision:

- **Uncertainty in the database** denotes the uncertainty that arises when the knowledge about the temporal data in the database is uncertain. E.g., a database record shows that *‘The car is in the garage around April.’*
- **Imprecision in the query specification** denotes the imprecision in the specification of temporal criteria by the user, when querying. E.g., *‘The user wants to obtain a car which is red and which is in the garage around April.’*

Representation. Several proposals for managing uncertain time in a database exist. Some proposals work with rough sets [19], other proposals rely on possibility distributions for representing uncertainty in time [11], [10]. In order to compare temporal possibility distributions, extensions of the classical Allen’s operators [1] are defined in [18] and [16]. In the proposal section, we will follow the representation by means of possibility distributions, in order to work with both satisfaction and dissatisfaction degrees. Also, in order to work properly with fuzzy operators, the underlying domain should be numeric. In this paper, the representation for the dates will follow the Julian Day Number (JDN) representation [12].

If the starting point and/or the end point of the interval representing the time are not known precisely, it is easy to fuzzify them, using, e.g., two triangular membership functions. A **Fuzzy Validity Period (FVP)** is defined as a fuzzy time interval specifying when an object is valid. A fuzzy time interval is then the fuzzification of a crisp time interval. Several options to transform fuzzy time intervals corresponding to a fuzzy starting point and a fuzzy end point into one consistent fuzzy time interval exist [11], e.g (Fig. 1):

- The **convex hull** approach is the most intuitive approach. The resulting FVP is the convex hull of the union of both fuzzy sets.
- The **information preserving** approach is less intuitive but more realistic. The total amount of information is maintained at the edges of the membership function of the fuzzy time interval [11].

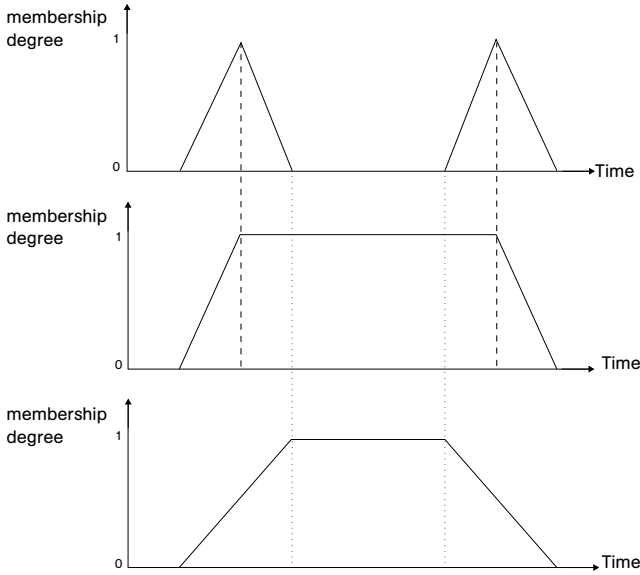


Fig. 1. Transformation to obtain the FVP. The top graph shows the two triangular membership functions. The middle graph shows the convex hull validity period, the bottom one shows the result of the second transformation, which maintains the information.

3 Proposal

In subsection [3.1](#), the context of this paper’s approach is described, and the general ideas behind this approach are discussed. In subsection [3.2](#), the approach itself is presented and in [3.3](#), an example database and query are presented, which are used to illustrate the proposed approach.

3.1 Context

In this paper, relational VTDB’s are considered, as the intention is to deal with cases in which the validity of an object is important knowledge, but the value of the transaction time is not.

Furthermore, a VTDB is queried by means of an extension of the bipolar ‘fuzzy’ querying approach introduced in [5](#) and briefly described in section [2.1](#). In accordance with [5](#), every non valid time attribute can contain imprecise data. To support valid time querying, the queries are extended to allow users to indicate a time period in which a result record is preferred to be valid. This introduces a consideration.

When departing from a query as specified in [5](#), a preference towards valid time can be inserted at two levels:

- **Local level.** Here, Q^{pos} , as well as Q^{neg} , are seen as logical aggregations of elementary query conditions and each of these elementary conditions is

extended with a time demand. This means that every elementary condition is given an elementary temporal constraint which is evaluated and aggregated with the evaluation result of the condition. The introduced demands are defined by the user when the query is constructed and each one specifies a time constraint related with the fulfillment of the corresponding non-temporal condition.

- **Global level.** Here, the entire query is extended with a single time demand. This means that the entire query is given one elementary temporal constraint. The introduced temporal demand is defined by the user when the query is constructed and specifies a condition or constraint on the valid time period of a record.

In the proposed approach, only the last option is chosen. This is further described in section 3.2.

It should be clear that it is still possible to query user defined time, without modifying the model: the query specification remains the same and the temporal constraint is aggregated in the Q^{pos} - or Q^{neg} -expression, depending on the positive or negative nature of the time constraint.

3.2 Approach

The Query Structure. The approach followed here introduces a global time demand: the user can define one time demand for the entire query. A query now has the following structure:

$$(Q^{time}, (Q^{pos}, Q^{neg}))$$

As in [5], Q^{pos} and Q^{neg} represent the positive and negative preference criteria, respectively, and Q^{time} represents the global temporal condition.

The user can specify the time demand using a time interval or a starting and end time. Both cases can contain imprecision concerning the starting and end times.

Time Imperfection. It should be noted that in the presented approach, time will be represented as an interval with a starting point and an end point and a time point will be seen as a time interval in which the starting point and the end point coincide. Only the starting point and the end point are allowed to contain some imperfection. Even though the time domain can be discrete, a time interval will be seen as a continuous interval to easily allow the transition to FVP's.

The time specification in the query will instantly be translated into one trapezoidal FVP. The presented approach stays indifferent to the way in which this query FVP is calculated based upon a possible given crisp time period, but the interpretation of the query FVP should always be that the user prefers an object that is valid during at least the entire query FVP (the interpretation is conjunctive) and the certainty with which the user needs the object to be valid is expressed by the membership degree of the FVP.

The valid time indications in the database can take the form of one point, an interval or a FVP. All three options will be treated as FVP's. The interpretation is that the object described by the record in the database is valid during its entire FVP, but there can be uncertainty about the exact starting point and the exact end point of the valid time, which is modelled by the membership function of the FVP of the record (the interpretation is, again, conjunctive).

The Evaluation of the Query. In the presented approach, every record r contains a trapezoidal FVP \tilde{V}_r to express the record's valid time. The approach then is the following:

For every record r in the database, the bipolar query criteria Q^{pos} and Q^{neg} are evaluated, resulting in a BSD of the form (s_r, d_r) , where s_r denotes the degree of satisfaction and d_r denotes the degree of dissatisfaction of the record. Separately, the temporal condition Q^{time} is evaluated in an attempt to define the degree to which r satisfies the query's temporal demand. Q^{time} contains a trapezoidal FVP \tilde{V}_q . For every record in the database, a validity satisfaction degree (VSD) $deg_{vs}(\tilde{V}_q, \tilde{V}_r)$ is computed using the fuzzy set gradual inclusion operator, with the minimum-function as t-norm.

$$deg_{vs}(\tilde{V}_q, \tilde{V}_r) = deg(\tilde{V}_q \subseteq \tilde{V}_r) = \frac{card(\tilde{V}_q \cap \tilde{V}_r)}{card(\tilde{V}_q)}$$

As time intervals are seen as continuous intervals, this formula is interpreted as follows.

$$deg_{vs}(\tilde{V}_q, \tilde{V}_r) = \frac{\int_{x \in U} \min(\mu_{\tilde{V}_q}(x), \mu_{\tilde{V}_r}(x)) dx}{\int_{x \in U} \mu_{\tilde{V}_q}(x) dx}$$

In this equation, U is the time domain and $\mu_{\tilde{V}_q}$ and $\mu_{\tilde{V}_r}$ are the membership functions of \tilde{V}_q and \tilde{V}_r respectively.

The interpretation is the following: as the interest here lies with the degree to which it is possible for the record to satisfy the query's temporal requirement, this method will compare the overlapping surface of the record's FVP and the query's FVP to the entire surface of the query's FVP. This results in a degree (VSD) measuring how much of the query's temporal requirement can be resolved by the record and thus how well the record fits the query's temporal demand. This VSD is now the requested degree to which r satisfies the query's temporal demand.

In the approach presented here, every trapezoidal FVP \tilde{V} is represented using four values $([\alpha_{\tilde{V}}, \beta_{\tilde{V}}, \gamma_{\tilde{V}}, \delta_{\tilde{V}}])$, where $\alpha_{\tilde{V}} \leq \beta_{\tilde{V}} \leq \gamma_{\tilde{V}} \leq \delta_{\tilde{V}}$. $\alpha_{\tilde{V}}$ and $\delta_{\tilde{V}}$ denote the beginning, resp. end of the support of \tilde{V} and $\beta_{\tilde{V}}$ and $\gamma_{\tilde{V}}$ denote the beginning, resp. end of it's core. The calculation of the VSD for a record can then be reduced to a case study on the relative positions of the α , β , γ and δ values of the query FVP and the record FVP.

Presenting the Results to the User. This approach was designed to present the user both the resulting records and the degrees to which these records satisfy

his or her query. To discover the records which are the most useful to the user, the results have to be ranked. As the Q^{time} is evaluated independently from the (Q^{pos}, Q^{neg}) for every record, the ranking for the VSD will be determined independently from the ranking for the BSD for every record. The BSD's are ranked as presented in section 2.1. The VSD's are values in $[0, 1]$ and thus have a natural ranking.

To achieve the final ranking, the rank of the total BSD ($Rank_{BSD}$) and the VSD of a record are combined using a convex combination of both ranks:

$$Rank_{Total} = \omega * Rank_{BSD} + (1 - \omega) * VSD$$

This somewhat unusual approach allows the final ranking to show the effects of the BSD and the VSD in chosen proportions. Thus, by increasing the parameter ω , the non-temporal demands of the user can be given more importance and by lowering the ω , the time demand can be emphasized.

3.3 Example

The Database. Consider a car rental service which uses the VTDB given in Table 1. Next to some general attributes (Color, Fuel consumption (noted F.C., in l/100 km), Age (in years)), the relation shows the FVP of each car in the $[\alpha, \beta, \gamma, \delta]$ - format explained above, where a null value in γ and δ means that the car is still valid right now. The ID field uniquely identifies a physical car, but the properties of a car can be modified: some engine optimizations could allow less fuel consumption, changes in the color could be made, etc. E.g. the car described in records 1 and 5 is the same (both records have the same ID value), but their colors differ. Therefore, a record in this VTDB will be uniquely defined by the combination of the ID field and the Instance ID (IID) field, which contains unique values for records with the same ID field value. A little remark: As the age of the car has nothing to do with when the car is available for rent, the 'Age'-attribute is considered UDT. Also, in this example, for the sake of simplicity, the chronons will be days.

The Query. Consider the following query:

The user does not want to rent a black car, and prefers a red car which is either younger than 6 years or has an average fuel consumption that is around 5 or 6 litres/100 km and the car should be available around February 2011.

Using the structure $(Q^{time}, (Q^{pos}, Q^{neg}))$ presented above, the query translates to:

$$(c^{time}, (c_{Color}^{pos} \wedge (c_{Age}^{pos} \vee c_{Fuel}^{pos}), c_{Color}^{neg}))$$

with

- $c_{Color}^{pos} = \{(Red, 1)\}$
- $c_{Color}^{neg} = \{(Black, 1)\}$
- $c_{Fuel}^{pos} = \{(5, 1), (6, 1), (7, 0.5)\}$
- $c_{Age}^{pos} = \{(0, 1), (1, 1), (2, 1), (3, 1), (4, 0.7), (5, 0.5), (6, 0.3)\}$

Table 1. The valid time car relation with fuzzy validity periods (FVP)

ID	IID	Color	F.C.	Age	FVP
001	1	Black	6	4	[01/12/2010,10/12/2010,10/01/2011,19/01/2011]
002	1	Red	5	6	[20/12/2010,25/12/2010,25/02/2011,28/02/2011]
003	1	Blue	4	2	[27/02/2011,01/03/2011, -, -]
004	1	Black	6	7	[28/12/2010,01/01/2011,05/03/2011,10/03/2011]
001	2	Red	6	4	[1/01/2011,11/01/2011, -, -]
002	2	Red	5	6	[16/02/2011,26/02/2011, -, -]
004	2	Black	6	7	[25/03/2011,05/04/2011, -, -]

Table 2. Result set of the query

ID	IID	$Rank_{BSD}$	VSD	$Rank_{Total}$				
				$\omega = 0$	$\omega = 0.25$	$\omega = 0.5$	$\omega = 0.75$	$\omega = 1$
001	1	-1	0	0	-0.25	-0.5	-0.75	-1
002	1	1	0.817	0.817	0.863	0.9085	0.954	1
003	1	0	0.142	0.142	0.107	0.071	0.036	0
004	1	-1	1	1	0.5	0	-0.5	-1
001	2	1	1	1	1	1	1	1
002	2	1	0.338	0.338	0.504	0.669	0.835	1
004	2	-1	0	0	-0.25	-0.5	-0.75	-1

The temporal condition is then represented by the next trapezoidal FVP.

$$c^{time} = [\alpha_t, \beta_t, \gamma_t, \delta_t] = [25/01/2011, 1/02/2011, 28/02/2011, 10/03/2011].$$

Results and Discussion. The example data set is shown in Table 1. The result set of the query, using the presented approach, is given in Table 2. Here, $Rank_{BSD}$ is the ranking of the BSD originating from the non-temporal bipolar constraints and $Rank_{Total}$ gives the total ranking (between -1 and 1) of the record with respect to the query, based on the chosen values for ω .

With $\omega = 0$, only the temporal constraint is taken into account. Because of the graduality of the VSD's, a richer ranking can be discerned than the simple acceptance or rejection which would occur when the query time interval would be requested to be fully contained in the record time intervals.

With growing values of ω , the non-temporal constraints grow in importance and the third record is ranked increasingly better than the fourth record, though the fourth record has a much higher VSD. This phenomenon can be desirable, as the fourth record has a much lower BSD rank.

With $\omega = 0.5$, only the record that scores high in both constraints gets a very high ranking (the fifth and the second records). Records with a low ranking for one of both constraints, are punished for this in the overall ranking, with the degree of punishment relative to the lowness of the ranking. Vice versa, low scores for either the temporal or the non-temporal constraints can be compensated for with high scores for the other constraints. This introduces a very natural

ordering, which ranks the records with respect to both the temporal and the non-temporal preferences of the user.

4 Conclusions

In the presented work we have introduced the imprecise querying of valid time as a new criterion in a bipolar query specification. This criterion is independent of the positive and negative non-temporal preferences. Thus, an independent way of evaluating and ranking the temporal constraint is suggested, which takes the user's preferences into account. It is reasoned that a poor satisfaction of an imprecise time constraint can be compensated with a good satisfaction of other constraints. In view of this, a general ranking system is proposed and discussed.

Further research work includes introducing bipolarity in the temporal query specification. E.g. the user may request one time period but reject another one, when specifying the valid time constraint in the query. More complex relations may be modelled using Allen's [\[1\]](#) operators.

Representing valid time in a bipolar way is of particular interest in historical databases. In these databases, time is not always precisely known. Sometimes the information about the validity period is expressed in an imprecise way. Applying these techniques to historical temporal databases will improve the research in this field and is also a topic for further research.

Acknowledgements. Part of the research is supported by the grant BES-2009-013805 within the research project TIN2008-02066: *Fuzzy Temporal Information treatment in relational DBMS*.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 832–843 (1983)
2. Atanassov, K.T.: Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87–96 (1986)
3. De Caluwe, R., Van der Cruyssen, B., De Tré, G., Devos, F., Maesfranckx, P.: Fuzzy time indications in natural languages interfaces, pp. 163–185. Kluwer Academic Publishers, Norwell (1997)
4. De Tré, G., De Caluwe, R., Van der Cruyssen, B.: Dealing with time in fuzzy and uncertain object-oriented database models. In: *EUFIT 1997*, 1157–1161 (September 1997)
5. De Tré, G., Zadrożny, S., Matthé, T., Kacprzyk, J., Bronselaer, A.: Dealing with Positive and Negative Query Criteria in Fuzzy Database Querying Bipolar Satisfaction Degrees. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) *FQAS 2009*. LNCS, vol. 5822, pp. 593–604. Springer, Heidelberg (2009)
6. Devos, F., Maesfranckx, P., De Tré, G.: Granularity in the interpretation of around in approximative lexical time indications. *Journal of Quantitative Linguistics* 5, 167–173 (1998)

7. Dubois, D., Prade, H.: Bipolar Representations in Reasoning, Knowledge Extraction and Decision Processes. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 15–26. Springer, Heidelberg (2006)
8. Dubois, D., Prade, H.: Handling bipolar queries in Fuzzy Information Processing. In: Dubois, D., Prade, H. (eds.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 97–114. Information Science Reference, New York, New York (2008)
9. Dyreson, C., Grandi, F., et al.: A consensus glossary of temporal database concepts. SIGMOD Rec. 23, 52–64 (1994)
10. Galindo, J., Medina, J.M.: Ftsql2: Fuzzy time in relational databases. In: EUSFLAT Conf. 2001, pp. 47–50 (2001)
11. Garrido, C., Marin, N., Pons, O.: Fuzzy intervals to represent fuzzy valid time in a temporal relational database. Int. J. Uncertainty Fuzziness Knowledge-Based Syst. 17 (2009)
12. Husfeld, D., Kronberg, C.: Astronomical time keeping (1996), <http://www.maa.mhn.de/Scholar/times.html>
13. Jensen, C.S., Mark, L., Roussopoulos, N.: Incremental implementation model for relational databases with transaction time. IEEE Trans. Knowl. Data Eng. 3, 461–473 (1991)
14. Jensen, C.S., Snodgrass, R.T., Soo, M.D.: The tsq2 data model. In: The TSQL2 Temporal Query Language, pp. 153–238 (1995)
15. Matthé, T., De Tré, G.: Bipolar query satisfaction using satisfaction and dissatisfaction degrees: Bipolar satisfaction degrees. In: Proc. of the ACM SAC 2009 Conference, Honolulu, Hawaii, USA, pp. 1699–1703 (2009)
16. Nagypál, G., Motik, B.: A fuzzy model for representing uncertain, subjective, and vague temporal knowledge in ontologies. In: Chung, S., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 906–923. Springer, Heidelberg (2003)
17. Nascimento, M.A., Eich, M.H.: Decision time in temporal databases. In: Proceedings of the Second International Workshop on Temporal Representation and Reasoning, pp. 157–162 (1995)
18. Ohlbach, H.J.: Relations between fuzzy time intervals. In: International Symposium on Temporal Representation and Reasoning, pp. 44–51 (2004)
19. Qiang, Y., Asmussen, K., Delafontaine, M., De Tré, G., Stichelbaut, B., De Maeyer, P., Van de Weghe, N.: Visualising rough time intervals in a two-dimensional space. In: Proceedings of 2009 IFSA World Congress/EUSFLAT Conference (July 2001)
20. Rowe, L.A., Stonebraker, M.: The Postgres Papers. University of California at Berkeley, Berkeley (1987)
21. Sarda, N.L.: Extensions to sql for historical databases. IEEE Trans. Knowl. Data Eng. 2, 220–230 (1990)

Implementation of X-Tree with 3D Spatial Index and Fuzzy Secondary Index

Sinan Keskin, Adnan Yazıcı, and Halit Oğuztüzün

Department of Computer Engineering,
Middle East Technical University, Ankara, Turkey
{e1594183,yazici,oguztuzn}@ceng.metu.edu.tr

Abstract. In spatial databases, traditional approach is to build separate indexing structures for spatial and non-spatial attributes. This article introduces a new coupled approach that combines a 3D spatial primary index and a fuzzy non-spatial secondary index. Based on tests with several types of queries on a meteorological data set, it is shown that our coupled structure reduces the number of iterations and the time consumed for querying compared with the traditional uncoupled one.

Keywords: spatial indexing, multidimensional data indexing, fuzzy indexing, X-tree.

1 Introduction

Spatial database management systems are optimized to store and query spatial objects. It is known that use of index structures is indispensable in spatial databases for good performance [1, 4, 7]. To work with fuzzy non-spatial attributes, traditional approach for indexing in spatial database management systems is to use separate indexes for spatial and non-spatial data [10, 11]. In such an uncoupled structure, there are three steps for querying: First step is querying the 3D spatial data in primary index, second step is querying fuzzy attributes in secondary index, and the last step is matching the results of the previous steps and producing the final result. The aim of this work is to develop a new approach that combines two separate the index structures for spatial and non-spatial data to improve querying performance. This new coupled index handles both the 3D spatial data and the values to be fuzzified, thus it is possible to operate spatial and fuzzy queries by using a two-layered but a single index structure.

To demonstrate the performance gains, two alternative index structures, the coupled one and the traditional uncoupled one, are built on a sample data set containing multidimensional meteorological data. In the uncoupled structure, spatial X-tree index is used for 3D spatial data and the B+Tree index, which is overlaid on the X-tree index, is used for measured attributes. In the coupled structure, there is an X-tree index handling both spatial 3D data and the measured attributes to be fuzzified. The meteorological measurements contain fuzzy values such as temperature, humidity and pressure. Accordingly, queries include fuzzy terms such as “hot”, “warm” or

“cold” temperature values. Therefore a fuzzification process is needed on attribute measurements. Experiments and tests include point, range and nearest neighbor queries, and iteration counts and time consumed for querying are used for measuring the performance of indexing mechanisms. The results have shown that the coupled index structure provides a remarkable performance gain compared with the uncoupled one, where primary indexing is spatial and secondary indexing is non-spatial.

The rest of paper is organized as follows. Section 2 includes the related background. Implementation details are presented in Section 3. Performance tests are described in Section 4 and finally, conclusions are explained in Section 5.

2 Background

In this section we provide an overview of the essential structure of the X-tree. In our work we use a base implementation of the X-tree supported by the XXL API [8].

2.1 X-Tree

An X-tree (eXtended node tree) is an index tree structure based on the R-tree for storing and efficient querying of multidimensional data [2]. It puts premium on preventing the overlapping of bounding boxes, a persistent problem in high dimensional data. In cases where nodes cannot be split without preventing overlap, the node split is deferred, resulting in extended variable-size nodes, called super-nodes [3, 6].

The X-tree can be viewed as a hybrid of a linear array-like and a hierarchical R-tree-like directory as shown Fig. 1.

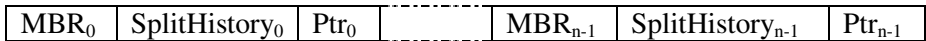


Fig. 1. Structure of a Directory Node [3]

The heterogeneous structure of the X-tree is illustrated in Fig. 2.

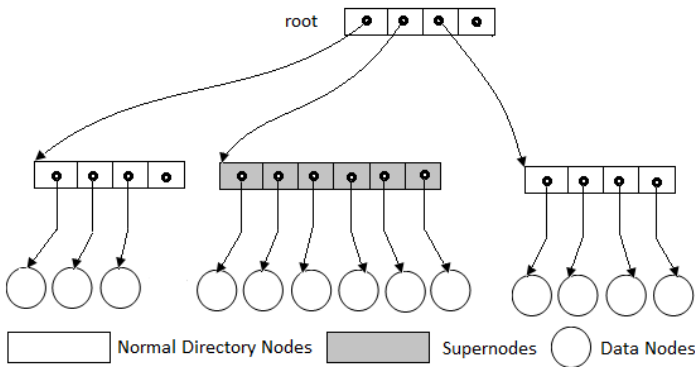


Fig. 2. Structure of the X-tree [3]

There are three different types of nodes in the X-tree:

- **Data Nodes:** These nodes contain rectilinear Minimum Bounding Rectangles (MBRs) together with pointers to the actual data objects.
- **Normal Directory Nodes:** These nodes contain MBRs together with pointers to sub-MBRs.
- **Supernodes:** These are large directory nodes with variable size. They are used mainly to avoid splits in the directory.

3 Implementation Details

In this section, it is described how to build an X-tree structure that contains both 3D spatial data and fuzzy data step by step. Then we present the structure of the uncoupled index structure for comparison.

3.1 Implementation of the X-Tree Structure

The major steps in this work are summarized as follows:

- Providing 3D spatial primary indexing on the X-tree by using point coordinates.
- Overlaying the non-spatial fuzzy secondary index on the X-tree by using meteorological attributes, thus yielding a coupled index structure.
- Building the uncoupled index structure that handles the 3D primary index and fuzzy secondary index separately
- Comparing the performance of the coupled and uncoupled index structures.

Handling 3D Rectangle in X-tree Node Structure. Our specifications require a node structure containing a 3D rectangle and meteorological attributes having values in a specified [min, max] range. Our X-tree is designed to have nodes containing both 3D rectangle data for spatial index and meteorological attributes for fuzzy secondary index.

In node structure of the X-tree, the meteorological attributes such as temperature, wind speed humidity and pressure have numerical values over an interval [min, max]. These values are allocated in a map structure to enable dynamical access.

Overlaying Secondary Index on the X-tree Structure. To handle primary and secondary indexes together, we overlay the fuzzy secondary index over the previously created X-tree structure.

We have three major steps of overlaying operation. These are (i) allocation 3D spatial data and fuzzy data in node, (ii) applying Fuzzy C-Means (FCM) algorithm [5] on meteorological data to calculate fuzzy membership value for fuzzy secondary index, and (iii) traversing over the X-tree primary index to build secondary index.

Allocating Meteorological Attributes. The base implementation of the XXL API [8] contains only rectangular objects in creating X-tree nodes.

We have built X-tree nodes containing both MBRs and meteorological attributes. For this purpose, we made some modifications to the MBR objects allocating these

attributes. Not only MBRs but also directory nodes and supernodes should allocate them. Therefore, the secondary index on X-tree involves meteorological attributes on the tree structure.

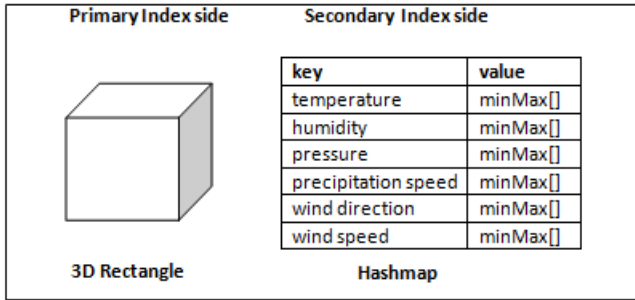


Fig. 3. 3D rectangle for spatial index and meteorological attribute for secondary index

To be specific, the X-tree node structure is extended with a hash map for meteorological attributes. The structure of this hash map contains a key for each attribute and two variables for the minimum and maximum values of that attribute as shown in Fig. 3.

Applying the FCM Algorithm. In our work, the FCM algorithm [5] is used to generate fuzzy values for secondary index by using meteorological attribute values. The algorithm is applied for each meteorological attribute of interest. A simple translation algorithm is developed to extract fuzzy values from the given input value of each meteorological attribute. The main advantages of this algorithm are its simplicity and speed, which allow it to run on large data sets.

In this step, first the entire data set was read from an input file and then the data related to each meteorological attribute were put on a separate set which is sorted in ascending order. Finally, fixed centroids were determined for each set. For example temperature values were ordered by ascending order and then fixed centroids in the name of cold, warm and hot were determined.

After we determined these three centroids, the FCM algorithm was applied for each meteorological attribute value to calculate fuzzy membership values. These membership values indicate the distance from each centroids. For example when we apply the FCM algorithm for 25 degree Celsius temperature, FCM generates [0.6, 0.3, 0.1] values which represent fuzzy membership for distance values from [Cold, Warm, Hot] centroids respectively. This means 25 degree Celsius is 0.6 far from cold.

We need fuzzy membership values like 25 degree Celsius is 0.11 cold. At this point, we apply the reverse ratio formula and determine the fuzzy membership values for [Cold, Warm, Hot].

Setting Meteorological Attributes of Each Node by Traversing the X-tree. In the creation step of the X-tree, only primary index that consists of spatial data is normally created by the insertion operation of the X-tree. After the primary index of the X-tree is created, there are no meteorological attributes in the directory node and supernode.

Only the data nodes have meteorological attributes. However at this point, the parent directory/super nodes of the data nodes have no meaningful information about their child nodes' meteorological attributes. So these meteorological attribute values should be arranged for creating our secondary index structure. In this work, a complete navigation should be done over all the data nodes, normal directory nodes or super nodes and also the root node. The min-max interval of child nodes' values are calculated and set to the parent node during the traversal.

The pseudo code of this recursive procedure is as follows:

```

Input: X-tree without secondary index
Output: X-tree with secondary index overlaid
1. If current_node.level > 0
    1.1. Initialize [min, max] values
        current_node[min] = MIN_INTEGER
        current_node[max] = MAX_INTEGER
    1.2. For each child_node of the current_node
        1.2.1. Recursive call
        1.2.2. If current_node[min] >= child_node[min]
            Then current_node[min] := child_node[min]
        1.2.3. If current_node[max] <= child_node[max]
            Then current_node[max] := child_node[max]
2. If current_node.level = 0
    2.1. Return [min, max] values of current_node
    
```

This operation starts executing from root node. Execution steps of this algorithm is shown in Fig. 4.

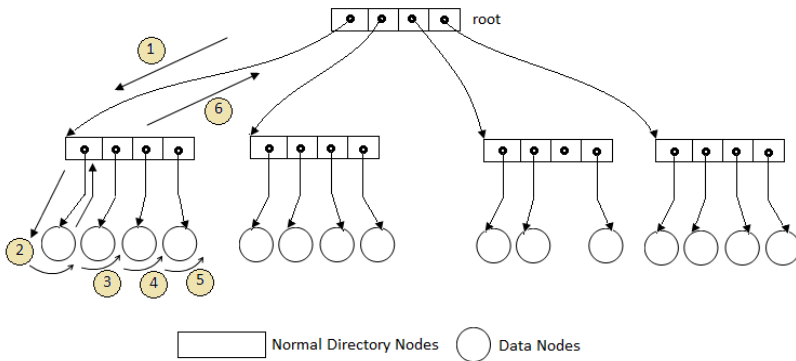


Fig. 4. Sequence of traversal on the X-tree

Building The Uncoupled Index Structure. By using each record in the input file we created a 3D spatial MBR object and stored it in the X-tree. We also created a fuzzy attribute object by using meteorological data and inserted it in the B+Tree. Finally we had two separate indexes, one of them is X-tree as primary index and the other one is B+Tree as secondary index as shown in Fig. 5. The basic steps of uncoupled index creation are (i) reading a record from input file and generate a unique ID for this

record, (ii) creating an MBR object by using spatial data, putting the MBR and the ID in a X-tree key object then inserting it to the X-tree, (iii) generating the fuzzy membership value by using meteorological attribute value via FCM and creating B+Tree object by using this fuzzy value and the ID then inserting it to the B+Tree.

X-tree Index Creation for 3D Spatial Primary Index. In this case we had simple X-tree that only contains 3D spatial MBR objects as shown in Fig. 5 (a). To create the X-tree, a data format as <latitude, longitude, altitude> was used to process our data set.

B+Tree Index Creation for Fuzzy Secondary Index. For secondary indexing we created B+Tree structure as shown in Fig. 5 (b). Before we inserted <temperature> data to the B+Tree index, we applied the FCM algorithm to each inserted record for getting the fuzzy membership values of related temperature data. Then, we inserted the fuzzy membership values to the B+Tree index. In coupled approach, secondary index was overlaid on the primary index by setting min and max fuzzy values. Note that the min and max values are not available for the secondary index as it is built independently.

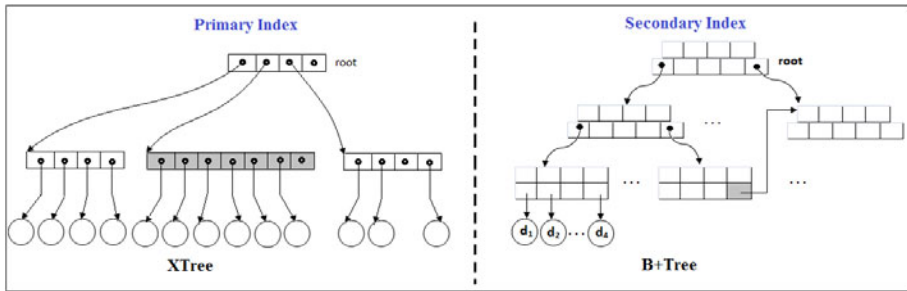


Fig. 5. The uncoupled index: (a) X-tree for primary and (b) B+Tree for secondary index

Management of 3D Spatial Index and Fuzzy Secondary Index. A management algorithm was implemented to manage two separate indexes and pseudo code is given below.

We firstly queried on the X-tree primary index by using 3D spatial parameters and we stored the results in a list. Then B+Tree secondary index was searched by using meteorological attribute parameter and the results were also stored in another list.

Input: Primary index root and Secondary index root

Output: A set of query result

1. Put X-tree query result to set S_p
2. Put B+Tree query result to set S_s
3. For each element E_p in S_p
 - 3.1. For each element E_s in S_s
 - 3.1.1. If $E_p = E_s$
 - 3.1.1.1. Put element to result set S_r

Finally, the elements in both lists are put in the list holding the result of the query.

4 Performance Tests

Test cases were performed to reveal the performance characteristics of both coupled and uncoupled approaches.

We noted the elapsed time values and iteration counts to measure the performances of these two approaches. Elapsed time values give an idea about query response time of each index structure and the iteration counts indicate their I/O performance. In each iteration, a node of index is fetched to see if it is proper for the query or not. A fetch operation incurs one unit of I/O cost for the index structure.

In the coupled index structure tests we only observed total execution time and total iteration count on the X-tree as this structure was monolithic. However, in the uncoupled index structure, we could observe the query time and the iteration count on primary and secondary indexes separately.

4.1 Point Query

This query is used for finding a specific point in index structure. It requires the capability of searching fuzzy attribute values on the secondary index. In point query, we generated 3D rectangle by using 3D point as in insertion operation.

A sample Point Query follows:

- Find the weather station located in the <10, 20, 30> coordinates where temperature is hot (fuzzy memberships are [cold, warm, hot] for temperature and query condition is hot membership value is greater than 0.01).

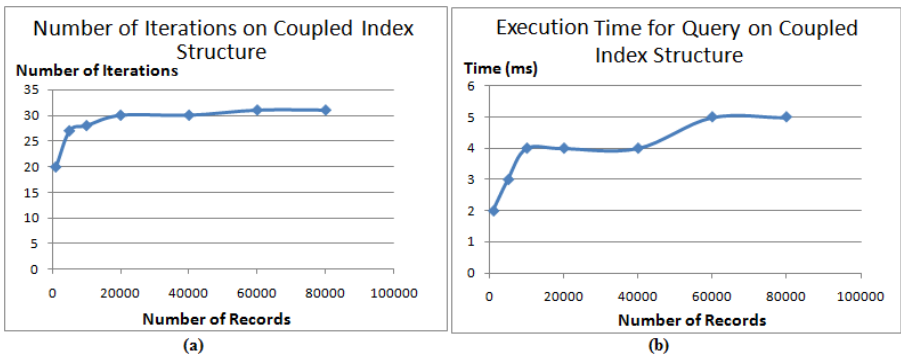


Fig. 6. Point query number of iterations and execution time charts for the coupled index

We generated the number of iterations and execution time charts for coupled index as in Fig. 6 and comparison charts between the coupled and uncoupled indexes as in Fig. 7.

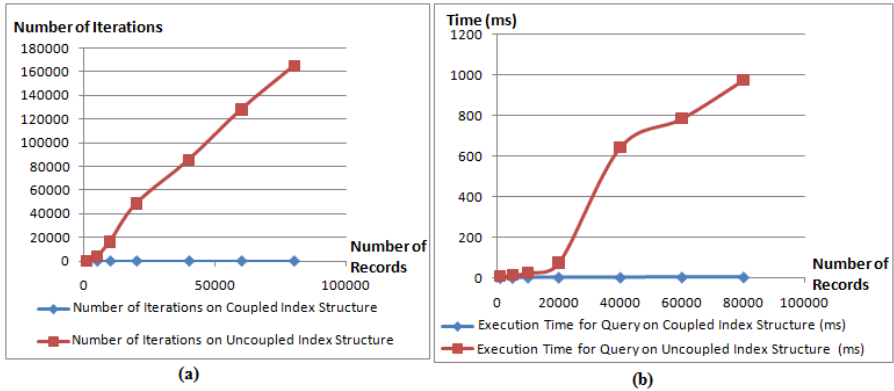


Fig. 7. Point query number of iterations iteration and execution time charts. Chart (a) shows number of iterations for coupled and uncoupled index structure, (b) shows execution time for coupled and uncoupled index structure.

By analyzing these charts, we can make the following statements:

- The cost of searching in coupled index structure in terms of iterations is logarithmic in point query. This is because the X-tree is height-balanced.
- The behavior of uncoupled index structure is linear in point query. The number of iterations required for the secondary index is increasing proportionally to the inserted record count in uncoupled index structure. In addition, the number of iterations done for match operation is also increasing linearly. As a result, these types of increases make the behavior linear.

Note that the charts for the elapsed time values of the coupled and the uncoupled indexes are similar to the charts for the iteration counts. In conclusion, we can make the same statements for the elapsed time values.

Table 1. Coupled and uncoupled index structures point query observed iteration count table

Number of Inserted Records	Coupled Index Structure	Uncoupled Index Structure			
	Total Iteration	Primary Index Iteration	Secondary Index Iteration	Iteration for match operation	Total Iteration
1000	20	20	37	167	224
5000	27	27	180	3669	3876
10000	28	28	424	15883	16335
20000	30	30	1024	47371	48425
40000	30	30	2087	83314	85431
60000	31	31	3241	124807	128079
80000	31	31	4357	160360	164748

On uncoupled index we applied three steps on querying: first step is the spatial query on X-tree, second step is the fuzzy query on B+Tree and the last step is the matching operation to find the final result set. The cost of retrieval using the X-tree is logarithmic, the cost of retrieval using the B+Tree is also logarithmic and the intersection of these two sets is proportional to product of the cardinalities of these sets. But we should remark that the cost of matching the spatial result set and the fuzzy result set is huge as seen in Table 1. Finally, logarithmic behavior closes to the linear behavior because of this matching operation.

4.2 Range Query

Range query is used for searching spatial elements in a given rectangular range covering lower and upper boundaries. If any element in the index structure provides searched spatial and fuzzy conditions, it is placed to the result set.

An example Range Query follows:

- Find the stations that are covered by the rectangle with lower boundary $\langle 10, 20, 30 \rangle$ and upper boundary $\langle 20, 30, 40 \rangle$ coordinates and it is temperature is hot (query condition is hot membership value is greater than 0.01).

In our experiments and tests, in addition to the range queries we also checked the element’s fuzzy attribute by using fuzzy secondary index. We also generated comparison charts between coupled and uncoupled indexes as they are shown in Fig. 8.

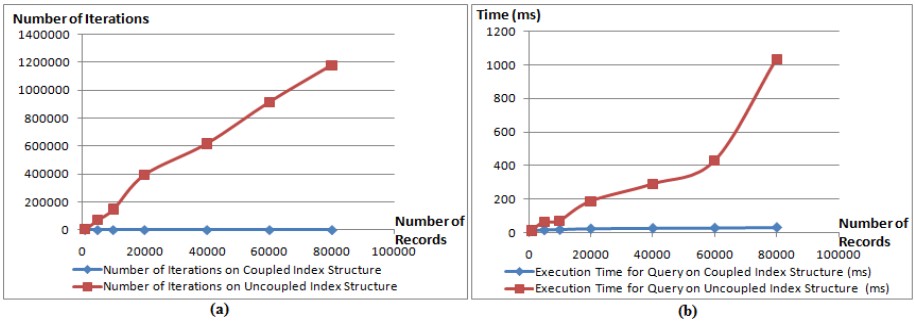


Fig. 8. Range query number of iterations and execution time charts. Chart (a) shows the number of iterations for coupled and uncoupled index structure, (b) shows execution time for coupled and uncoupled index structure.

Based on these charts make the following statements:

- The behavior of coupled index structure is logarithmic in range queries with respect to the iteration counts as the X-tree has a balanced structure.
- Uncoupled index structure performance is linear in range queries. Number of iterations for querying on the secondary index and matching operation are increasing in proportion to the inserted record counts.

According to the execution time over growing number of inserted record shown in Fig. 8 (b), we can remark that the behavior of range query on uncoupled index structure is linear. However coupled index has logarithmic behavior.

4.3 Nearest Neighbor Query

The Nearest Neighbor (NN) query is used for searching closest spatial elements in a metric space M . Given a point q in M the NN query searches for a prescribed number of points that are closest to q .

In our tests, an NN query against the tree determining the 10 nearest neighbor entries at the target level concerning the input rectangle is performed. In addition we tested a fuzzy condition in querying.

An example NN query is:

- Find 10 nearest stations to the given station which is based in $\langle 10, 20, 30 \rangle$ coordinates and its temperature is hot (query condition is that hot membership value is greater than 0.01)

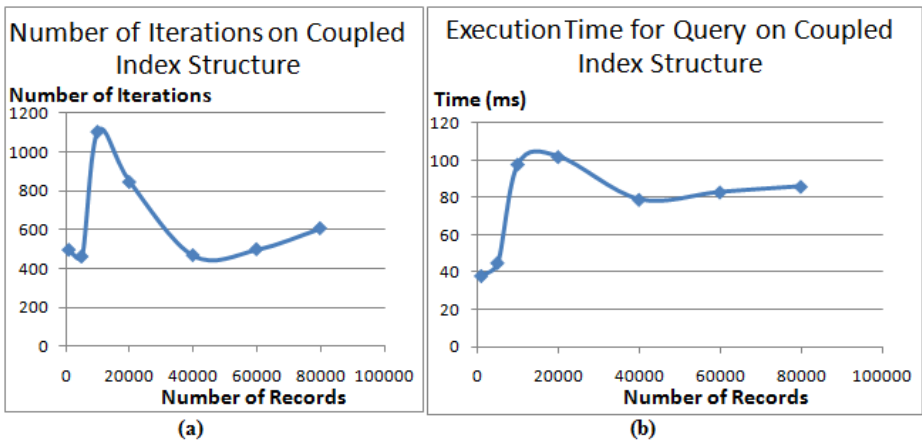


Fig. 9. Range query number of iterations and execution time charts for the coupled index

Number of total iterations and execution time with respect to increasing number of inserted record charts are shown in Fig. 9.

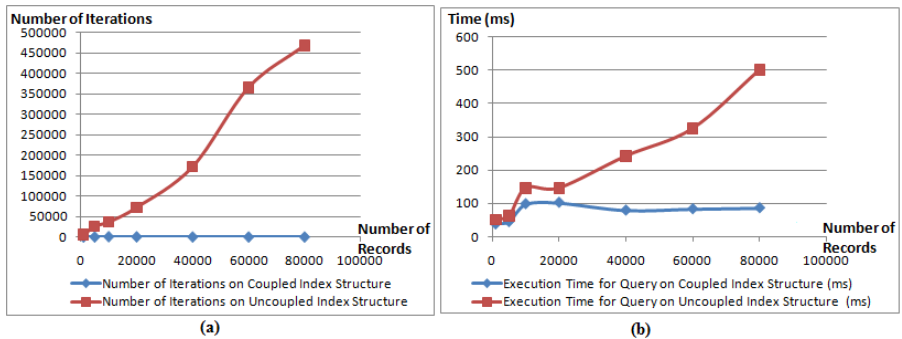


Fig. 10. Nearest Neighbor query charts. Chart (a) shows number of iterations for coupled and uncoupled index structure, (b) shows execution time for coupled and uncoupled index structure.

By analyzing these charts we can argue that

- Coupled index structure has no coherent behavior in nearest neighbor query, because observed iteration count is related to distance-based search. Cost is dependent on the point density within the searched boundary.
- On the other hand uncoupled index structure has linear behavior in NN query because of matching the results of primary and secondary indexes.

Execution time behaviors observed on the comparison charts are essentially the same as those observed on the iteration charts. Again, the coupled index structure has no consistent relationship with the number of inserted records and also uncoupled index structure behavior is linear due to matching the results of primary and secondary indexes.

5 Conclusion

In our work, we focused on a new indexing approach that combines 3D spatial primary indexing and fuzzy secondary indexing in a monolithic structure. Firstly, we implemented a coupled index structure that comprises both primary index and secondary index in a single X-tree index structure. Secondly, we developed a pair of separate index structures in the uncoupled approach. In this approach, the X-tree still serves as the index for 3D spatial objects, but the B+Tree is employed for indexing fuzzy meteorological attributes. Then we compared these two approaches with respect to insertion and some common types of queries to understand their performance characteristics.

With respect to insertion, the uncoupled index structure was more efficient than coupled index structure because of the cost of overlaying the secondary index on the X-tree. On the other hand, querying the coupled index structure was more efficient than uncoupled index structure where we had to run the query in two different indexes and intersected the two result sets with the match operation. The queries used for comparing coupled and uncoupled index structures contain the search operations on the values of attributes within a spatial region. Coupled index structure offers advantage in queries that firstly search for a spatial property and then for a non-spatial property within the region of interest. Therefore we have demonstrated that the coupled index structure is more efficient than the uncoupled one for the types of queries considered in this work. Further research is needed to investigate performance characteristics of the coupled approach for other types of queries.

References

1. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1990), pp. 322–331 (1990)
2. Auxiliary Data Structures, <http://undergraduate.csse.uwa.edu.au>
3. Berchtold, S., Keim, D.A., Kriegel, H.-P.: The X-tree: An index structure for high-dimensional data. In: Proceedings of the 22th International Conference on Very Large Data Bases (VLDB 1996), pp. 28–39 (1996)

4. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1984), pp. 47–57 (1984)
5. Cannon, R.L., Dave, J.V., Bezdek, J.C.: Efficient Implementation of the Fuzzy c-Means Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 248–255 (1986)
6. Nam, B., Sussman, A.: A Comparative Study of Spatial Indexing Techniques for Multidimensional Scientific Datasets. UMIACS and Dept. of Computer Science, pp. 3–4. University of Maryland (2002)
7. Yang, G., Zhang, J.: A Dynamic Index Structure for Spatial Database Querying Based on R-trees, pp. 1–3. Institute Remote Sensing Applications, Chinese Academy of Sciences, Chinese Academy of Surveying and Mapping, Beijing, P.R.China
8. XXL API, <http://www.xxl-library.de/>
9. Weiss, M.A.: *Data Structures and Algorithm Analysis in C++ 3/E*. Addison-Wesley, Florida (1996)
10. Zhang, J., Pan, H., Yuan, Z.: A novel spatial index for case based geographic retrieval. In: International Conference on Interaction Sciences, Seoul, Korea, pp. 342–347. ACM, New York (2009)
11. Martins, B., Silva, M.J., Ribeiro, L.A.: Indexing and ranking in geo-ir systems. In: CIR, Bremen, Germany, pp. 31–34. ACM, New York (2005)

Semantic Processing of Database Textual Attributes Using Wikipedia

Jesús R. Campaña, Juan M. Medina, and M. Amparo Vila

Dept. of Computer Science and Artificial Intelligence, University of Granada,
Daniel Saucedo Aranda s/n, 18071 Granada, Spain
{jesuscg,medina,vila}@decsai.ugr.es

Abstract. Text attributes in databases contain rich semantic information that is seldom processed or used. This paper proposes a method to extract and semantically represent concepts from texts stored in databases. This process relies on tools such as WordNet and Wikipedia to identify concepts extracted from texts and represent them as a basic ontology whose concepts are annotated with search terms. This ontology can play diverse roles. It can be seen as a conceptual summary of the content of an attribute, which can be used as a means to navigate through the textual content of an attribute. It can also be used as a profile for text search using the terms associated to the ontology concepts. The ontology is built as a subset of Wikipedia category graph, selected using diverse metrics. Category selection using these metrics is discussed and an example application is presented and evaluated.

Keywords: Databases, text processing, ontologies, Wikipedia category graph, similarity metrics.

1 Introduction

Text attributes in databases contain useful information that is usually not processed nor stored anywhere in the database. Although text attributes are usually queried, these queries only take into account syntactic patterns and neglect any kind of semantics associated with terms. Processing text attributes we can obtain labels to tag the contents of the text according to its semantics, or compute a descriptive summary of the attribute.

In [2] we presented a method to extract semantic information from unstructured texts stored in databases. Our approach although general was mainly based on the use of WordNet [3] for semantic extension. WordNet is a valuable tool in natural language processing used to determine synonyms and syntactic relations between words. WordNet's taxonomy is designed from a linguistic point of view which makes it not very descriptive from a conceptual point of view. As the method presented is general in regard to the tools that can be used in each of the steps performed, we propose the use of Wikipedia as a tool to perform semantic representation and extension of unstructured texts.

Wikipedia is an online collaborative encyclopedia where users can generate and edit contents. Wikipedia pages are organized in categories, and categories

are connected between them using hierarchical relations. The structure of the categories and their connections it is known as the Wikipedia Category Graph. In [16] the Wikipedia category graph is extensively tested and it is shown that it is as a valuable tool as other electronic dictionaries. The authors also present some statistics comparing size and coverage of Wikipedia to that of WordNet. The results back up the use of Wikipedia category graph as a tool for natural language processing.

In this paper we present an approach to semantics extraction from unstructured texts in databases using Wikipedia. The goal is to obtain relevant sub-graphs of the whole Wikipedia category graph to represent terms extracted from text. The ontological representation of unstructured text is a summarised semantic representation of the contents of the text attribute. This taxonomy can be extended with search terms associated to each of the concepts. Using the terms associated to the concepts it is possible to compose queries to retrieve other texts related to the concepts in the query.

The paper is organized as follows. Section 2 presents the general process to follow in order to obtain an ontology from a text attribute stored in a database. Section 3 depicts details about the role of Wikipedia in the process presented before. The use of Wikipedia category graph and the way to use similarity metrics to select a coherent sub-graph to represent texts are discussed. The similarity and relatedness metrics employed are discussed in Sect. 4. An application and the evaluation of the results obtained using the different metrics are presented in Sect. 5. Finally Sect. 6 gives some conclusions and highlights future work.

2 Semantic Representation of Non-structured Texts

The whole process of extraction, processing, representation and extension of unstructured texts using Wikipedia is shown in Fig. 1. A detailed version of the steps performed before semantic extension can be seen in [2].

The generation of an ontology from non-structured text using Wikipedia, comprises a series of basic steps.

- *Selection* of the text attribute to process from the database.
- *Syntactic Preprocessing* of data using the appropriate filters (tokenization, stop word removal, stemming, etc).

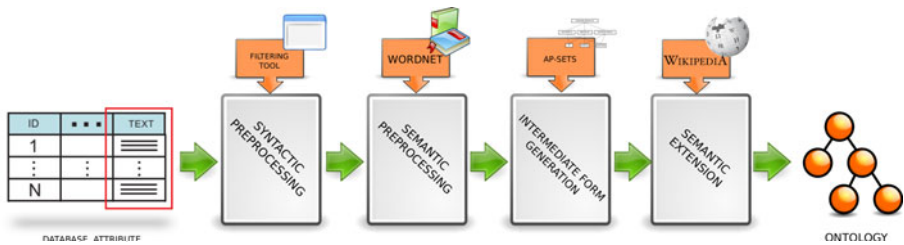


Fig. 1. Wikipedia based semantic representation of non-structured texts

- The *Semantic Preprocessing* step comprises different tasks that can be performed with a wide variety of tools, along with the task we cite the tool we have decided to use in each case. POS Tagging (using The Stanford Natural Language Processing Group POS Tagger [17]), word sense disambiguation (using Adapted Lesk Algorithm [1]), and synonym substitution (substitution of the original words for the term selected as canonical representative of a WordNet *synset*).
- *Generation of an Intermediate Form* for knowledge representation. Once the pre-processing stage is finished data must be expressed as an intermediate form. This form can be expanded and used to generate the final ontology. In our case we choose to represent data using the AP-Sets [9,10] intermediate form. An AP-Set is a lattice like structure for a maximal itemset found in text. An AP-Structure is a set of AP-Sets. When processing texts with this technique we obtain two different structures, a domain AP-Structure defining the general domain for a text attribute, and a set of sub-AP-Structures (one for each tuple) defining the text content of a tuple. The domain AP-Structure is a lattice structure summarising the text content of an attribute based on statistical properties. This structure can be seen as a basic inclusion ontology.
- *Semantic Extension of the Intermediate Form*. The chosen intermediate form is extended with the aid of external resources. In our case we use Wikipedia, specifically the Wikipedia category graph.

The remainder of this paper is devoted to analyse different ways to extend the AP-Structure obtained from texts using the Wikipedia category graph.

3 Semantic Extension Using Wikipedia

When selecting a knowledge source to extend the AP-Structure we must take into account that the source needs to be structured, well defined, and designed from a practical point of view. Lots of ontologies present these requirements but they lack of proper maintenance and frequent updates. We choose the Wikipedia category graph as a tool for the task of extending an AP-Structure into an ontology.

3.1 Wikipedia Category Graph

A Wikipedia category is a special type of page used to group other regular pages related to a common subject. Each page of Wikipedia it is associated with one or more categories. Any category may branch into subcategories, and all categories have at least one parent category except for the top-level category, Category:Contents. The Wikipedia category graph approximates a directed graph containing cycles.

Categories are connected to each other by relations of hyponymy/hypernymy. All relations are defined by Wikipedia editing users. The Wikipedia graph has a similar tree structure as that of WordNet, and can be seen as a thesaurus

combining collaborative tagging and hierarchical indexation. The main drawback of Wikipedia category graph it is the existence of cycles.

The Wikipedia category graph G_W is defined as a directed graph $G_W = (V, A, r)$, where V is a set of vertexes, each vertex maps to a category in Wikipedia, A is a family or arcs defining hierarchical relation between categories, and r represents the main Wikipedia category, the root node of G_W .

A Wikipedia page is defined as $p = (t, c, L, LC)$, where t is the title, and c is the page content, $L = \{l_1, l_2, \dots, l_n\}$ is the set of links referencing other pages and $LC = \{lc_1, lc_2, \dots, lc_n\}$ is the set of link referencing the categories on which the page is classified.

Figure 2 shows a Wikipedia page with all the components and the way they are related to the Wikipedia category graph. A node in the category graph represents a category with links to its parent categories and a set of pages belonging to the category.

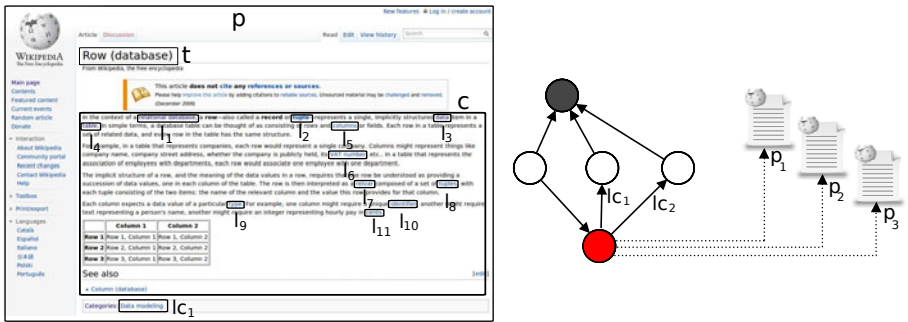


Fig. 2. Components of a Wikipedia page

Each category contains information about the pages on it, therefore a category can be defined by its pages. Each page can be seen as a label describing a topic related to the category.

A Wikipedia category $C = (P, LC)$ is defined as a set of Wikipedia articles or pages $P = \{p_1, p_2, \dots, p_n\}$ and a set of links to its parent categories $LC = \{lc_1, lc_2, \dots, lc_n\}$.

The Wikipedia category graph is a complex structure, its huge size and the high level of interconnection between its nodes makes its automatic processing a non-trivial task. If we get all the connections for a given random term, selecting all the categories to which it is connected and the ones connected to them recursively, the chance of ending up with a graph with a similar size to the whole Wikipedia graph it is very high.

We intend to create an ontology where AP-Structure nodes/itemsets are contextualized. This ontology is going to be used by human users, so the size must be relatively small in order ease comprehension. To do this, we propose to select a sub-graph of the whole Wikipedia to properly define a node taken from an AP-Structure.

In order to select the related Wikipedia categories we need to compare them. Our work is focused on exploiting information using three different approaches:

Graph exploration: Starting from a initial category a path in the graph must be traversed while selecting categories.

Semantic structure: Using WordNet semantic similarity measures adapted to Wikipedia as in [16], it is possible to obtain similarity values between categories attending to topological criteria of the underlying structure. The similarity value obtained allows to discard categories loosely related to the initial ones. Getting only the related categories allows to create a structure coherent with the initial categories obtained from the nodes in the AP-Structure.

Content: Each of the Wikipedia categories has a set of pages. These pages contain a title that can be used as an identifier. Categories can be evaluated taking into account the amount of shared pages.

3.2 Semantic Extension of AP-Sets Using Wikipedia

In order to extend the AP-Structure we must map each node in the basic inclusion hierarchy to a Wikipedia category. When all categories are identified, a path is traversed and the appropriate categories are selected.

In order to select the appropriate sub-graph of Wikipedia category graph using similarity measures we perform the following steps:

- For each node in the AP-Structure a category in the graph must be selected.
- If it is not possible to find a category for the node, search for a page containing the terms in the title. For compound terms, search for each of its components i.e. if no page is found for *Fuzzy Relational Database*, search for *Fuzzy Database* and *Relational Database*.
- Select the categories to which the pages are associated and use them as base categories or start categories. From these categories we start exploration of the graph.
- While traversing the graph the categories in the path are compared. As we will see later the comparison can be performed in different ways, but always return a value. A threshold t is previously set. Each evaluated category whose value it is greater than the threshold is included in the final ontology.
- The process is the same for each node. Once all nodes are processed, all the sub-graphs obtained are merged.

There are two important aspects that we have not mentioned until now, the way in which comparison is performed and the scope of the comparison. The particular way of comparing categories and the obtained results will depend on the measure employed, so we will deal with it when describing the different measures. Regarding the scope of the comparisons performed. That is, which categories to compare, we distinguish the three following scopes:

Comparison with the start category: This is the basic method, comparisons are performed with the start category while traversing the graph.

As we walk the graph towards the root each new category found is compared to the start category. If the value for the comparison is equal or greater than a previously set threshold t , the category is selected and added to the final result. If the comparison value is below the threshold the category not only is not added to the result, but the exploration of that branch stops.

Comparison with the current category: This comparison scope compares the new accepted categories to the ones closest to them. If a new category is selected that means it is relevant, and thus it acts as a new start category. Using this comparison scope, categories are compared only to their parent categories. This allows to update the context in each new step.

Comparison to the aggregated category: If the previous comparison scope took decisions on a local level, this new scope uses a context to take decisions on a global level. Each category accepted is added to the context. Each time a new category is going to be added, it is compared to the aggregated category acting as context. This way, the decision of selecting a category depends on all the categories previously selected.

The main problem with this approach is that each time a new category is added to the aggregated category, it becomes more general. Due to this increasing generality each time it is easier to select a new category.

4 Measures for Category Selection

We will use different kinds of measures to select categories as the graph is traversed. Topology based measures just evaluate the position of the categories in the graph, while content based measures need more processing.

4.1 Statistical Indexes

In order to compare categories we characterize each category as a set of pages. Each page is represented by its title. The title is processed as a string identifier and hence it is not split into words. Figure 3 shows a comparison example between two categories C_1 and C_2 , where each category has a set of pages $C_1 = \{p_a, p_b, p_c\}$ and $C_2 = \{p_b, p_d\}$. Each page has a title, the set of titles obtained from the pages in C_1 is labelled as A , and the set of titles obtained from pages in C_2 is labelled B . When a statistic index is applied over the two sets A and B representing categories to compare, it returns a similarity value between both sets and hence between the two categories.

In general any set based similarity measure can be used with this approach, but we use it with known measures as Jaccard Similarity Coefficient [5], Sorensen Similarity Coefficient [15] and Mountford Index of Similarity [11].

The evaluation of these measures show that as the distance to the start category increases, similarity decreases. It is important to stress that this measures are based only in the categories content, but we can see how this behaviour is coherent with the topological measures. These results reinforce the idea that the farthest two categories are the less similar they are.

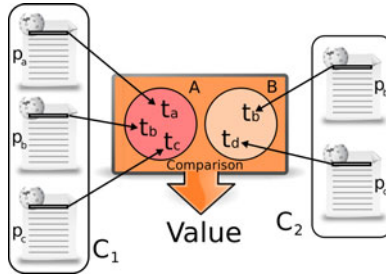


Fig. 3. Category comparison using statistical indexes

When tests are performed on statistical measures changing the comparison scope to the current category, the results highlight an interesting behaviour. Although the comparison is performed between parent-child categories the values obtained tend to decrease as we get closer to the root. This phenomenon is due to the increasing generality of categories as we approach the root node. This means that leaf nodes have more meaning than nodes closer to the root.

Comparison to the aggregated category presents a good behaviour at first as allows to explore the immediate vicinity and jump to other general categories. However, this poses an inconvenient as the upper parts of the graph are highly connected. In this zone each new category added makes easier the acceptance of new categories, therefore it is possible to end up with a very big sub-graph.

The threshold value allows us to control the size of the final ontology. But we must be very careful as a low threshold value can induce a massive selection of categories located close to the root. This results in bigger ontologies with a lot of non relevant classes.

4.2 Semantic Similarity Measures

The next group of measures to test are semantic similarity measures. We have selected the most relevant ones for study, Rada et al. Path-length [12], Leacock-Chodorow [7] and Wu-Palmer [18]. This are all semantic similarity measures and rely on the hypernym/hyponym structure of the graph. These measures do not use additional information from the categories, they rely only in information encoded in the category graph structure.

Conceptually, the adaptation of these measures to use with Wikipedia category graph is trivial, we just need to apply them directly over the category graph. In practice this task is complex because the measures are designed to work with trees, so we must deal with multiple choices for a path between nodes and cycles.

These measures are created to determine the similarity between any two nodes in an ontology, based only in the structure's topology.

In our case similarity measures do not provide meaningful information because while traversing the graph we ensure a degree of semantic similarity.

Data obtained confirm that although the topological characterization of two nodes is equivalent, their semantics are not.

4.3 Information Content Measures

Resnik information content (IC) measure [13] combines information obtained from the taxonomic structure with probabilistic estimations for concepts, obtained empirically from a huge text corpus. Similarity is modelled as the degree in which concepts share information and IC is computed as the information content of its lowest common subsumer (LCS). Other relevant information content measures are Lin Similarity Measure [8] and Jiang-Conrath Distance [6] reformulated as a relatedness measure. Both measures use the idea of information content presented by Resnik.

Seco et al. [14] propose a variant to Resnik information content relying only on Wordnet for information content calculation without using any external corpus. The idea is that general concepts have less information content than those more specific. The way of computing generality in a tree is determining the number of hyponyms (concepts subsumed) by a concept, weighted by the total size of concepts in the taxonomy. This reformulation of the information content is equivalent to the Resnik formulation if the frequency of each term is 1. Both definitions yield similar values, which means that information about frequency in an external corpus is not as important as topological information, particularly information concerning the depth of the concept in the taxonomy.

In order to compute values for information content measures we rely in the JWPL (Java Wikipedia Library) API [19]. JWPL is an open source application programming interface that enables Wikipedia information access. This API can build a hierarchy from the Wikipedia category graph removing cycles and redundant paths in order to compute similarity values using information content measures. The measures implemented in JWPL use the IC definition of Seco. As the graph is highly connected most of the nodes have low IC values.

5 Method Selection

One criteria to determine the utility of a method against another is the computation of the correlation of its evaluations with human judgements. Semantic relevance of the methods is tested using WordSimilarity-353 [4]. WS-353 contains 353 pairs of terms and relatedness judgements provided by humans. Measuring correlation between the relatedness values provided for the pairs by humans and those provided for the methods evaluated, we can determine which makes judgements closer to that provided by humans. As WS-353 contains evaluations over pairs of words and we evaluate pairs of categories, we must adapt the procedure. When determining a category for a term we must determine a page for that term and then select a category from the page. This process introduces some ambiguity as a term can have more than one sense. In order to avoid the noise introduced by the disambiguation procedure we select an unambiguous subset of WS-353 containing 209 term pairs (WS-209). Another source of ambiguity is the category selection from the page, this is avoided using all categories for the pages and doing pairwise comparison, we take the average value of the comparisons performed and the best value. Evaluation of semantic similarity measures and

information content measures is performed using an extension over JWPL API. Correlation values obtained are computed using Pearson correlation coefficient and are presented in Table II. All the tests have been conducted over a dump of the english Wikipedia for August the 17th 2010, stored in a MySQL database.

Table 1. Correlation for the Methods

Method	Correlation WS-209
Statistical Indexes	
Average Jaccard	0,2559
Best Jaccard	0,2993
Average Sorensen	0,2741
Best Sorensen	0,3122
Average Mountford	0,2303
Best Mountford	0,2841
Semantic Similarity Measures	
Average Rada	-0,2974
Best Rada	-0,2942
Average Leacock-Chodorow	0,3358
Best Leacock-Chodorow	0,3635
Average Wu-Palmer	0,2764
Best Wu-Palmer	0,2635
Information Content Measures	
Average Resnik	0,2670
Best Resnik	0,3032
Average Lin	0,2760
Best Lin	0,3113
Average Jiang-Conrath	-0,0565
Best Jiang-Conrath	0,1079

In order to compare the different approaches we propose an application to perform concept querying using text terms. The application generates a structured representation of our research interests, which are stored in a database in the form of research article titles. This application generates a conceptual search profile from a set of texts in the form of a taxonomy-like ontology. This conceptual profile will be used to retrieve conceptually similar texts from other sources, although it also can be used to navigate through the original texts.

We extract frequent itemsets from unstructured text and build a taxonomy-like ontology, where each concept in the ontology is annotated with search terms extracted from WordNet. These search terms provide a way to search for a concept over a set of unstructured texts.

In this example we select a set of texts containing 500 research articles written by researchers of the Computer Science department at the University of Granada. The texts are syntactically preprocessed and a subset of 30 titles are randomly selected. With the selected titles we perform semantic preprocessing, and compute the domain AP-Structure. After processing the selected subset the AP-Structure contains the following maximal itemsets: *Genetic Algorithm*, *Neural Network*, *Programming*, *Database* and *Software*. Each of the terms are mapped to a Wikipedia category and then the methods are applied.

To test and evaluate the methods we perform conceptual searches using the ontologies generated. For each concept in the ontology we perform a query over

the original set of 500 titles using its associated search terms. As the context is the same for all texts, we can compare the different approaches using recall over the original set. As can be seen in Table 2 the recall increases with the number of nodes. If we evaluate the methods using a ratio between recall and the number of nodes, those methods closest to the baseline get better values, so we compute the ratio between gain in recall from the baseline and the new nodes added. We have selected the threshold value providing better results.

Table 2. Evaluation Results

	Nodes	Max depth	Tangledness	Recall	Recall/Nodes	Gain
Baseline	6	1	0	0,196	0,0326	0
Baseline + WN	6	1	0	0,26	0,0433	0
Jaccard Start (t=0.015)	15	4	0,2	0,342	0,0228	0,0091
Jaccard Current (t=0.015)	45	14	0,29	0,494	0,0109	0,0060
Jaccard Aggregated (t=0.015)	18	6	0,22	0,342	0,0190	0,0068
Sorensen Start (t=0.03)	15	4	0,2	0,342	0,0228	0,0091
Sorensen Current (t=0.03)	32	7	0,25	0,442	0,0138	0,0070
Sorensen Aggregated (t=0.03)	21	6	0,29	0,342	0,0162	0,0054
Mountford Start (t=0.002)	9	3	0	0,266	0,0295	0,0020
Mountford Current (t=0.002)	10	3	0,1	0,266	0,0266	0,0015
Mountford Aggregated (t=0.002)	10	3	0,1	0,266	0,0266	0,0015
Rada (t=0.5)	22	3	0,27	0,286	0,0130	0,0016
Leacock-Chodorow (t=0.6)	22	3	0,27	0,286	0,0130	0,0016
Wu-Palmer (t=0.6)	42	15	0,36	0,568	0,0135	0,0085
Resnik Start (t=0.6)	6	1	0	0,26	0,0433	0
Resnik Current (t=0.6)	6	1	0	0,26	0,0433	0
Lin Start (t=0.6)	6	1	0	0,26	0,0433	0
Lin Current (t=0.6)	6	1	0	0,26	0,0433	0
Jiang-Conrath Start (t=0.6)	6	1	0	0,26	0,0433	0
Jiang-Conrath Current (t=0.6)	6	1	0	0,26	0,0433	0

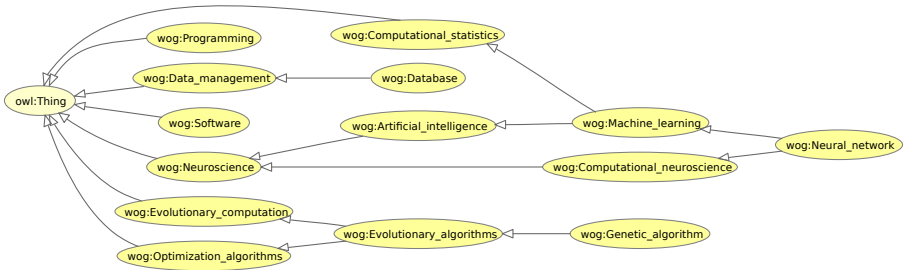


Fig. 4. Ontology generated using Sorensen Similarity Coefficient

Our choice for a method to use in this basic application scenario would be the Sorensen Similarity Coefficient using a comparison with the start category. This measure presents a good correlation value and the best gain together with the Jaccard index. The ontology generated by the method is shown in Fig. 4.

The selection of the method to use in each case may vary as different applications require different kinds of ontologies and thus our selection may change according to the specifics of each problem.

6 Conclusions and Future Work

In this paper we have presented a proposal for taxonomy-like ontologies generation using Wikipedia category graph and different measures. The ontologies obtained are annotated with search terms in order to retrieve texts containing the concepts searched. This ontologies can be seen as summaries for text content in databases or can be used as search profiles.

Future work will deal with the use of information retrieval measures to determine similarity between the categories textual content, in order to provide more complete ontologies. Additional relations to enrich the hierarchy can be obtained from Wikipedia and WordNet and other knowledge sources can be used in addition to Wikipedia. Our focus is also in the automatic evaluation of the obtained ontologies, developing new metrics and procedures to ensure the quality of the ontologies obtained and providing means to compare ontologies.

Acknowledgment. This work has been partially supported by the Spanish “Ministerio de Ciencia e Innovación” (MCYT) under grant TIN2006-07262/ and the “Consejería de Economía, Innovación y Ciencia de Andalucía” (Spain) under research projects P06-TIC-01433 and P06-TIC-01570.

References

1. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) CILCling 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
2. Campaña, J.R., Martín-Bautista, M.J., Medina, J.M., Vila, M.A.: Semantic Enrichment of Database Textual Attributes. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 488–499. Springer, Heidelberg (2009)
3. Fellbaum, C.: WordNet: an electronic lexical database. MIT Press, Cambridge (1998)
4. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppín, E.: Placing search in context: The concept revisited. *ACM Transactions on Information Systems* 20(1), 116–131 (2002)
5. Jaccard, P.: Etude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 547–579 (1901)
6. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the 10th International Conference on Research in Computational Linguistics, Taipei, Taiwan, pp. 19–33 (1997)
7. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification, ch. 11, pp. 265–283. The MIT Press, Cambridge (1998)
8. Lin, D.: An information-theoretic definition of similarity. In: Shavlik, J.W. (ed.) Proc. 15th International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
9. Marín, N., Martín-Bautista, M.J., Prados, M., Vila, M.A.: Enhancing Short Text Retrieval in Databases. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 613–624. Springer, Heidelberg (2006)

10. Martín-Bautista, M.J., Martínez-Folgoso, S., Vila, M.A.: A New Semantic Representation for Short Texts. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 347–356. Springer, Heidelberg (2008)
11. Mountford, M.D.: An index of similarity and its application to classification problems. In: Murphy, P.W. (ed.), pp. 43–50 (1962)
12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics* 19(1), 17–30 (1989)
13. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 448–453 (1995)
14. Seco, N., Veale, T., Hayes, J.: An intrinsic information content metric for semantic similarity in WordNet. In: Proceedings of ECAI-2004, vol. 16, pp. 1089–1090 (2004)
15. Sorensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Dan. Vidensk. Selsk. Biol. Skr.* (5), 1–34 (1948)
16. Torsten Zesch, I.G.: Analysis of the wikipedia category graph for nlp applications, pp.1–8 (2007)
17. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003, pp. 252–259 (2003)
18. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138. Association for Computational Linguistics, Morristown (1994)
19. Zesch, T., Gurevych, I.: Wisdom of crowds versus wisdom of linguists—measuring the semantic relatedness of words. *Natural Language Engineering* 16(01), 25–59 (2010)

Querying Class-Relationship Logic in a Metalogic Framework

Jørgen Fischer Nilsson

DTU Informatics
Technical University of Denmark
jfn@imm.dtu.dk

Abstract. We introduce a class relationship logic for stating various forms of logical relationships between classes. This logic is intended for ontologies and knowledge bases and combinations thereof. Reasoning and querying is conducted in the DATALOG logical language, which serves as an embracing decidable and tractable metalogic.

Keywords: Querying knowledge bases and ontologies. DATALOG as metalogic. Analytic vs. synthetic knowledge.

1 Introduction

We address a two-level logic for ontologies and knowledge bases dealing with relationships between classes. The proposal is distinguished by encoding of the applied class relationship logic utilizing DATALOG as a formal metalogic.

The proposed class-relationship logic, CRL, offers predicate logical sentences with quantifiers stating relationships between classes. We identify four elementary relationship forms corresponding to the four pairs of quantifier prefixes formed by \forall and \exists .

The predominant form is the following relationship between classes c and d

$$\forall x(c(x) \rightarrow \exists y(r(x, y) \wedge d(y)))$$

comprising class inclusion and partonomic ontologies as special cases. These sentences are shaped into a *prima facie* atomic, variable free form, which hides the underlying quantifiers. The entire setup is termed CRL-META.

The embracing DATALOG metalogic enables controlled reasoning of CRL (including, notably, inheritance) and querying of the stated relationships. Thus in CRL-META one may ask queries about for instance which logical relationships there exist between given classes. The DATALOG level may readily be implemented on top of a relational database platform, thereby taking advantage of existing relational query languages with their capabilities for efficient access to large knowledge bases with the CRL level being stored as data.

CRL-META is intended in particular for applications where ontologies are to be combined with – possibly large scale – knowledge bases describing relationships between classes. Bio/pharma-science & -engineering are application areas

calling for tools for computing, say, various causal and effect relationships in combination with partonomies and ontological inheritance reasoning. In our [1] CRL is further coined into a diagram logic which utilizes the dynamic potential of computer screens. A prototype of this CRL diagram system is available.

Although CRL has affinity to description logics (DL), CRL-META is not an attempt to design a crossover of a DL sub-language and DATALOG, cf. e.g. [2], since there remain certain fundamental differences to DL, notably the adoption of the closed world assumption, as to be discussed. Nor is CRL an extension of DATALOG as such, unlike the DATALOG extension with existential quantification in heads of clauses introduced in [3].

The paper is organized as follows: Following an introduction to the applied class-relationship logic CRL in Section 2, Section 3 and 4 present metalogic inference rules providing reasoning and querying functionalities. Section 5 discusses some common relationships. Section 6 compares CRL and CRL-META with description logic. Section 7 discusses handling of class definitions and various epistemic modes pertaining to knowledge bases and ontologies. The final Section 8 concludes and summarizes.

2 Class-Relationship Logic CRL

First order logic states relations between individuals, and accordingly the quantified variables range over individuals. In formal ontologies and in many knowledge bases, unlike in databases, one is primarily interested in relations between classes rather than individuals. Introducing variables ranging over classes in logic requires either resort to logical type theory or encoding of classes as the sort of individuals in a metalogic set up, cf. [4,5]. We pursue here the latter approach choosing at the object logic level class relationship logic, CRL. This section describes CRL, while the metalogic providing embedding and encoding is deferred to Section 3.

A CRL knowledge base and/or ontology takes form of a finite number of sentences in first order logic. The salient sentence form is the $\forall\exists$ -relationship sentence of the elementary form

$$\forall x(c(x) \rightarrow \exists y(r(x, y) \wedge d(y)))$$

stating a relationship between relata classes c and d with relator r . This form may serve to assign the property to members of the class c of being r -related to members in d , cf. [6,7]. This form and related ones are also applied in the biomedical ontologies in [9,10,11,12,13]. In [14] this form is examined for parthood (i.e. part-whole) relationships in particular.

A distinguished case of $\forall\exists$ -relationships is the class inclusion¹, c *isa* d ,

$$\forall x(c(x) \rightarrow d(x))$$

¹ Since we intend an intensional conception of classes throughout, the present extensional specification of class inclusion is understood to be only necessary but not sufficient, cf. our [8].

It may be conceived to come about logically by setting r to the identity relation. Thus the class inclusion relationship in this view is a special, though prominent, case of the $\forall\exists$ -relationship. As a notational convenience therefore we may use $\forall x(c(x) \rightarrow \exists y(r(x, y) \wedge d(y)))$ with r being "=", when the pertinent relationship is the *isa* relation.

The $\forall\exists$ form is foundational and crucial in ontologies because it serves to assign properties to the classes as illustrated in the below sample knowledge base. It appears that inheritance of class properties (cf. the property of being related via r to class d) is catered for implicitly by the first order logic *per se*.

In addition to the $\forall\exists$ -relationship in CRL we consider here three more relationship forms

$$\begin{array}{ll} \exists\exists & \exists x(c(x) \wedge \exists y(d(y) \wedge r(x, y))) \\ \exists\forall & \exists x(c(x) \wedge \forall y(d(y) \rightarrow r(x, y))) \\ \forall\forall & \forall x(c(x) \rightarrow \forall y(d(y) \rightarrow r(x, y))) \end{array}$$

We assume throughout that the classes are non-empty, $\exists xc(x)$ for any c , for reasons clarified below, but we are not concerned with the actual, extensional content of the classes. Individuals in classes may be recognized and dealt with by lifting to singleton classes.

The above relationships between classes c and d can be abstracted as triples $Q'Q''(c, r, d)$ where the 3-ary combinator $Q'Q''$ is either of $\forall\exists$, $\exists\exists$, $\exists\forall$, and $\forall\forall$. As special case of $\forall\exists(c, r, d)$ there is the class inclusion written *isa*(c, d). Thus formally, with the combinator made a predicate, the class relationships may be re-conceived as atomic ground formulae in first order predicate logic where variables may range over (encoded) classes. This abstraction principle forms the basis for the CRL metalogic introduced in Section 3.

2.1 Example Knowledge Base

Consider a sample KB fragment stated first as triples in stylized natural language:

```
betacell isa cell
pancreas haspart betacell
betacell produces insulin
insulin isa hormone
```

The corresponding underlying predicate logical form is:

$$\begin{array}{l} \forall x(\text{betacell}(x) \rightarrow \text{cell}(x)) \\ \forall x(\text{pancreas}(x) \rightarrow \exists y(\text{betacell}(y) \wedge \text{haspart}(x, y))) \\ \forall x(\text{betacell}(x) \rightarrow \exists y(\text{insulin}(y) \wedge \text{produces}(x, y))) \\ \forall x(\text{insulin}(x) \rightarrow \text{hormone}(x)) \end{array}$$

where the predicates *insulin* and *hormone* correspond to mass nouns, unlike the other predicates, and as such ontologically are conceived of as referring to portions of the pertinent substance.

These CRL sentences, when using the below CRL metalogic setup, are then imaginary in that they are actually represented as DATALOG atomic sentences

$isa(betacell, cell)$
 $\forall\exists(pancreas, haspart, betacell)$
 $\forall\exists(betacell, produces, insulin)$
 $isa(insulin, hormone)$

The elementary deductive closure of a CRL knowledge base KB is defined as the set of elementary sentences of the class-class relationship forms, notably, $\forall x(c(x) \rightarrow \exists y(r(x, y) \wedge d(y)))$ (with *isa* giving a special case) which are entailed in the usual first order logical sense from the sentences in the KB.

For instance, in the example we have

$$KB \models \forall x(betacell(x) \rightarrow \exists y(produces(x, y) \wedge hormone(y)))$$

corresponding to a would-be inferred atomic fact $\forall\exists(betacell, produces, hormone)$ at the metalogic level. Section 4 introduces metalogic inference rules reflecting the desired deductions at the CRL logical level.

This closure of elementary sentences is finite let alone due to the finite number of classes and binary relations in a KB.

2.2 Knowledge Base Dually as Graph and Natural Logic

The abstraction of class relationships as triples adorned with quantifier prefixes invites an alternative conception of a CRL knowledge base as a directed, labelled graph, where nodes are uniquely labelled with classes, and arcs are labelled (not necessarily uniquely) with quantifier prefix and binary relations representing class-class relationships. There may be multiple arcs between any pair of nodes.

The *c* class (start) node of such a triple (c, r, d) is called the subject node, and the *d* class (end) node is called the object node of the relationship. The ingoing arcs of a node form the inlets and the outgoing ones form the outlets. This knowledge base graph represents a formalization of the notion of semantic network endowed with a precise logical meaning through the above logical explication of triples.

Thus a CRL knowledge base is dually conceived of as

- (1) a finite, directed, labelled graph with arcs representing class-class relationships, and
- (2) a finite collection of predicate logical sentences of the above stated atomic form.

The graph view is much favored by ontologists, whereas the complimentary predicate logical view determines the permissible logical inferences to be made from the knowledge base.

The considered logical relationship forms as it appears from the above example closely reflect to simple natural language forms. Moreover, the below inference rules support reasoning close to linguistic forms, thereby enabling a natural logic in the sense of [15].

2.3 Ontology Part Proper

The part of a KB consisting of the *isa* triples forms the ontology proper of a CRL knowledge base. In an alternative parlance, if the entire knowledge based is conceived of as an ontology, the *isa* relationships constitutes the skeleton ontology.

Following a common convention arcs showed un-labelled and pointing upwards are *isa* relationships. The concomitant logical explication forces certain restrictions on the shape of this subgraph: The *isa* arrows are to form an acyclic graph except for the (implicit) presence of loops of length one due to $\models \forall x(c(x) \rightarrow c(x))$ for any c . The definition of *isa* relationships ensures further transitivity, making *isa* a preorder becoming a partial order relation with the mentioned acyclicity restriction.

3 CRL Metalogic Level in DATALOG

As stated above the CRL level is formalized in a embracing CRL metalogic level, along the lines in [5][8], cf. also the combinatory logic programming principles in [4]. The employed metalogic at the outset is the DATALOG subset of first order predicate logic. DATALOG consists of definite clauses of the form

$$p_0(t_{01}, \dots, t_{0n_0}) \leftarrow p_1(t_{11}, \dots, t_{1n_1}) \wedge \dots \wedge p_m(t_{m1}, \dots, t_{mn_m})$$

where the predicate argument terms t_{ij} are either constants or variables, where variables are implicitly universally quantified.

The $\forall \exists$ relationships cannot be represented in DATALOG clauses simply by the well-known rewriting to clause form, since removal of the existential quantifier calls for Skolem functions introducing compound terms.

However, in the use of DATALOG as metalogic CRL relationships in the KB are recorded straightforwardly as ground atomic facts (as special cases of the above definite clauses), viz. $Q'Q''(c, r, d)$ supplemented with the *isa* relationships.

In addition to these given (manifest) sentences there are deducibles in the form of triples belonging to the deductive closure being derivable by means of available axioms as to be presented below. These deducibles may be made actually present in the KB (they being of finite extent) or they may remain as virtual triples[2]. Some triples may be furnished with various modes, e.g. epistemic modes as to be explained.

Logical reasoning is then performed at the meta-level by adding appropriate axioms in the form of DATALOG clauses.

4 Axioms and Deduction at the CRL Metalogic Level

Appropriate axioms, *CRL-axioms*, have to be introduced so as to reflect the logical entailment of relationship sentences, e.g. with the requirement:

$$\text{KB} \models \forall x(c(x) \rightarrow \exists y(r(x, y) \wedge d(y))) \quad \text{iff} \quad \text{KB} \cup \text{CRLaxioms} \vdash \forall \exists(c, r, d)$$

² cf. Hasse diagrams in lattice theory, where arcs derivable by transitivity are omitted.

As mentioned we assume that all classes are non-empty (existential import): $\exists xc(x)$ for all classes c . This assumption serve to streamline the inference rules, and it admits further of stating overlap as well as disjointness of a pair of classes. Besides, non-empty classes tend to be suspect from an ontological point of view. Thus, there is no notion of empty class in CRL.

4.1 Partial Ordering of Inclusion

For the inclusion relation further we stipulate

$$\begin{aligned} isa(X, X) \\ isa(X, Z) \leftarrow isa(X, Y) \wedge isa(Y, Z) \end{aligned}$$

Both way inclusions give rise to synonym classes (if not precluded)

$$ident(X, Y) \leftarrow isa(X, Y) \wedge isa(Y, X)$$

The *isa* relation is then effectively established as a partial order (that is, reflexive, anti-symmetric, and transitive) at the meta level.

4.2 Inheritance Axioms

Appropriate axioms have to be introduced so as to reflect logical entailment of relationship sentences. The crucial mechanism here is what is generally understood as inheritance. Following our [1], for class relationships $\forall\exists(c, r, d)$ there is an inheritance inference rule

$$\frac{\forall\exists(C, R, D) \quad isa(C', C) \quad isa(D, D')}{\forall\exists(C', R, D')}$$

or correspondingly the pair

$$\frac{\forall\exists(C, R, D) \quad isa(C', C)}{\forall\exists(C', R, D)}$$

$$\frac{\forall\exists(C, R, D) \quad isa(D, D')}{\forall\exists(C, R, D')}$$

which at the metalogic level may be paraphrased in DATALOG as the definite clauses

$$\begin{aligned} \forall\exists(C', R, D) \leftarrow \forall\exists(C, R, D) \wedge isa(C', C) \\ \forall\exists(C, R, D') \leftarrow \forall\exists(C, R, D) \wedge isa(D, D') \end{aligned}$$

The inference rule and its clausal axiom form, as it appears, expresses inheritance of relationships as properties to sub-classes (first clause), as well as inheritance of more general properties to underlying relationships (second clause). This is reminiscent of the notion of monotonicity in natural logic [15].

For the other class relationship forms following [1] similarly we posit

Relationship inheritance

$$\exists\forall(C, R, D') \leftarrow \exists\forall(C, R, D) \wedge isa(D', D)$$

$$\forall\forall(C', R, D) \leftarrow \forall\forall(C, R, D) \wedge isa(C', C)$$

$$\forall\forall(C, R, D') \leftarrow \forall\forall(C, R, D) \wedge isa(D', D)$$

Relationship generalization:

$$\exists\exists(C', R, D) \leftarrow \exists\exists(C, R, D) \wedge isa(C, C')$$

$$\exists\exists(C, R, D') \leftarrow \forall\exists(C, R, D) \wedge isa(D, D')$$

$$\exists\forall(C', R, D) \leftarrow \exists\forall(C, R, D) \wedge isa(C, C')$$

Weakening of quantifier (recall non-empty classes):

$$\forall\exists(C, R, D) \leftarrow \forall\forall(C, R, D)$$

$$\exists\forall(C, R, D) \leftarrow \forall\forall(C, R, D)$$

$$\exists\exists(C, R, D) \leftarrow \forall\exists(C, R, D)$$

$$\exists\exists(C, R, D) \leftarrow \exists\forall(C, R, D)$$

4.3 Axioms for Inverses

The inverses r^{-1} of the binary relations r may also be appealed to in CRL. The inverse r_{inv} of some relation r is established in CRL-META, say, with the ground atomic auxiliary KB clause

$$inv(r_{inv}, r)$$

with the axiom

$$inv(X, Y) \leftarrow inv(Y, X)$$

Then there is

$$\forall\forall(D, R', C) \leftarrow \forall\forall(C, R, D) \wedge inv(R, R')$$

$$\exists\exists(D, R', C) \leftarrow \exists\exists(C, R, D) \wedge inv(R, R')$$

It is easy to verify that $\forall\exists$ and $\exists\forall$ are connected with

$$\forall\exists(D, R', C) \leftarrow \exists\forall(C, R, D) \wedge inv(R, R')$$

but not *vice versa*, conforming with

$$\exists x(c(x) \wedge \forall y(d(y) \rightarrow r(x, y))) \models \forall y(d(y) \rightarrow \exists x(c(x) \wedge r(x, y)))$$

5 Miscellaneous Relationship Patterns

Having described the logical aspects of of CRL-META, we now turn to the pragmatics of the language.

5.1 Classification Hierarchies

Ontologies, being fundamentally clasifications, are often tree-shaped or close to hierarchical. Consider a hierarchical classification (bi-partitioning) of a class a into classes b and c . In CRL this is specified by the two sentences

$$isa(b, a) \text{ and } isa(c, a)$$

The classes b and c are then disjoint in so far that there is no overlapping class below b and c . If overlap between classes b and c is intended this is achieved in CRL by introducing a class, say, bc accompanied by $isa(bc, b)$ and $isa(bc, c)$, recalling that classes are understood to be non-empty. The question of overlap of a pair of classes is settled in CRL-META with the clause

$$overlap(C, D) \leftarrow isa(X, C) \wedge isa(X, D)$$

checking for existence of a common subclass X .

Complementarily, disjointness is answered by

$$disjoint(C, D) \leftarrow \neg overlap(C, D)$$

appealing to $DATALOG^{\neg}$, that is $DATALOG$ extended with negation as failure.

The appeal to Closed World Assumption (CWA) conforms with presence only of positive knowledge (as in relational databases) in a CRL KB.

In description logic (DL), see e.g. [2], the sample classification is expressed by $b \sqsubseteq a$ and $c \sqsubseteq a$ and $b \sqcap c \sqsubseteq \perp$, where the latter sentence specifies disjointness (non-overlap of classes). One may observe that the move from bi-partitioning to n-partitioning of classes in ontological classifications would severely complicate the expressing of disjointness. The difference between CRL and DL here is bound up with the principle of Closed World Assumption vs. Open World Assumption.

5.2 Inverse and Reciprocal Relations

For any relation r the inverse individual relation r_{inv} is bound to exist mathematically. By contrast, given the class relationship $\forall\exists(c, r, d)$, the class relationship $\forall\exists(d, r_{inv}, c)$ may or may not be in effect in a knowledge base. In the positive case the two relationships are called reciprocals in [11], and we say that they form a tight relationship in case of co-presence.

Commonly occurring tight relationships are the parthood class relationships discussed in [14, 11]. Given a binary mereological relation, $part$, expressing parthood at the instance level, cf. e.g. [17], the relationship

$$\forall\exists(C, part, D)$$

expresses that everything in C has a part instance in D . Dually

$$\forall\exists(D, part_{inv}, C)$$

expresses that everything in D is part of something in C . These reciprocals are independent; co-presence is possible but not mandatory.

5.3 Composite Relations

Queries to a CRL knowledge base may be stated in CRL-META simply as goal clauses. In knowledge bases it may be of interest to identify relationships between some given classes, say c and d . This may be done in CRL-META simply by a goal clause $\leftarrow \forall\exists(c, R, d)$. In the course of computing answers there may be appealed to the various clausal inference rules, e.g. the inheritance rules.

Relationships with composition of relations may be deductively queried with additional $(n + 2)$ -ary predicates in CRL-META as e.g. in

$$\forall\exists(C, R_1, R_2, \dots, R_n, D) \leftarrow \forall\exists(C, R_1, C_1) \wedge \dots \wedge \forall\exists(C_{n-1}, R_n, D)$$

that is, with an additional $(n + 2)$ -ary predicate. The mathematical properties of composed class relationships are examined in [16].

6 Comparison with Description Logic

The class inclusion, $isa(c, d)$, has as counterpart the description logic sentence $c \sqsubseteq d$. More interestingly one may observe that $\forall\exists$ -relationships are closely related to formulations using the two-argument \exists -operator (peirce-product) in description logic in that $\forall\exists(c_1, r, c_2)$ with the predicate logical specification $\forall x(c_1(x) \rightarrow \exists y(r(x, y) \wedge c_2(y)))$ in description logic (see e.g. [2]), becomes

$$c_1 \sqsubseteq \exists r.c_2$$

The independent reciprocal $c_2 \sqsubseteq \exists r_1.c_1$, where r_1 is the inverse of r , i.e. r^{-1} , is absent or present at the discretion of the domain analyst. The $\exists\exists$ -relationships do not seem to have any simple counterparts in description logic.

As a principal difference to description logic the class relationship logic appeals to the Closed World Assumption (CWA), according to which a relationship fails to hold if it is not either given or deducible in the considered domain specification. This implies that two classes are conceived to be disjoint unless there is given a third proper class which is a common subclass of the two classes in question. This principle is applied in relational databases and seems also to conform with ontology practice. It is, however, in contrast to description logic, where disjointness is to be specified explicitly, say, with $c \sqcap d \equiv \perp$.

More generally our abandoning of classic logic in favour of CRL-META implies that relationship logic specifications are bound to be logically consistent as logic programs, unlike description logic specifications. This may be viewed as a deficiency; however, consistency constraints may be formulated at the introduced metalogic level in a well-known manner, say by imposing constraint clauses defining a distinguished predicate, $error(X)$, whose extension, if non-empty when invoked, is construed as inconsistency.

The \forall -operator in description logic in

$$c \sqsubseteq \forall r.d$$

expresses $\forall x(c(x) \rightarrow \forall y(r(x, y) \rightarrow d(y)))$ not to be confused with the $\forall\forall$ -relationship in CRL as defined previously.

7 Coping with Ontological Definitions

Consider the following situation with 3 class relationships, say $\forall\exists(a, r, b)$, $\forall\exists(c, r, d)$, and $isa(d, b)$, viz.

$$\begin{array}{c} a - r \rightarrow b \\ \qquad \qquad \qquad \uparrow \\ c - r \rightarrow d \end{array}$$

One might here be tempted to deduce a second inclusion relationship added below and supposedly derived as a kind of inheritance relationship

$$\begin{array}{ccc} a - r \rightarrow b & & \\ (\uparrow) & & \uparrow \\ c - r \rightarrow d & & \end{array}$$

Appealing to common sense, for instance dog owners may be thought of as pet owners given that dogs are pets. However, this example is seductive but misleading. This fallacy may more specifically be ascribed to misleading use of the monotonicity of the Peirce product, which may be stated (cf. e.g [18], see also [6]) as:

$$(r : d) \text{ isa } (r : b) \quad \text{if} \quad d \text{ isa } b$$

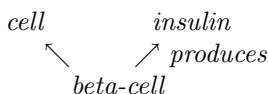
In Description Logic there is the corresponding

$$\exists r.d \sqsubseteq \exists r.b \quad \text{if} \quad d \sqsubseteq b$$

This problem is reminiscent of the historical 19th century discussion of the so-called De Morgan Argument, an important issue in pre-Fregean logic of relations, see the comprehensive and instructive account in [19]. The De Morgan Argument is often rendered as "*Si est caput hominis, et animalis*", that is "Whatever is head of a man is a head on an animal".

What is at stake here is the distinction between "only-if" definitions and full "if-and-only-if" definitions.

Consider



which expresses that the KB contains

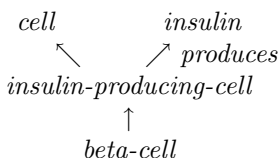
$$\begin{array}{l} \text{isa}(\text{betacell}, \text{cell}) \\ \forall\exists(\text{betacell}, \text{produces}, \text{insulin}) \end{array}$$

If we wish to achieve the complete definition of beta-cell this could be amended with

$$\text{isa}(X, \text{betacell}) \leftarrow \text{isa}(X, \text{cell}) \wedge \forall\exists(X, \text{produces}, \text{insulin})$$

One may introduce the convention that such "if" part of a class definition is implied by default for class nodes having more than one outlet triplet.

In cases where one does not wish to achieve this definitional effect one may use instead



relying on a convention that class nodes with a single outlet are not treated as if-and-only-if definitions. So this latter formulation is open for other cell types as potential candidates for insulin production.

7.1 Mode Options

In order to cater for the situation that one or more outlet triplets $\forall\exists(c, r, d)$ for a class c in the KB are not to participate in the completed definition of c , we suggest that such relationships are tagged as "observable". This annotation is to tell that the relationship does not contribute to the "only-if" definition of c . Apart from this it is to function as any other KB relationship.

The distinction between definition-contributing relationships and the suggested (presumably empirically based) observables is reminiscent of the analytic/synthetic distinction and the *a priori/a posteriori* distinction among propositions.

Straightforwardly an additional inferential mode tag may distinguish between given and deduced relationships in the knowledge base. This further invites gracefully degraded compound relationships (e.g. causal ones) according to the length of the (shortest) deduction.

8 Summary

We have described a logic for ontology structured knowledge representation. Prominently the logic comes with a two-level (i.e. meta) architecture facilitating controlled reasoning and deductive querying about classes and their relationships. Informally CRL-META bears similarity to DL languages, however, it appeals to the Closed World Assumption, contrast the Open World Assumption of DL. As argued, CWA for class relationships might be relevant for comprehensive ontologies, be they hierarchical or trans-hierarchical. The limited expressivity of CRL-META *vis-à-vis* DL (in particular with respect to classical negation) may be compensated by the potential for implementation of the decidable and tractable DATALOG (cf. [2]) meta-level on a relational database platform.

The atomic form of relationships considered here may subject to enrichment with compound classes and relations, e.g. classes formed as conjunctions of classes. In the perspective of natural logic this would open for enrichment of the applied simple linguistic forms with relative clauses, and adnominal and adverbial prepositional phrases.

Acknowledgments. I would like to thank Hans Bruun, Jacobo Rouces, and Jørgen Villadsen for helpful comments.

References

1. Fischer Nilsson, J.: Diagrammatic Reasoning with Classes and Relationships, 18 pages (2010) (submitted)
2. Groszof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic programs with Description Logic. In: 12th WWW. ACM, New York (2003)

3. Cali, A., Gottlob, G., Lukasiewicz, T.: A General Datalog-based Framework for Tractable Query Answering over Ontologies. In: De Virgilio, R., Giunchiglia, F., Tanca, L. (eds.) *Semantic Web Information Management: A Model-Based Perspective*. Springer, Heidelberg (2010)
4. Hamfelt, A., Fischer Nilsson, J.: Towards a Logic Programming Methodology Based on Higher-Order Predicates. *New Generation Computing* 15(4), 421–448 (1997)
5. Fischer Nilsson, J., Palomäki, J.: Towards Computing with Intensions and Extensions of Concepts. In: Charrel, P.-J., et al. (eds.) *Information Modelling and Knowledge Bases IX*. IOS Press, Amsterdam (1998)
6. Fischer Nilsson, J.: A Conceptual Space Logic, in Kawaguchi, E. In: Kawaguchi, E., et al. (eds.) *9th European-Japanese Conferences on Information Modelling and Knowledge Bases, Information Modelling and Knowledge Bases XI*, Iwate, Japan, May 24–28. IOS Press, Amsterdam (1999/2000)
7. Fischer Nilsson, J.: On Reducing Relationships to Property Ascriptions. In: Kiyoki, Y., Tokuda, T., Jaakkola, H., Chen, X., Yoshida, N. (eds.) *Information Modelling and Knowledge Bases XX. Frontiers in Artificial Intelligence and Applications*, vol. 190 (January 2009) hardcover ISBN: 978-1-58603-957-8
8. Fischer Nilsson, J.: Ontological constitutions for classes and properties. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 37–53. Springer, Heidelberg (2006)
9. Andreasen, T., et al.: A Semantics-based Approach to Retrieving Biomedical Information. In: Christiansen, H., et al. (eds.) *FQAS 2011. LNCS (LNAI)*, pp. 108–118. Springer, Heidelberg (2011)
10. Blondé, W., et al.: Metarel: an Ontology to support the inferencing of Semantic Web relations within biomedical Ontologies. In: *International Conference on Biomedical Ontology*, Buffalo, NY (2009)
11. Smith, B., et al.: Relations in biomedical ontologies. *Genome Biology* 6, R46 (2005)
12. Zambach, S.: A formal framework on the semantics of regulatory relations and their presence as verbs in biomedical texts. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) *FQAS 2009. LNCS*, vol. 5822, pp. 443–452. Springer, Heidelberg (2009)
13. Zambach, S., Hansen, J.U.: Logical Knowledge Representation of Regulatory Relations in Biomedical Pathways. In: Khuri, S., Lhotská, L., Pisanti, N. (eds.) *ITBAM 2010. LNCS*, vol. 6266, pp. 186–200. Springer, Heidelberg (2010)
14. Smith, B., Rosse, C.: The Role of Foundational Relations in the Alignment of Biomedical Ontologies. In: *MEDINFO 2004*. IOS Press, Amsterdam (2004)
15. van Benthem, J.: *Essays in Logical Semantics*. Reidel, Dordrecht (1986)
16. Ajspur, M., Zambach, S.: Reduction of composites of relations between classes within formal ontologies. In: *ARCOE Working Notes*, Barcelona (2011)
17. Simons, P.: *Parts, A Study in Ontology*. Clarendon Press, Oxford (1987)
18. Brink, C., et al.: Peirce Algebras. *Formal Aspects of Computing* 6(3) (1994)
19. Sanchez Valencia, V.: The Algebra of Logic. In: Gabbay, D.M., Woods, J. (eds.) *Handbook of the History of Logic. The Rise of Modern Logic: From Leibniz to Frege*, vol. 3. Elsevier, Amsterdam (2004)

A Semantics-Based Approach to Retrieving Biomedical Information

Troels Andreassen¹, Henrik Bulskov¹, Sine Zambach¹, Tine Lassen²,
Bodil Nistrup Madsen², Per Anker Jensen²,
Hanne Erdman Thomsen², and Jørgen Fischer Nilsson³

¹ CBIT, Roskilde University, Universitetsvej 1, Roskilde, Denmark
{troels,bulskov,sz}@ruc.dk

² ISV, Copenhagen Business School, Dalgas Have 15, Frederiksberg, Denmark
{paj.isv,bmn.isv,het.isv,tla.isv}@cbs.dk

³ IMM, Technical University of Denmark, Richard Petersens Plads, Kongens Lyngby, Denmark
jfn@imm.dtu.dk

Abstract. This paper describes an approach to representing, organising, and accessing conceptual content of biomedical texts using a formal ontology. The ontology is based on UMLS resources supplemented with domain ontologies developed in the project. The approach introduces the notion of ‘generative ontologies’, i.e., ontologies providing increasingly specialised concepts reflecting the phrase structure of natural language. Furthermore, we propose a novel so called ontological semantics which maps noun phrases from texts and queries into nodes in the generative ontology. This enables an advanced form of data mining of texts identifying paraphrases and concept relations and measuring distances between key concepts in texts. Thus, the project is distinct in its attempt to provide a formal underpinning of conceptual similarity or relatedness of meaning.

Keywords: Domain modelling, Ontology engineering, Natural language processing, Ontological, Content-oriented text search.

1 Introduction¹

Search in texts is progressing beyond conventional keyword search in order to make it less syntactic and more semantically oriented. This paper presents endeavours in the SIABO project aiming at achieving content-based text search within the application area of biomedicine.

Our main thesis is that a content-based search functionality can be achieved by computerised text analysis using ontologies enhanced with domain models and language processing.

The remainder of this section describes the aims of the SIABO project in general, Section 2 introduces the notion of generative ontology, Section 3 presents the kind of domain modelling carried out in the project, Section 4 sets out two approaches to concept extraction which we are currently pursuing, one synthetic, and the other

¹ This paper is an extended and substantially revised version of [12].

pattern-based. Section 5 addresses the problems related to querying information and knowledge, and finally, in Section 6, we present our conclusions.

1.1 The SIABO Project

The aim of the SIABO project is to provide an approach to representing, organising, and accessing the conceptual content of biomedical texts using a formal ontology.

In order to be competitive, companies need to have access to the contents of the increasing amount of documentation about their products, processes and projects. Retrieval of information and knowledge from huge, diverse resources is vital, and only a semantics-based approach to information management is adequate to that task.

This project presents an approach in which the meaning content of each document is described as a set of arbitrarily complex conceptual feature structures facilitating detailed comparison of the content of documents. The properties of an ontology-based system lead to easier access to data sources, locally as well as globally.

Ontologies are formal tools for structuring the concepts of a scientific domain by means of relationships between concepts, e.g., along the specialization/generalization dimension. The SIABO approach introduces the notion of generative ontologies, i.e., infinite ontologies providing increasingly specialised concepts. The project sets up a novel, so called ontological semantics, which maps the conceptual content of phrases into points in the generative ontology. Text chunks with identical meaning but different linguistic forms are to be mapped to the same node in the generative ontology. Thus, the approach facilitates identification of paraphrases, conceptual relationships and assessment of distances between key concepts in texts. The project focuses on ontological engineering of biomedical ontologies founded on lattices and relation-algebras, and has clear affinities to contemporary research in the Semantic Web area, description logic as well as XML approaches. However, it gains its distinct innovative scientific profile by means of the above-mentioned notions.

2 Generative Ontology

A generative ontology is an ontology where the usual class inclusion relationships are extended with property ascriptions. In other words a generative ontology is based on a finite ontology with the *isa* concept inclusion relation (called the ‘skeleton ontology’), enriched with a set of semantic relations providing generativity. For instance, the skeleton ontology may specify the inclusion paths:

secretion *isa* process *isa* event
insulin *isa* hormone *isa* stuff

The extension with property ascription by feature structures provides generativity by virtue of their recursive structure. Accordingly, with generative ontologies we move from finite ontologies to infinite systems of concepts, thereby reflecting the recursive productivity of the phrase structures in natural language. This makes it possible to map complex linguistic structures into correspondingly complex concepts associated with nodes in the generative ontology.

Semantic relations enable construction of feature structures such as *disease* [*CausedBy*: *lack*[*WithRespectTo*: *insulin*]]. This feature structure corresponds to

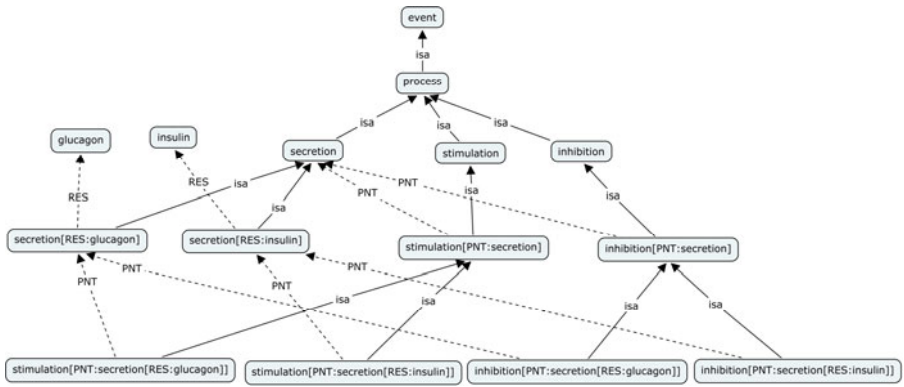


Fig. 1. Fragment of a generative ontology for insulin production

linguistic paraphrases found in a text or a query like *diseases caused by insulin lack*, *diseases induced by insulin deficiency*, *insulin deficiency disease*, etc.

2.1 Class Relationship Logic

The logic underlying generative ontologies is a class-relationship logic (CRL), see [12]. CRL basically introduces sentences of the logical form

$$\forall x (c(x) \rightarrow \exists y (r(x,y) \wedge d(y)))$$

where c and d are ontological concepts (*i.e.* classes in CRL) and r a binary relationship.

An important special case of the above sentence form comes about by letting r be equality, in which case we get

$$\forall x (c(x) \rightarrow d(x))$$

that is, (extensional) class inclusion, c *isa* d . In general we may prefer to conceive of these various sentences simply as triples (c, r, d) , with r being *isa* as an option. The triple form then provides the CFS $c[r:d]$ as $(c-r-d, r, d)$ together with $(c-r-d, isa, c)$, where $c-r-d$ is a new concept. These triples simply correspond to arcs in Fig. 1.

However, the logic devises additional implicit triples: The logical explication of the triples justify inter alia the key inference rules of property inheritance

$$(C_{sub}, R, D) \text{ if } (C, R, D) \ \& \ (C_{sub}, \textit{isa}, C)$$

and property generalisation

$$(C, R, D_{super}) \text{ if } (C, R, D) \ \& \ (D, \textit{isa}, D_{super})$$

2.2 Conceptual Feature Structures

Conceptual feature structures (CFSs) are recursive structures of the following form:

$$c[r_1:c_1, r_2:c_2, \dots, r_n:c_n]$$

where c is a concept from the skeleton ontology, r_1, r_2, \dots are semantic relations, and c_1, c_2, \dots, c_n , for $n \geq 0$ are CFSs. Note that an atomic concept is also a CFS.

The feature-value pairs (attributions) $[r_1:c_1, r_2:c_2, \dots]$ consist of relations and concept arguments, and function as conceptual restrictions on the head concept c . This means that e.g. $c[r_1:c_1]$ is situated below the node c in the ontology. In this way, paths lead to increasingly specialised concepts in the ontology. However, the generative ontology does not admit arbitrary combinations of relations and concepts: The relations express ontologically admissible ways of combining concepts according to so called ontological affinities. The affinities are specified so as to rule out category mistakes. Thus the affinities license ontologically admissible combinations analogously to types in programming languages. In our context of ontologies for scientific texts within biomedicine, we concentrate on physical-chemical-biological categories and disregard metaphors. Currently, ontological affinities are expressed as affinity triples $\langle c', r, c'' \rangle$, which specifies that the concept c' and its subconcepts can be related via r to c'' and its subconcepts. Thus, a general affinity is achieved for r by letting c' and c'' be the universal top concept.

3 Domain Ontologies

Domain ontologies in this context are terminological ontologies, which express a fragment of the generative ontology validated by terminologists and domain experts. We construct domain ontologies supplementing and refining already existing ontologies for the domain, e.g. the UMLS (*Unified Medical Language System*) repository, which maps across several ontologies. The construction of validated domain ontologies is necessary because existing ontologies, for our domain of interest, are not sufficiently specific regarding concepts and concept relations. We work from the assumption that, in an information retrieval setting, the more precise the domain ontology is, the more precise the answers will be, and thus, we need precise and validated domain ontologies. Thus, to facilitate more precise query answering, we have constructed domain ontologies that supplement already existing ontologies such as UMLS.

3.1 Terminological Ontologies

The structure of terminological ontologies is based on characteristics and subdivision criteria. We use an extended set of concept relations as proposed in [5]. The terminological modeling approach can be described as an iterative procedure (cf. [14]), as outlined below:

- a. Find co-ordinate concepts related to one superordinate concept.
- b. Identify the characteristics of the co-ordinate concepts.
- c. Can the co-ordinate concepts be separated by one characteristic? If yes, introduce an attribute-value pair on each concept.
- d. Group the co-ordinate concepts by one or more subdivision criteria.
- e. If step c is not possible, i.e. there is a need for more delimiting characteristics on each concept, introduce an extra layer of concepts so that the co-ordinate

concepts form part of a polyhierarchy, i.e. inherit characteristics from two (or more) superordinate concepts belonging to two (or more) different subdivision criteria.

- f. Repeat the procedure for a more fine-grained concept system.

This approach provides coherent and consistent domain models. However, terminological ontologies are not strictly speaking formal ontologies but they may be transformed into such: The generative ontology excerpt in Fig. 1 is a representation in terms of CFSs of the terminological ontology excerpt shown in Fig. 2 below.

In terminological ontologies, concepts are represented as terms (linguistic expressions), e.g. *insulin secretion*, with added feature specifications expressing characteristics of the concept, e.g., *RESULT:insulin* for the concept *insulin secretion*. In a generative ontology, concepts are solely represented as CFSs, e.g., *secretion [RES:insulin]*, which can be construed as a definition of the concept. In this case the concept is ‘a kind of *secretion* which has the result *insulin*’. Thus, on the basis of ordering relations and characteristics, terminological ontologies may be transformed into CFSs of the generative ontology. For instance, the domain ontology concept *insulin secretion*, which has the superordinate concept *secretion* and the characteristic *RESULT:insulin*, will straightforwardly translate into the CFS *secretion[RES:insulin]*. Thus, the two representation forms are closely interlinked in that any terminological representation can be translated into a generative representation, but not necessarily vice versa, since the generative ontology may contain concepts which have no terminological counterparts.

3.2 An Ontology for Insulin Production

In Fig. 2, an excerpt of one of the resulting ontologies is shown. Boxes with text in capital letters represent subdivision criteria, the other boxes represent concepts. The lines without arrows represent *isa* relations, the arrow lines represent other relations. Characteristics are given in the form of feature specifications below the concept boxes.

Based on an analysis of the characteristics of the concepts *stimulation* and *inhibition* within the insulin domain, these concepts are grouped under the subdivision criterion *INFLUENCE*, where the distinct characteristics show the difference between them. Where possible, concepts have been mapped to UMLS in order to obtain one coherent ontology.

In order to transform this ontology into one that can be used in the project prototype, a number of steps are required. As described in Section 5, the prototype works on an index containing semantic descriptions of texts. The indices contain descriptions of larger text chunks, e.g. noun phrases or sentences, and not just terminological units. Thus, not all items in the index will match a concept in the domain ontology. However, we wish to represent the domain ontology in a form that will in fact match index terms when appropriate. For instance, if a text contains the sentence *insulin is secreted from the mouse’s pancreas*, it receives the annotation *secretion[RST:insulin, LOC:pancreas[POF:mouse]]*, parts of which should also be identified as domain concepts. Thus, it requires a CFS that can be matched against

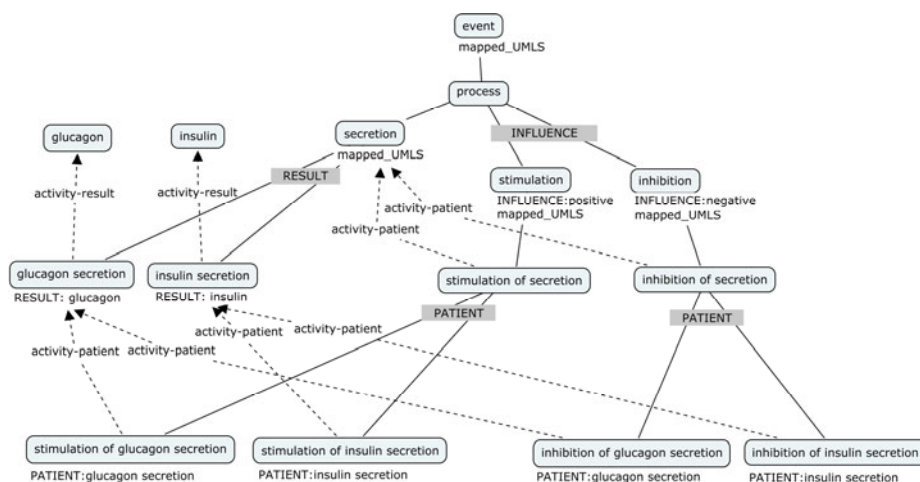


Fig. 2. Fragment of the domain ontology for insulin production, corresponding to the generative ontology in Fig. 1

CFSs in the index. This enables the ontology to be used both for querying and answering in the project prototype.

For the domain user, an ontology browsing search function may also be implemented, and for this purpose, a natural language representation, as included in the terminological ontology in Fig. 2 is probably the most appropriate representation form.

4 Extraction of Concepts from Text

The SIABO project investigates two approaches to concept extraction in parallel: a synthesis-approach relying on the generative ontology, and a pattern-based approach which relies on knowledge extracted from a variety of lexical resources. The two approaches are described below.

4.1 The Synthesis Approach

The purpose of the synthesis approach is to map phrases in the analysed text into CFSs and hence, into nodes in a generative ontology. Ideally, paraphrases are mapped into the same node so that they can be recognised as such. If a target CFS is not already present as a node in the current, actualised part of the generative ontology when analysing a text phrase, it is generated. This may necessitate generating further intermediate nodes connecting to already existing nodes.

The computerised text analysis employed in this approach is conducted chiefly by the generative ontology, however assisted by conventional grammars. Ideally, a sentence would be turned into one CFS in the generative ontology, which is supposed to represent the ontological meaning content of the sentence. This is in contrast to

other approaches to the characterization of propositional content which take into account determiners, negations, and logical conjunctions.

The ontology-driven rather than syntax-driven text processing is performed by a so called OntoGrabber [11, 13]. In principle, the OntoGrabber generates, in a top-down manner, candidate CFSs to be matched against the target sentence in the text. However, since, in general, parts of sentences have to be skipped as unrecognisable, the current OntoGrabber prototype conducts a bottom-up analysis for synthesising CFSs according to the generative ontology. In addition, the OntoGrabber is guided by conventional linguistic grammar rules. As a key point, many potential syntactical analyses due to structural ambiguities are never actualised since they are dismissed as category mistakes by the ontological affinities specified on the set of ontological relations. For instance, multiple adnominal prepositional phrases bring about syntactical ambiguities in the noun phrase since they can be either aligned or nested. Also, the logical affinities serve lexical disambiguation.

In this synthesis approach, adjectives and prepositional phrases give rise to CFS contributions to be attached to the concept coming from the head noun in noun phrases. Thus it is assumed that adjectives and prepositional phrases express constraints on the head noun. Verbs are dealt with by nominalisation. Crucially, this approach admits partial, incomplete analysis, which in the worst case falls back on keywords appearing in the analysed sentence as well as in the generative ontology.

4.2 The Pattern-Based Approach

In parallel with the synthesis approach, we explore a pattern-based approach to concept extraction. The patterns are generated from information available from existing lexical resources, currently the nominalisation lexicon NOMLEX-plus, the verb-lexicon VerbNet and WordNet. These resources provide syntactic argument realization rules for verbs and their arguments and nominalised forms of verbal expressions with semantic information in the form of semantic roles. In principle, any lexical resource may be modified and plugged into the framework. The pattern-based approach performs extraction in two phases: a syntactic annotation phase and a semantic generation phase. Meaningful chunks of text are transformed into CFSs and mapped into concepts in the generative ontology. By this approach, synonymous but linguistically quite distinct expressions are extracted and mapped to the same concept in the ontology, providing a semantic indexing which enables content-based search.

The syntactic annotation phase, phase 1, performs a shallow parsing of the text using lexical and syntactic tools to identify linguistic structures and annotate them. In other words, coherent text chunks such as phrases and compound words are identified, their boundaries are marked up and they are assigned a syntactic annotation. Thus, an example like *increased insulin concentration inhibits hepatic production* will be annotated as follows using a set of rewriting rules employing the Penn Treebank tagset:

$$\langle increased \rangle_{AP} \langle insulin \rangle_{NN} \langle concentration \rangle_{NN} \rangle_{NP} \rangle_{NP} \langle inhibits \rangle_{VBZ} \langle hepatic \rangle_{AP} \langle production \rangle_{NN} \rangle_{NP}$$

The processing of the rules leads to nested annotations and consequently the input forms a sequence of nested sequences $S = s_1 s_2 s_3 \dots$ with singletons at the innermost level. Initially, singletons correspond to words in the input, but during rewriting,

sequences are reduced to singletons as they are replaced by the concepts they denote. A word w annotated with a syntactic label a appears in the input as w/a . A sequence of k elements $s_1 \dots s_k$ annotated with a , where s_i is either a word or a sequence, appears as $\langle s_1 \dots s_k \rangle / a$. Phase 1 can introduce ambiguity since processing of text can lead to several different annotations. Furthermore, ambiguity can stem from the order of application of tools and grammars. There is not necessarily one single path through phase 1, i.e. one unique sequence of application of the lexical and syntactic tools.

Phase 2 transforms the annotated input from phase 1 in order to provide semantic descriptions in the form of CFSs. This is done by transformation schemes, i.e. rules that combine a pattern P , a possibly empty set of constraints C and a template T :

$$P C \rightarrow T$$

The pattern P matches the surface form of the annotated input, combining word constants, word variables and tags. The constraint C expresses restrictions on the constituents of the pattern P that have to be met for the rule to apply. The template T is an expression with unbound variables that instantiates to a CFS by unification with the pattern. A sample rule covering for instance *hepatic cancer* is the following:

$$hepatic/_{JJ} X/_{NN} \{isa(omap(X), DISEASE)\} \rightarrow omap(X) [PNT: LIVER]$$

Here *hepatic* is a constant, X an open variable, $omap(X)$ is a function that maps X to a concept in the ontology. The predicate expression $isa(X, Y)$ evaluates to true if X is a specialization of Y . The constraint $\{isa(omap(X), DISEASE)\}$ restricts X to the ontology $DISEASE$, that is, the mapping to the ontology $omap(X)$ must be a specialization of the concept $DISEASE$. The template in this case forms the CFS $omap(X)[PNT: LIVER]$. Thus, *hepatic cancer* when annotated as *hepatic/_{JJ} cancer/_{NN}* will be transformed into the concept $CANCER[PNT: LIVER]$.

The principle in the rewriting is to iteratively pair a subsequence and a matching rule and to substitute the subsequence by the template from the rule. In order to make a subsequence subject to substitution, the subsequence must be unnested, i.e. it must be a sequence of singletons. Thereby, rewriting is performed bottom-up starting at the innermost level. The rewriting of the annotated input into a semantic description proceeds as follows:

Input: A sequence S of nested annotated sequences with annotated words at the innermost level and a set of rewriting rules $R = \{R_1, \dots, R_n\}$

- 1) Select a combination of a rule $R_i: P_i C_i \rightarrow T_i$ and an unnested subsequence $\langle s_1 \dots s_m \rangle / a$ of m annotated singletons, $m \geq 1$, such that $s_1 \dots s_m$ matches P_i and complies to C_i
- 2) Rewrite S replacing subsequence $s_1 \dots s_m$ with T_i , instantiating variables unifying with P_i
- 3) If any remaining subsequences of S match a P_i go to 1)
- 4) Extract from the rewritten S all generated CFSs ignoring attached annotations

Output: A set of CFSs describing the text annotated in S

Notice that this principle is nondeterministic and varies with the order in which matching rewriting rules are selected. Further, the result of repeated rewriting in 1) to 3) will not always replace all annotated sequences in S . Step 4) takes care of partially rewritten sequences: generated CFSs are extracted and everything else ignored.

The following set of simple rewriting rules introduces heuristics for assigning semantic relations between concepts. We stipulate by R_1 that AP pre-modifiers of NP heads are related to the NP head by a characterization relation (CHR) and that pre-modifying nouns (rule R_2) as well as PP post-modifiers of NP heads (rule R_4) are related to the head by a with-respect-to relation (WRT).

- $$R_1: Y/_{AP} Z/_{NP} \rightarrow omap(Z)[CHR:omap(Y)]$$
- $$R_2: X/_{NN} Y/_{NN} \rightarrow omap(Y)[WRT:omap(X)]$$
- $$R_3: Y/_{NP} \rightarrow omap(Y)$$
- $$R_4: X/_{NP} Y/_{IN} Z/_{NP} \rightarrow omap(X)[WRT:omap(Z)]$$

As an example, applying the transformation principle sketched above with these rules on *increased insulin concentration*, two rewritings starting from the annotated input can lead to a single output CFS:

Text: increased insulin concentration inhibits hepatic production

Input: $\langle increased/_{AP} \langle insulin/_{NN} concentration/_{NN} \rangle /_{NP} \rangle /_{NP} inhibits/_{VBZ} \langle hepatic/_{AP} production/_{NN} \rangle /_{NP}$

- a) $\langle increased/_{AP} \langle CONCENTRATION[WRT:INSULIN] /_{NP} /_{NP} inhibits/_{VBZ} \langle hepatic/_{AP} production/_{NN} /_{NP} \rangle \rangle$
- b) $\langle CONCENTRATION[WRT:INSULIN,CHR:INCREASED] /_{NP} inhibits/_{VBZ} \langle hepatic/_{AP} production/_{NN} /_{NP} \rangle \rangle$
- c) $\langle CONCENTRATION[WRT:INSULIN,CHR:INCREASED] /_{NP} inhibits/_{VBZ} \langle PRODUCTION [CHR:HEPATIC] /_{NP} \rangle \rangle$

Output: {CONCENTRATION[WRT:INSULIN,CHR:INCREASED], PRODUCTION [CHR:HEPATIC]} □

Here, rule R_2 applied on the NN-NN subsequence of the input leads to a), rule R_1 applied on the remaining subsequence of a) leads to b), and c) is the result of applying rule R_1 again to *the hepatic production*.

In this derivation the verb *inhibits* is ignored. Using the resource NOMLEX we extract rules covering the nominalization of verbs, which makes it possible to interpret them as atomic concepts in the same way as nouns. An example of such a nominalization rule covering *inhibit* is:

- $$R_5: X/_{NP} Y/_{VG} Z/_{NP} \{head(Y) = inhibit, isa(omap(Z), PROCESS)\} \rightarrow omap(inhibition)[AGT:omap(X), PNT:omap(Z)]$$

Using this rule, the next step in the derivation will yield a single concept for the whole text:

INHIBITION[AGT:CONCENTRATION[WRT:INSULIN,CHR:INCREASED], PNT:PRODUCTION [CHR:HEPATIC]].

By nominalising verbs as shown we can represent the conceptual content of sentences as complex concepts where the reified verbal concept is attributed with the concepts realised by the verb's arguments. This approach has an additional important advantage in that semantically related noun and verb forms receive identical conceptual interpretations. Thus, the linguistic expressions *inhibition of hepatic production by increased insulin concentration* as well as *inhibiting hepatic production by increasing insulin concentration* become conceptually identical to the original text.

5 Querying Information and Knowledge

Given a domain ontology as shown above and a set of documents in which concepts have been identified, the task is to provide means for query interpretation and evaluation that draw on conceptual content and exploit the conceptualisation in the ontology.

In the present approach, query evaluation relies on comparison of a conceptual description of the query with conceptual descriptions of texts from the database. A conceptual description is a set of CFSs providing a mapping from the text or the query to the ontology. Search in a text collection indexed by concepts can employ concept similarity-measures so that conceptual reasoning can be replaced by simple similarity computation, thereby allowing for scaling to very large information bases. Thus, a major challenge is to define conceptual description similarity in terms of the structure and relations in the ontology.

One obvious way to measure similarity in ontologies is to evaluate the distance in the graphical representation between the concepts being compared, where shorter distance implies higher similarity. A number of different ontological similarity measures have been proposed along these lines, for instance, Shortest Path Length [8], Information Content [9], see also [2].

An essential part of document querying is to establish a mapping that, given a description for the query, indicates matching – or similar – descriptions for texts. One option is to let similarity reflect the skeleton ontology by deriving it from the syntactic derivation relation for CFSs, where longer derivation paths correspond to smaller degree of similarity. However, the comparison of conceptual descriptions should not be merely syntactic. Rather, description resemblance can be measured in terms of similarity derived from all concept relations in the ontology. Initially, in the processing of a query, a description is generated. Then this query description is compared, in principle, to every conceptual description of every document appearing in the database. Finally, documents are ranked by the degree to which their respective descriptions resemble the conceptual description of the query. The query answer is a ranking of the documents that are most similar to the query.

In a framework where the domain of texts is reflected in a knowledge base, as comprised by the ontology, obviously not only the texts, but also the domain ontology may in some cases be the target of interest for queries. Knowledge about the existence of concepts, how concepts are related and about similarities between concepts is also relevant. In addition, knowledge about the actual content of texts can be viewed through the ontology simply by revealing only concepts that exist in the texts. In other words, the ontology plays a specific role here, since it constitutes the means by which we can obtain a conceptual view of the text content.

Thus as an additional functionality, the user may browse the generative ontology directly and then follow the links to the relevant text parts by descending to an ontological level of specialisation with a manageable number of links to the target text.

6 Conclusion

We have presented an approach to representing, organising, and accessing conceptual content of biomedical texts using a formal, generative ontology. In particular, we have presented the key ideas addressing exploitation of ontologies for carrying out content-based text search within a scientific domain recognising not only synonyms

but also more general paraphrasations. Currently, we are evaluating working prototypes. However, the viability of the approach remains to be validated on a large scale, in particular whether the devised ontological text processing prototypes afford a significant improvement compared with conventional keyword search.

References

1. Andreassen, T., Fischer Nilsson, J.: Grammatical Specification of Domain Ontologies. *Data & Knowledge Engineering* 48, 221–230 (2004)
2. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Semantic Distance. *Computational Linguistics* 32(1) (2006)
3. Jensen, P.A., Fischer Nilsson, J.: Ontology-based Semantics for Prepositions. In: Saint-Dizer (ed.) *Syntax and Semantics of Prepositions, Text, Speech and Language Technology*, vol. 29, pp. 229–244. Springer, Heidelberg (2006)
4. Ben-Avi, G., Francez, N.: Categorical Grammar with Ontology-refined Types. In: *CG 2004* (2004)
5. Madsen, B.N., Pedersen, B.S., Thomsen, H.E.: Semantic Relations in Content-based Querying Systems: a Research Presentation from the OntoQuery Project. In: *Proceedings of the 1st International Workshop on Ontologies and Lexical Knowledge Bases, OntoLex 2000*, pp. 72–82. OntoText Lab., Sofia (2002)
6. Madsen, B.N., Thomsen, H.E., Vikner, C.: Multidimensionality in terminological concept modelling. In: *Terminology and Content Development, TKE 2005, 7th International Conference on Terminology and Knowledge Engineering*, Copenhagen, pp. 161–173 (2005)
7. Fischer Nilsson, J.: A Logico-Algebraic Framework for Ontologies. *ONTOLOG*. In: Jensen, P.A., Skadhauge, P. (eds.) *Proceedings of the First International OntoQuery Workshop on Ontology-based Interpretation of Noun Phrases*, Kolding (2001)
8. Rada, R., Bicknell, E.: Ranking Documents with a Thesaurus. *Journal of the American Society for Information Science* 40(5) (1989)
9. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* (1999)
10. Andreassen, T., Bulskov, H., Lassen, T., Zambach, S., Jensen, P.A., Madsen, B.N., Thomsen, H.E., Nilsson, J.F., Szymczak, B.A.: SIABO - Semantic Information Access through Biomedical Ontologies. In: Dietz, J.L.G. (ed.) *KEOD 2009 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Funchal, pp. 171–176. INSTICC Press (2009)
11. Szymczak, B.A.: *Computing an Ontological Semantics for a Natural Language Fragment*, PhD-thesis, IMM-PHD-2010-242, DTU Informatics, Technical University of Denmark (2011)
12. Fischer Nilsson, J.: Querying Class-Relationship Logic in a Metalogic Framework. In: Christiansen, H., et al. (eds.) *FQAS 2011. LNCS (LNAI)*, vol. 7022, pp. 96–107. Springer, Heidelberg (2011)
13. Fischer Nilsson, J., Szymczak, B.A., Jensen, P.A.: ONTOGRABBING: Extracting Information from Texts Using Generative Ontologies. In: Andreassen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) *FQAS 2009. LNCS*, vol. 5822, pp. 275–286. Springer, Heidelberg (2009)
14. Zambach, S., Madsen, B.N.: Applying terminological methods and description logic for creating and implementing an ontology on inhibition. In: Dietz, J.L.G. (ed.) *KEOD 2009 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Funchal, pp. 452–455. INSTICC Press (2009)

Knowledge Extraction for Question Titling

Carolina Gallardo Pérez and Jesús Cardeñosa

Validation & Business Applications Research Group
Universidad Politécnica de Madrid–Spain
{carolina,carde}@opera.dia.fi.upm.es

Abstract. This article describes the work performed over the database of questions belonging to the different opinion polls carried during the last 50 years in Spain. Approximately half of the questions are provided with a title while the other half remain untitled. The work and implemented techniques in order to automatically generate the titles for untitled questions are described. This process is performed over very short texts and generated titles are subject to strong stylistic conventions and should be fully grammatical pieces of Spanish.

Keywords: titling, KE, summarization, opinion polls, subjective clustering.

1 Introduction¹

The wide variety of elements that search systems can look for, like press news with headings and titles as active fields for search, questions for the creation of opinion polls, audiovisual materials (where their metadata can be longer than a mere title but shorter than a summary) or the media, represent different contexts where the title is a key concept to guide and perform the search. These contexts have used tools and techniques to generate titles and summaries with different results.

Although titling can be considered as a summarization task, there are some distinctive features between them, like the usually shorter length of titles as opposed to summaries, stronger stylistic conventions imposed over titles, and the different nature of the input texts to be summarized or titled.

There are two main approaches for the production of a summary, headline or title: to compose it from extracts of the input document (extractive summarization) or to compose it from an abstract of the document, identifying its central subject matter. Extractive summarization tries to identify the relevant sentences of a document. Thus summarization is viewed as a classification problem of relevant/non relevant sentences, which mainly relies on statistical knowledge [1]. To distinguish relevant from irrelevant sentences, several criteria can be used: position of sentences (where sentences heading or closing the paragraph are considered to be more relevant) is used in [2]; the presence of *signature words* (that can be defined by means of frequency measures like the tf-idf schema) is determinant in [3]; or sentence length combined

¹ The work reported in this article has been sponsored by CIS (Center for Sociological Research) of the Ministry of the Presidency of Spain.

with positional criteria and the presence of certain words is reported in [4]. Besides, since extractive techniques may produce incoherent or ungrammatical outputs, it can be the case that the generated summary or title is not required to constitute a completely grammatical expression [5, 6]. By far, these techniques are the most frequent approach for generating titles, headlines or summaries.

On the other hand, non-extractive trends in single-document summarization rely on knowledge-based techniques and tend to be domain-dependent approaches. For instance, a paradigmatic system like SUMMONS [7] is restricted to the summarization of news about terrorism. SUMMONS firstly extracts relevant information (like places, victims, authors, date, etc.) from texts using predefined templates. Then the extracted information is passed through a language generator module, which is also template-based. Other knowledge-based approaches make use of linguistic processing, like [8], together with domain knowledge [9], [10]. In any case, during the last decade there has been much less research and work in knowledge-based summarization (see [11] for a comprehensive review of the summarization task).

Due to their generality, it is difficult to find appropriate ad-hoc solutions based on these techniques to very specific problems, like the one presented in this paper: titling of a huge corpus of questions of the opinion polls carried out by the *Center for Sociological Research* (hereafter CIS) of Spain. CIS is a public institution with a long and stable tradition with its origins in the early 60ies. This means more than 50 years collecting sociological data from the Spanish society stored in different formats, means, databases and information supports. This institution decided to homogenize the structure of all their questions and surveys, a decision that involved the task, among others, of question titling, since titles would be used to identify similar questions in order to reuse them when composing new surveys based on previous ones or to establish temporal series of similar questions. Although all updating processes were initially manually performed by the CIS staff, beginning from the newest survey to oldest one, it turned out quite unmanageable and highly unproductive to manually review thousands of questions.

This article describes the methodology, specifically designed for this institution, to automate the process of question titling. The new generated titles should serve for the aforementioned search purposes. To do that, Information Extraction techniques have been applied in order to extract and identify the relevant and distinctive parts of the questions in order to build up the whole title. Although our proposal is based on domain-dependent criteria, it could be applicable to similar problems. The article is structured as follows: section 2 contextualizes this work and the problem to be solved; section 3 describes the preliminary analysis of the domain that ends up with a typology of titles; the resolution strategy is exemplified with a case study, developed in section 4; final remarks are stated in section 5.

2 Context

CIS carries out regular opinion polls to extract sociological data from the Spanish society. A sociological variable can vary from a specific piece of information about the interviewee (like labour situation, education level, social class, number of cars that

the interviewee has, etc.) to the interviewee's opinion about a given issue, institution or public person. These opinion polls follow a fixed structure in order to elicit sociological variables and usually consist of a set of ordered questions.

A question can be viewed as a more or less complex structure with the following parts:

- A title or expression of the concept that underlies the whole question.
- The question text, that is, the exact wording of what is asked to the interviewee (includes an introduction, the question itself and instructions to the interviewer).
- Sociological variables (from one to many) that are covered by the question. They can coincide with the title or the title can be a grouping of the variables.
- The answer categories that establish the range and scope of the permissible answers to the question.

Our specific problem is defined by the necessity to assign a title to the questions that belong to surveys –opinion polls– of CIS. Besides, a title should meet two conditions:

- a) It should contain the topic concept of the question (what the question is about)
- b) It should imply the typology of answers categories of the question (yes/no question, multiple choice, scales and degree of evaluation).

When CIS begins with the updating processes, the situation is defined by the following figures:

- Number of questions in the database: 87221
- Number of manually titled questions: 39257
- Number of untitled questions: 47964 (from which, 1627 questions are dismissed since they resulted from an updating process)
- Erroneous questions: 150

As already said, the process of question titling is manually performed by experts. It is a creative process, like summarizing or translation, which heavily depends on the style and understanding of each person. Besides, for a specific question several alternatives can be posed and be acceptable. Thus, when facing the task of the automatic creation of title, we are proposing an automatic solution for a creative process.

3 Preliminary Analysis

The nature of the problem is determined by two main factors: a) titling is a creative process; and b) there already exists a big corpus of titled questions so that new titles have to be similar to the existent ones. Taking into account these determining factors, we proceeded to study the plausibility of the task.

The corpus of titled questions was thoroughly analyzed with a clear objective in mind: look for regularities in titles and their associated questions. The analysis focused on the linguistic features of titles from both a syntactic and a pragmatic point of view, highlighting aspects like the type of linguistic construction and the subjacent intention of the title. It also attempted to unveil determining aspects like the relation of the title with regard to the question.

Under such a formalist perspective, any thematic analysis fell out of the scope of this work (although CIS is specialized in pre- and post-electoral surveys and political issues and so it was expected to find many regularities and frequencies in the questions about these topics).

3.1 Setting the Work

This section depicts the analysis of existent titles and its associated questions. The main aspects that are analyzed are the type of linguistic construction (noun phrases, quotations, existence of paraphrases) and the degree of subjectivity/objectivity of the implicit linguistic enunciation in titles.

Most titles are noun phrases headed by a noun followed by a prepositional phrase of diverse complexity. There is almost no variety in this syntactic configuration. What is really striking is the variety in the subjective or objective nature of the linguistic expressions: evaluation, classification, dichotomies establishment, election from within a list of options, or simply assertion of an objective piece of information. The study of the intrinsic features of titles delivers two broad categories of titles: *subjective titles* and *objective titles*.

3.1.1 Subjective Titles

Titles under this category express an interviewee's judgement of any sort about a given topic. The judgement can be an approval, rejection, preference, evaluation, etc. of a topic or person. The type of judgement is explicitly expressed in the title, together with the object of the judgement.

In general, the structure of subjective titles follows the general schema of:

Type of judgement + Nexus + Topic

Where *type of judgement* is a word like "opinion", "preference", etc., *nexus* is the preposition or conjunction required by the head noun, and *topic* is the nominal group, clause or even quotation denoting what the question is about. Let's look at two particular examples:

TITLE: Opinion on the degree of interest of the central government in issues of the Valencian Community.

QUESTION: Q.24 Do you believe that the Central Government ...?

- Tries hard to solve problems in the Valencian Community.
- Is interested in the economic progress of Valencia.
- Is fair in the sharing of the economical assets in Valencia.
- A lot -- Sufficient – A little – Nothing

TITLE: ETA Terrorists' image

QUESTION: Which of these two statements do you most agree with?

- ETA terrorists are criminals, heartless delinquents
- ETA terrorists are idealist freedom fighters.

In these two examples, the titles turn out to be almost a personal interpretation of the question as well as answer categories. In the next example, the situation is slightly different, since the title implies recovering information that is absent in the question (underlined in the example).

TITLE: Adequacy of the training provided by the company to do the job
 QUESTION: Q13 Have you been provided with information and training to do your job?
 -- Yes, enough -- Yes, but insufficient -- No, but I can managed – No and I have difficulties

These examples show quite a complex process of human interpretation of the question and answer categories, like making explicit the implicit, synonymy and paraphrasing. Their automatic processing will call at deep natural language processing techniques, accompanied by domain knowledge, computational lexicons and grammars for natural language understanding and generation. Since we look for a quick and unexpensive solution, deep natural language processing falls out of the scope of this work.

3.1.2 Objective Titles

These titles refer to an objective piece of information about the interviewee. Thus, in their linguistic structure there is not an initial word denoting a judgement but a concrete referent or property. Two types are distinguished: *specific* and *fixed*.

Objective Specific Titles. They are particular to a given survey and usually refer to habits like smoking, sports, leisure, possession of assets, acknowledgement of persons, etc. They follow the general schema of:

Initial word + nexus + Topic

Where initial word can be “Person”, “Entity”, “Possession”, “Likelihood”, “Frequency” ... and the topic modifies or characterizes the initial word.

The following is an example of an objective specific title and its question, where it can be observed a clear linguistic relation between both items: part of the title is included in the question (relevant fragments are underlined in the question).

TITLE: Likelihood that Communities will rise, lower, or leave as they are, taxes
 QUESTION: Once implemented the new system, and provided that Autonomous Communities can partially modify tax rates of the Income Tax, what do you think is more likely to happen: that communities will raise taxes, lower taxes or leave them as they are?
 - Will raise taxes | - Will lower taxes | - Will leave them as they are

The extraction of the relevant pieces of information in this type of titles does not require deep natural language understanding as in the examples of 3.1.1, shallow text processing techniques and even regular expressions will suffice to process them.

Objective Fixed Titles. They are obligatory in all surveys and refer to the so-called socio demographic variables like Sex, Age, Labour situation or Social class of the interviewee. Apart from any consideration about their linguistic features, titles under this category present two characteristics: they are very frequent and they are exactly repeated over all their occurrences. For example, the following question is exactly repeated 1564 times in the corpus with its exact title:

TITLE: Age of the Interviewee
 QUESTION: How old were you in your last birthday?

So these fixed titles could represent the simplest case of the problem, where a fixed title is assigned to a finite set of questions without further linguistic analysis.

After this preliminary analysis, it is evaluated the amount of questions belonging to the different types present in the corpus of titled questions in order to apply the same percentages to the bulk of untitled questions. It is also interesting to obtain the estimated quantity of titles that result from an interpretation of the whole question (referred as *Non Assignable* titles). To do that, a sample of 240 titles are analyzed, that for a confidence interval of 95%, yields an error rate of 6.3. Estimated frequencies are given in table 1.

Table 1. Frequencies for the different title categories

Title Category	TOTAL	%
Non assignable Titles	59	24,58%
Objective Fixed Titles	89	37,08%
Objective Specific Titles	44	18,33%
Subjective Titles	48	20%

4 Resolution Strategy

This section describes the solution to the problem and how we approach the work in the light of the results of the preliminary analysis. In essence, a title can be viewed as the concatenation of relevant pieces of information that are present in the question (namely, initial word and the topic of the question) and these pieces of information have to be found in the question. For space and clarity reasons, we will have a closer look at objective specific questions in order to illustrate the resolution strategy.

Objective specific questions refer to an objective piece of information about the interviewee addressing a wide variety of topics, including frequencies of actions, possession of things, persons with a give feature or remembering of vote. Due to the variety of topics, the identification of this type of questions relies on a number of linguistic constructions like different configurations for wh-questions mainly (considering wh-questions those headed by the equivalent pronouns of *who*, *what*, *which*, *where*, *when* and *how* in Spanish). Let's have a closer look at two paradigmatic types of objective specific questions: those about frequency of actions and those about persons/entities that do something.

EXAMPLE 1: FREQUENCY OF ACTIONS

Consider the following untitled questions:

From the following types of alcoholic beverages, could you tell me how often you consume them? (INTERVIEWER: read each type of beverage and SHOW CARD G).

ONLY FOR THOSE WHO HAVE CONSULTED WITH A PHYSICIAN IN THE LAST TWO WEEKS (1 in Q8). Q9 How many times?

The wh-phrase present in both sentences (*how often* and *how many times*) clearly identifies the intention of the question (*Initial word*) as "Frequency". On the other

hand, the *Topic* part of the title is not included in the interrogative sentence. In the first example, the topic is expressed as the pragmatic focus of the text, heading the interrogative sentence. In the second case, the topic is expressed in the interviewers' instruction. This implies that it is required to identify focused extrasentential elements.

The grammar rules that cover these questions are (the rule is adapted to English, although it is originally expressed in Spanish):

```
IF Question =~ /How many times do you <anyWord>
[ObjectPronoun][QuestionMark]/
→ { InitialWord = "Frequency";
Content = FocusedTopic}
```

PROPOSED TITLE: Initial word + "for" + FocusedTopic

That is, the presence of an accusative pronoun in the interrogative sentence implies that the Content is expressed outside the sentence, and it triggers the rules for focused topics. The corresponding rule for identifying the focused topic is the following:

```
IF Question =~ /From [ARTICLE] following
<anySequenceOfWords> <PUNC|that>/
→ FocusedTopic = <anySequenceOfWords>
```

Where <anySequenceOfWords> is recognized by means of regular expressions. These two rules generate the title:

TITLE: Frequency of consuming the following types of alcoholic beverages
--

The second question is similarly processed: the topic is to be found in the interviewer's instruction. So when the interrogative sentence consists of the wh-pronoun and just one word, the topic is extracted from the interviewer's instruction text. Besides, when generating the title, it has to be converted into lowercase characters. The next rule applies:

```
IF Question =~ /ONLY FOR THOSE <WHO|THAT>
<anySequenceOfWords> (QuestionID)* How many times <PUNC>
→ { Initial Word = "Number of times"
Topic = lowercase(anySequenceOfWords) }
PROPOSED TITLE: Initial word + "that" + "the interviewee"
+ anySequenceOfWords
```

And it produces the following title:

TITLE: Number of times that the interviewee has consulted with physician in the last two weeks
--

EXAMPLE 2: "PERSON/ENTITY THAT ..."

Within Objective Specific questions, it is frequent to ask about the mere acknowledgement of persons or facts and prejudgement about people. This sort of questions revolve around the pronoun *who* (Sp. *quién*) with variations. Here are some paradigmatic examples:

FROM Q001 and Q013. ONLY FOR THOSE WHO AT PRESENT LIVE AT HOME WITH SOMEONE IN THEIR OWN OR RENTED HOUSING (More than 1 in P001 and 1 or 2 IN P013). Who is the holder of the rental or the owner of this housing? - - (Write down)

```
IF Question =~ /Who is (AnySequenceOfWords) "?"/
  → {   Initial Word → "Person that"
        Topic → AnySequenceOfWords  }
```

And the following title is generated:

TITLE: Holder of the rental or the owner of this housing

Thus, the general strategy follows a grammar-based approach; where each sentence is subject to the following processes:

1. Extraction of the initial word: identify the linguistic construction that hints it (be it in answer categories or the wh-pronoun).
2. Extraction of the topic
 - a. In the interrogative sentence
 - b. In the anteposed topic before the interrogative sentence
 - c. In the instructions to the interviewer.
3. Generation of the title.
 - a. Concatenate both items, add nexus if needed
 - b. Include formatting instructions like upper to lower case, substitution of demonstrative of “the”, pronoun *Usted* (En. you) for “the interviewee”.

5 Results and Conclusions

There are two main aspects to be evaluated: the quantity and the quality of the generated titles. Quantitative results are summarized in table 2. As can be seen, at the end of the process, we were able to generate 22347 titles and leaving apart 1627 questions as filtered ones. This means that we automatically titled around 47% of the questions.

Table 2. Results for untitled questions

Question Status	N
Titled Question	22347
Untitled Question	23990

We also reviewed the quality of the generated titles. To do that, we extracted a sample of 300 titles and evaluated their quality, focusing on two main aspects: legibility of the sentence and presence of relevant information. The average percentage of correct titles for all the samples was **96%**.

Thus after the evaluation, we can ask ourselves again whether our initial hypothesis were correct. From the quantitative point of view, our hypothesis about the frequency of the different types of questions is only partially correct. Untreated questions represented 24% of the titled questions, whereas they represent 50% of untitled questions. Fixed questions are also less numerous in the corpus of untitled questions. However, from a qualitative point of view, we have assured homogeneous and correct titles.

The obtained results made us think about the differences in the distribution of the frequency of the different types of questions. This shift is probably due to the evolution of society that is reflected in the topics of the questions. The followed techniques and strategies also deserve a reflection. Linguistic processing is kept to a minimum, since the linguistic resources are expensive. On the other hand, domain-dependent strategies prove to be highly efficient while quick to be developed.

Acknowledgments. The authors would like to thank Pilar Rey del Castillo for her continuous support and valuable help during the production and evaluation of this work.

References

1. Kupiec, J., Pedersen, J.O., Chen, F.: A trainable document summarizer. In: Proceedings of SIGIR-1995, Seattle, WA, pp. 68–73 (1995)
2. Brandow, R., Mitze, K., Rau, L.F.: Automatic condensation of electronic publications by sentence selection. *Information Processing and Management* 31(5), 675–686 (1995)
3. Lin, C., Hovy, E.: Identifying topics by position. In: Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-1997), pp. 283–290 (1997)
4. Osborne, M.: Using maximum entropy for sentence extraction. In: Proceedings of the Acl-2002 Workshop on Automatic Summarization, vol. 4 (2002)
5. Tseng, Y.-H., Lin, C.-J., Chen, H.-H., Lin, Y.-I.: Toward Generic Title Generation for Clustered Documents. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) AIRS 2006. LNCS, vol. 4182, pp. 145–157. Springer, Heidelberg (2006)
6. Kong, S.-y., Wang, C.-c., Kuo, K.-c., Lee, L.-s.: Automatic Title Generation for Chinese Spoken Documents with a delicate scored Viterbi algorithm. In: Spoken Language Technology Workshop, pp. 165–168. IEEE, Los Alamitos (2008)
7. Radev, D.R., McKeown, K.: Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics* 24(3), 469–500 (1998)
8. Tucker, R.I., Sparck Jones, K.: Between shallow and deep: An experiment in automatic summarising. Technical Report 632, Computer Laboratory, University of Cambridge (2005)
9. Saggion, H., Lapalme, G.: Generating informative-indicative summaries with SumUM. *Computational Linguistics* 28(4), 497–526 (2002)
10. Hahn, U., Reimer, U.: Knowledge-based text summarisation: Saliency and generalisation for knowledge base abstraction. In: Mani, Maybury (eds.) *Advances in Automatic Text Summarisation*, pp. 215–222. MIT Press, Cambridge (1999)
11. Spärck Jones, K.: Automatic summarising: The state of the art. *Information Processing and Management* 43, 1449–1481 (2007)

Multilingual Video Indexing and Retrieval Employing an Information Extraction Tool for Turkish News Texts: A Case Study

Dilek Küçük¹ and Adnan Yazıcı²

¹ Power Electronics Group
TÜBİTAK - Uzay Institute
06531 Ankara, Turkey

`dilek.kucuk@uzay.tubitak.gov.tr`

² Department of Computer Engineering
Middle East Technical University
06531 Ankara, Turkey
`yazici@ceng.metu.edu.tr`

Abstract. In this paper, a multilingual video indexing and retrieval system is proposed which relies on an information extraction tool, a hybrid named entity recognizer, for Turkish to determine the semantic annotations for the considered videos. The system is executed on a set of news videos in English and encompasses several other components including an automatic speech recognition system for English, an English-to-Turkish machine translation system, a news video database, and a semantic video retrieval interface. The performance evaluation demonstrates that the system components achieve promising results which provides evidence for the applicability of the system. The proposed system and its application on the video set are significant as they constitute a plausible case study targeting at the problem of multilingual video indexing and retrieval utilizing information extraction as the central technique for semantic video indexing.

1 Introduction

Broadcast news video archives keep increasing in size at an unprecedented rate in a wide range of languages. In order to enable users to benefit from these video archives, multilingual video retrieval systems are required, considering the fact that prospective users may not know the language of all of the available video archives. However, video retrieval systems are usually language-specific and are often proposed for videos in well-studied languages such as English, French, or Spanish.

Multilingual or cross-language information retrieval (CLIR) aims at the determination of relevant documents in a given language based on queries expressed in another language [1,2]. Although the topic is given considerable attention in general, to the best of our knowledge, there are comparatively less studies on

multilingual video indexing and retrieval such as [3] which presents a multilingual video search engine combining automatic speech recognition (ASR) and multilingual search engine technologies.

In this paper, we propose a multilingual video retrieval system which comprises an information extraction (IE) component for Turkish along with an ASR system for English and an English-to-Turkish machine translation (MT) system. Thereby, we demonstrate the feasibility of multilingual video retrieval provided that the aforementioned IE, ASR, and MT systems are available. The main contribution of the current case study is that the proposed system enables indexing of the videos in English with the extracted semantic information in Turkish, hence multilingual –or, cross-lingual– video retrieval is achieved through this semantic information. Therefore, to a certain degree, the system facilitates the exploitation of multilingual videos by people who do not know the language of these videos.

The rest of the paper is organized as follows: In Section 2, the proposed multilingual video retrieval system is described with details on the characteristics of its components and the evaluation results of these components. In Section 3, query execution and processing within the presented system is clarified with a sample query. Finally, Section 4 concludes the paper along with some plausible future research directions based on the current study.

2 System Description

The multilingual video indexing and retrieval system is presented schematically in Figure 1. This text-based system mainly relies on an ASR system to obtain the video texts from raw video archives. It is well-known that practical and ready-to-use ASR systems exist for well-studied languages like English, French, and Chinese. Within the course of this study, we consider a video data set in English and hence we utilize an available ASR system for English to transcribe these videos as will be clarified soon. Next, the resulting video texts are fed into a convenient MT system so that the original video texts (transcripts) are translated into a target language. As our target language is Turkish, we employ an English-to-Turkish MT system to arrive at the Turkish translations of the original transcripts. Finally, the IE component for Turkish executes on these translations to extract semantic information about the contents of the videos. A hybrid named entity recognizer for Turkish texts [4] is employed as the IE component of the system.

Apart from these components, the proposed system encompasses a news video database and a semantic video retrieval interface, which have been previously employed in the text-based semantic indexing and retrieval system proposed in [5]. The database stores the semantic information as well as production metadata regarding the underlying video data set and the interface facilitates multilingual video retrieval basically through boolean query formulations.

In the following subsections, we first describe the components of the proposed system and next we present information on the video data set in English together

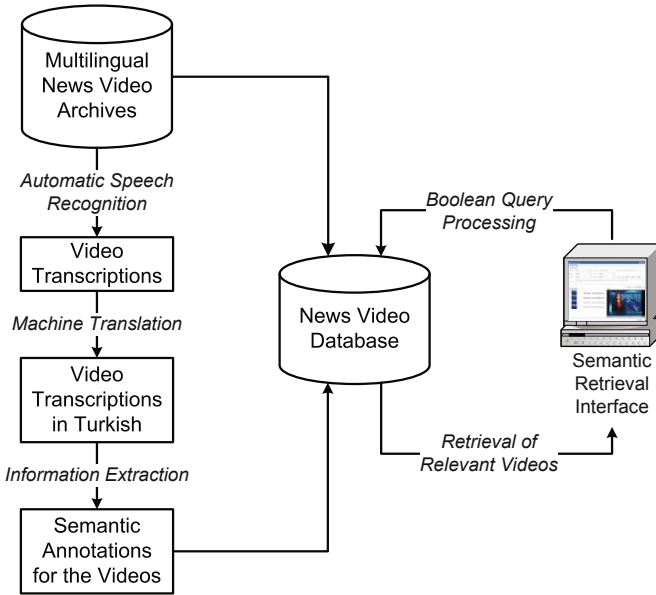


Fig. 1. The Multilingual Video Indexing and Retrieval System Employing An Information Extraction Tool for Turkish

with the evaluation results of the hybrid named entity recognizer on the Turkish translations of the transcripts of the videos in the set.

2.1 Main Components of the System

The Automatic Speech Recognition and Machine Translation Systems. The ASR system employed is the Sphinx system which carries out continuous speech recognition in a speaker independent manner [6]. The Sphinx system has prebuilt language models for several languages including English, Chinese, and French [7]. Hence, it is readily utilized to transcribe the videos in our video data set in English to be described in the following subsection. The particular version of the ASR system employed is PocketSphinx [8] which reportedly achieves word-error-rate values ranging between 9.73–13.95% on a 994-word task where better rates are obtained when speed is sacrificed and lower rates are obtained when the system is improved to execute faster.

As the MT system, the English-to-Turkish MT system proposed in [9] is employed to translate the transcripts to Turkish. The current BLEU score [10] of this MT system is 20.

The Hybrid Named Entity Recognition System for Turkish. The hybrid named entity recognition (NER) system for Turkish [4] is built upon the previously proposed rule based named entity recognizer for Turkish news texts [11]. The rule based recognizer relies on a set of manually engineered lexical resources

and pattern bases for the extraction of person, location, and organization names along with date/time and money/percentage expressions in generic news texts.

Due to the fact that rule based systems suffer from the portability problems, i.e., when they are ported to other domains they suffer from performance degradation, and also that it is necessary to keep the already existing information sources up-to-date, we equip the initial rule based recognizer with a rote learning [12] component and thereby turn it into a hybrid named entity recognizer. The ultimate hybrid recognizer is trained on other text types including financial news texts, child stories, and historical texts as well, hence it currently supports these text types in addition to generic news texts. This hybrid named entity recognizer [4] is exploited as the information extraction component of the proposed multilingual news video retrieval system.

The News Video Database and the Semantic Retrieval Interface. As pointed out previously, the news video database of the overall system stores production information regarding the raw videos of the video archive under consideration in addition to the outputs of the ASR, MT, and NER systems, namely, the original (noisy) speech transcripts of the videos, their Turkish translations, and the named entities extracted from these translations.

The semantic retrieval interface, previously presented in [5,13], is incorporated into the retrieval system to enable prospective users to access their videos of interest. As will be demonstrated in Section 3, users can specify their queries through boolean formulations utilizing named entities as literals. After the retrieval of the relevant videos satisfying a user query, the videos can be played through the interface. The original transcripts of the videos and the corresponding Turkish translations can also be examined through the interface. The users can also specify their queries in natural language (Turkish in this case) where the interface processes the natural language queries to determine the included named entities and using these entities it formulates the corresponding boolean queries. Thereafter, the resulting queries are processed exactly the same as boolean queries.

2.2 Evaluation and Discussion

The video data set on which the proposed system is executed comprises 23 videos broadcasted by Youtube [14], all of which belong to the category of *News & Politics*. The total duration of these videos is about 1 hour.

The videos are first given as input to the ASR system described in the previous section and their transcripts in English are obtained. Next, these transcripts are translated to Turkish by the English-to-Turkish MT system and hence we arrive at the speech transcripts of the original videos in Turkish.

The total number of tokens in the resulting text is 6,549 and the number of named entities in the text is 270 (42 person, 123 location, and 44 organization names, 52 date and 9 money expressions, with no instances of time or percent expressions). Overall evaluation results of our hybrid NER system on this text are given in Table 1 and the evaluation results for each named entity type are provided in Table 2.

Table 1. Evaluation Results of the Hybrid NER System on the Turkish Translations of the Original Transcripts of the Video Data Set in English

<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
58.20%	72.34%	64.50%

Table 2. Evaluation Results of the Hybrid NER System on the Turkish Translations of the Original Transcripts of the Video Data Set in English for Each Named Entity Type

<i>Named Entity Type</i>	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
Person	14.14%	32.53%	19.70%
Location	79.92%	84.49%	82.14%
Organization	41.94%	44.83%	43.33%
Date	92.38%	94.17%	93.27%
Money	100%	100%	100%

The evaluation results on the translation texts are comparatively lower than those results on genuine news video texts in Turkish provided in [5]. This is an expected result basically due to the proliferation of foreign names uttered in the videos which are not covered by our named entity recognizer. A deeper examination of the results in Table 2 reveals that the performance is especially hurt when person and organization names are considered. The results are particularly low for person name extraction since foreign person names are not covered by our recognizer (except the names of some well-known political figures) and also there are quite many common names in the translations which are erroneously extracted as person names resulting in a decrease in precision. The performance of the recognizer is superior during the extraction of location names as well as money and date expressions and is comparable to the results on genuine Turkish texts presented in [5]. Though the performance of the NER system on the translation texts is comparatively lower than its performance on original video transcripts in Turkish, the former results can be improved by extending the lexical resources and the pattern bases utilized by the NER system to cover common named entities in English.

3 Query Expression and Processing

We store production metadata regarding the video data set overviewed in the previous section, the original transcripts of the videos in this set, the corresponding translated texts in Turkish, and named entity information to the news video database. A query example over this video data set through the semantic retrieval interface of the system is provided in Figure 2.

The boolean query illustrated in Figure 2 includes the named entities of *Bush* (former president of the USA), *ABD* (*‘the USA’*), and *Amerika* (*‘America’*) which are combined with boolean operators. This query is also provided below where each named entity is expressed as a surface form/named entity type pair.

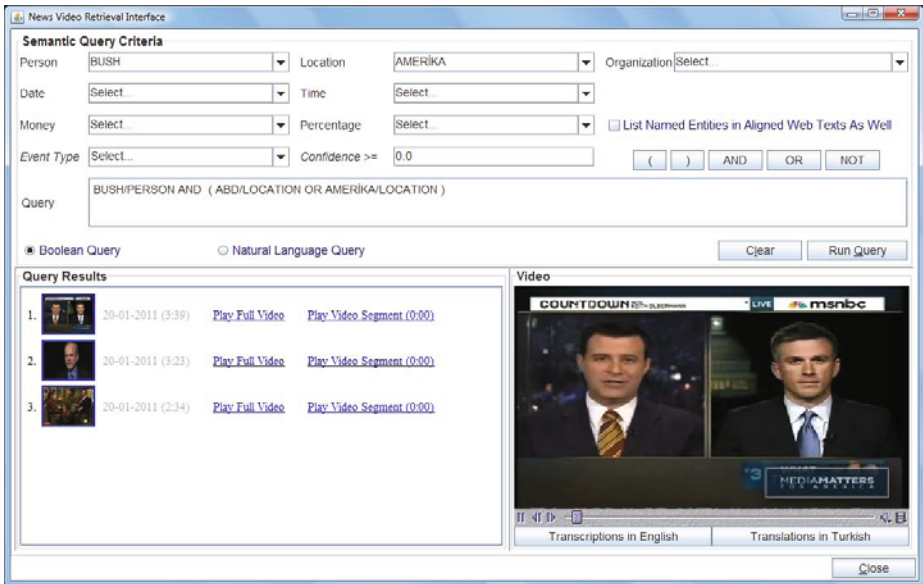


Fig. 2. A Boolean Query Example Over the Video Data Set in English Through the Semantic Retrieval Interface

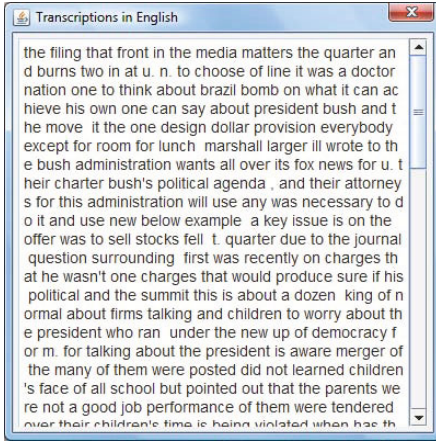
BUSH/PERSON AND (ABD/LOCATION OR AMERİKA/LOCATION)

The boolean queries are transformed into convenient SQL expressions to be executed on the news video database. After the execution of the queries, the satisfying videos are listed on the corresponding result panel.

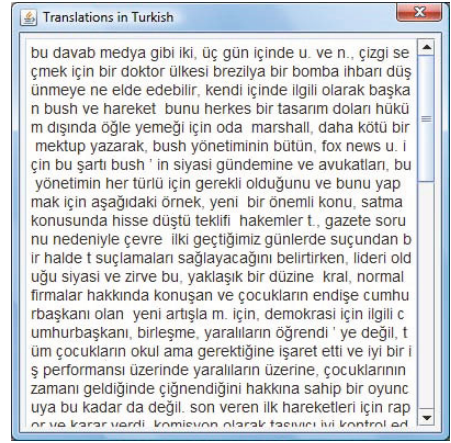
Within the proposed semantic video indexing and retrieval system, all of the video transcription, machine translation, and subsequent information extraction tasks are carried out off-line. Therefore, queries are executed over the previously extracted information which leads to highly favorable overall query execution times for the current –considerably small-scale– video data set. Yet, it is a plausible future direction of research to employ this system for a large-scale video data set to examine its performance in such settings.

After the retrieval of the relevant videos, the semantic video retrieval interface also enables the users to examine the English transcripts of the videos and their translations in Turkish through the corresponding buttons at the bottom of the video panel, as illustrated in Figure 3(a) and Figure 3(b). With this extension, the users who do not know the language of the original videos (English, in this case) will also be able to examine the translations of the original video transcripts through the interface.

To sum up, named entities extracted from the Turkish translations of the original transcripts of the videos provide a plausible means to access the videos over this semantic information. If ASR systems for some other languages and MT systems from these languages to Turkish can be supplied, then this system



(a) English Transcripts.



(b) Translations in Turkish.

Fig. 3. English Transcripts and Their Translations in Turkish Corresponding to the Selected Video in Figure 2

can be implemented for video archives in these languages as well, hence the overall system facilitates multilingual semantic video retrieval.

4 Conclusion

Video archives in a wide range of languages are increasing in size which requires multilingual indexing facilities to make them available to users who do not know the video language. In this paper, we present a multilingual video indexing and retrieval system to target at this problem. The implemented system is executed on a news video data set in English and utilizes an information extraction tool for Turkish texts to determine the semantic annotations for the videos. Other components of the system include an automatic speech recognition system for English, an English-to-Turkish machine translation system, a news video database, and lastly a semantic video retrieval interface. As the information extraction component, a hybrid named entity recognizer is exploited which achieves promising results on the Turkish translations of the original transcripts of the video data set. The system is significant as it demonstrates the feasibility of multilingual video retrieval provided that convenient systems can be accessed for the tasks of automatic speech recognition, machine translation, and information extraction. Below listed are some plausible future research directions based on the current study:

- The named entity recognizer component of the proposed system can be improved to extract named entities in English as well, or an existing named entity recognizer for English can be incorporated into the overall system. Thereby, the performance of the IE procedure can be considerably increased.

- As exploited in the semantic video indexing and retrieval system for Turkish proposed in [13], several other IE components for Turkish such as full person entity extractors and event extractors can be integrated into the proposed system. Doing so, relevant videos can be retrieved through semantic queries specified using such diverse types of information.
- The video data set utilized can be extended in size so that the ultimate evaluation results can better help to improve the system components.

Acknowledgments. This work is supported in part by a research grant from TÜBİTAK EEEAG with grant number 109E014.

References

1. Grefenstette, G., Segond, F.: Multilingual on-line natural language processing. In: Mitkov, R. (ed.) *The Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford (2003)
2. Oard, D.W., He, D., Wang, J.: User-assisted query translation for interactive cross-language information retrieval. *Information Processing and Management* 44(1), 181–211 (2008)
3. Delezoide, B., Le Borgne, H.: SemanticVox: a multilingual video search engine. In: *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, pp. 81–84 (2007)
4. Küçük, D., Yazıcı, A.: A hybrid named entity recognizer for Turkish with applications to different text genres. In: *Proceedings of the International Symposium on Computer and Information Sciences*, pp. 113–116 (2010)
5. Küçük, D., Yazıcı, A.: A text-based fully automated architecture for the semantic annotation and retrieval of Turkish news videos. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–8 (2010)
6. Lee, K.F., Reddy, R.: *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. Kluwer Academic Publishers, Norwell (1988)
7. CMU Sphinx Home Page, <http://cmusphinx.sourceforge.net/wiki/> (accessed February 21, 2011)
8. Huggins-Daines, D., Kumar, M., Chan, A., Black, A.W., Ravishankar, M., Rudnický, A.I.: Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In: *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP* (2006)
9. Köprü, S., Yazıcı, A.: Lattice parsing to integrate speech recognition and rule-based machine translation. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 469–477 (2009)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 311–318 (2002)
11. Küçük, D., Yazıcı, A.: Named entity recognition experiments on Turkish texts. In: *Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 524–535*. Springer, Heidelberg (2009)

12. Freitag, D.: Machine learning for information extraction in informal domains. *Machine Learning* 39(2-3), 169–202 (2000)
13. Küçük, D., Yazıcı, A.: Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos. *Knowledge-Based Systems* 24(6), 844–857 (2011)
14. Youtube, <http://www.youtube.com/> (accessed February 21, 2011)

A Search-Engine Concept Based on Multi-feature Vectors and Spatial Relationship

Tatiana Jaworska

Systems Research Institute, Polish Academy of Sciences,
6 Newelska Street, Warsaw, Poland
Tatiana.Jaworska@ibspan.waw.pl

Abstract. At present a great deal of research is being done in different aspects of Content-Based Image Retrieval System (CBIR). Unfortunately, these aspects are mostly analysed separately. We propose how to put together vectors of features for segmented objects and a spatial relationship of the objects. To achieve this goal we have constructed a search engine taking into account multi-set data mining and object spatial relationship. Additionally, we have constructed a graphical user interface (GUI) to enable the user to build a query by image. The efficiency of our system will be evaluated in the near future. In this paper we present the search engine for our CBIR.

Keywords: CBIR, spatial relationship, search engine, GUI, feature vector, multi-set.

1 Introduction

In recent years, the availability of image resources on the WWW has increased tremendously. This has created a demand for effective and flexible techniques for automatic image retrieval. Although attempts to perform the Content-Based Image Retrieval (CBIR) in an efficient way have been made before, a major problem in this area has been computer perception. In other words, there remains a considerable gap between image retrieval based on low-level features, such as shape, colour, texture and spatial relations considered separately, and image retrieval based on high-level semantic concepts that perceive an image as a complex whole. This problem becomes especially challenging when image databases are exceptionally large.

Images and graphical data are complex in terms of visual and semantic contents. Depending on the application, images are modelled and indexed using their

- visual properties (or a set of relevant visual features),
- semantic properties,
- spatial or temporal relationships of graphical objects.

Over the last decade a number of concepts of the CBIR [1], [2], [3], [4], have been used. In Wikipedia we can also find a list of CBIR engines, used either for commercial or academic research purposes [5].

Proposals can be found for the relational [6], object-oriented [7], [8] and object-relational database models [9]. Nevertheless, programmers have limited tools when they need to develop graphical applications dealing with imperfect pictorial data. Within the scope of semantic properties, as well as graphical object properties the first successful attempt was made by Candan and Li [10] who constructed the Semantic and Cognition-based Image Retrieval (SEMCOG) query processor to search for images by predicting their semantic and spatial imperfection. This new approach has been very important because earlier, and even present-day, queries to the database are put as query-by-example images.

Hence, in order to give the user the opportunity to compose their own image, consisting of separate graphical objects as a query, we have had to create our own system. An image created in GUI has its own unique object location in the image space. Thus, many researchers Chang [11,12], Chang and Wu, [13, 14], Zhou et al., [15] highlighted the importance of perceiving spatial relationships existing among the components of an image for efficient representation and retrieval of images in the CBIR.

We have dealt successfully with numerous problems involved in the CBIR system, with one final issue that still requires our attention. Ultimately, we have managed to form a new paradigm in comparing images with the search engine.

In this paper we present a concept of a search engine which takes into account object feature vectors, together with different spatial location of segmented objects in the image. In order to improve the comparison of two images, we need to label these objects in a semantic way.

1.1 CBIR Concept Overview

In general, our system consists of four main blocks (fig. 1):

1. The image preprocessing block (responsible for image segmentation), applied in Matlab, cf. [16];
2. The Oracle Database, storing information about whole images, their segments (here referred to as graphical objects), segment attributes, object location, pattern types and object identification, cf. [17];
3. The search engine responsible for the searching procedure and retrieval process based on the feature vector for objects and spatial relationship of these objects in the image, applied in Matlab;
4. The graphical user's interface (GUI), also applied in Matlab.

A query by image allows users to search through databases to specify the desired images. It is especially useful for databases consisting of very large numbers of images. Sketches, layouts or structural descriptions, texture, colour, sample images, and other iconic and graphical information can be applied in this search.

An example query might be: *Find all images with a pattern similar to this one*, where the user has selected a sample query image. In the QBIC system [3] the images are retrieved based on the above-mentioned attributes separately or using distance functions between features. Tools in this GUI include some basic objects, such as: polygon outliner, rectangle outliner, line draw, object translation, flood fill, eraser, etc. More advanced systems enable users to choose as a query not only whole images

but also individual objects. The user can also draw some patterns, consisting of simple shapes, colours or textures [18]. In the SEMCOG query processor [10], the user could organize an image as a spatial composition of five semantic groups of objects, such as: car, woman, man, house and bicycle. Additionally, the user could choose the colour, size and shape of a graphical object. In order to retrieve a matched image, the system integrated an image query statement with non-image operation statement.

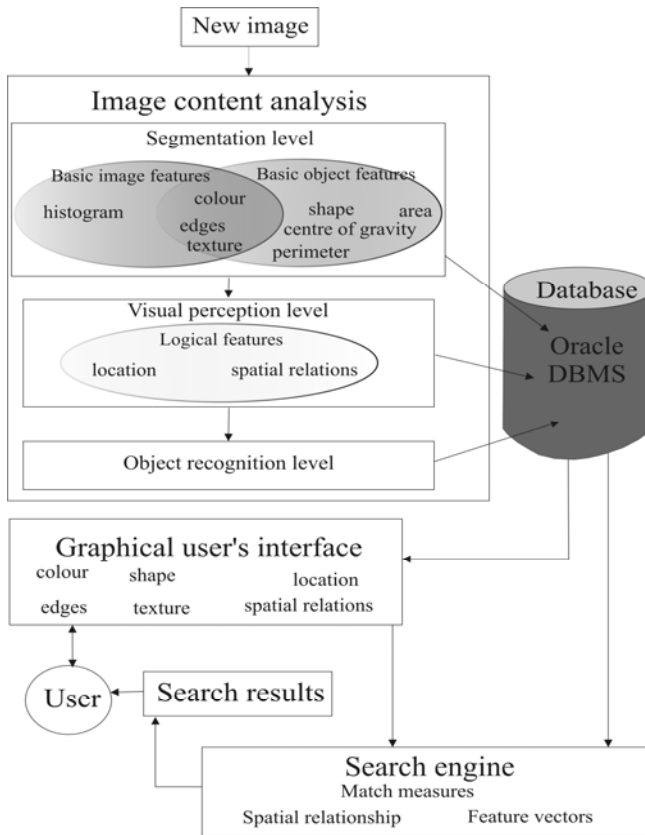


Fig. 1. Block diagram of our content-based image retrieval system

There have been several attempts made by the research community to disperse the demands in the design of efficient, invariant, flexible and intelligent image archival and retrieval systems based on the perception of spatial relationships. Chang [19] proposed a symbolic indexing approach, called the nine directional lower triangular (9DLT) matrix to encode symbolic images. Using the concept of 9DLT matrix, Chang and Wu [20] proposed an exact match of the retrieval scheme, based upon principal component analysis (PCA). Unfortunately, it turned out that the first principal component vectors (PCVs) associated with the image and the same image rotated are not the same. Eventually, an invariant scheme for retrieval of symbolic images based upon the PCA was prepared by Guru and Punitha [21].

2 Graphical Data Representation

In our system, Internet images are downloaded. Firstly, the new image is segmented, creating a collection of objects. Each object, selected according to the algorithm presented in detail in [16], is described by some low-level features. The features describing each object include: average colour k_{av} , texture parameters T_p , area A , convex area A_c , filled area A_f , centroid $\{x_c, y_c\}$, eccentricity e , orientation α , moments of inertia m_{11} , bounding box $\{bb_1(x,y), \dots, bb_s(x,y)\}$ (s – number of vertices), major axis length m_{long} , minor axis length m_{short} , solidity s and Euler number E and Zernike moments Z_{00}, \dots, Z_{33} . All features, as well as extracted images of graphical objects, are stored in the DB. Let F be a set of features where:

$$F = \{k_{av}, T_p, A, A_c, \dots, E\} \quad (1)$$

For ease of notation we will use $F = \{f_1, f_2, \dots, f_r\}$, where r – number of attributes. For an object, we construct a feature vector O containing the above-mentioned features:

$$F_O = \begin{bmatrix} O(k_{av}) \\ O(T_p) \\ O(A) \\ \vdots \\ O(Z_{33}) \end{bmatrix} = \begin{bmatrix} O(f_1) \\ O(f_2) \\ O(f_3) \\ \vdots \\ O(f_r) \end{bmatrix}. \quad (2)$$

The average colour is an average of each red, green and blue component which is summed up for all the pixels belonging to an object, and divided by the number of object pixels $k_{av} = \{r_{av}, g_{av}, b_{av}\}$. The next complex feature attributed to objects is texture. Texture parameters are found in the wavelet domain (the Haar wavelets are used). The algorithm details are also given in [16]. The use of this algorithm results in obtaining two ranges for the horizontal object dimension h and two others for the vertical one v :

$$T_p = \left\{ \begin{array}{l} h_{\min,1,2}; h_{\max,1,2} \\ v_{\min,1,2}; v_{\max,1,2} \end{array} \right\}. \quad (3)$$

Additional features of the low level for objects are shape descriptors. They are also included in the above mentioned feature vector. We apply the two most important shape descriptors such as moments of inertia:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y), \quad p, q = 0, 1, 2 \quad (4)$$

and Zernike moments [22]. Zernike moments are a set of complex polynomials $\{V_{pq}(x,y)\}$ which form a complete orthogonal set over the unit disk of $x^2 + y^2 \leq 1$. Hence, the definition of 2D Zernike moments with p^{th} order with repetition q for intensity function $f(x,y)$ of the image is described as:

$$Z_{pq} = \frac{p + 1}{\pi} \iint_{x^2 + y^2 \leq 1} V_{pq}^*(x, y) f(x, y) dx dy \tag{5}$$

where: $V_{pq}^*(x, y) = V_{p, -q}(x, y)$. (6)

For our purpose, the first 10 Zernike moments are enough, it means we calculate moments from Z_{00} to Z_{33} . Fig. 2 presents a module applied to find similarities between separate segmented elements.

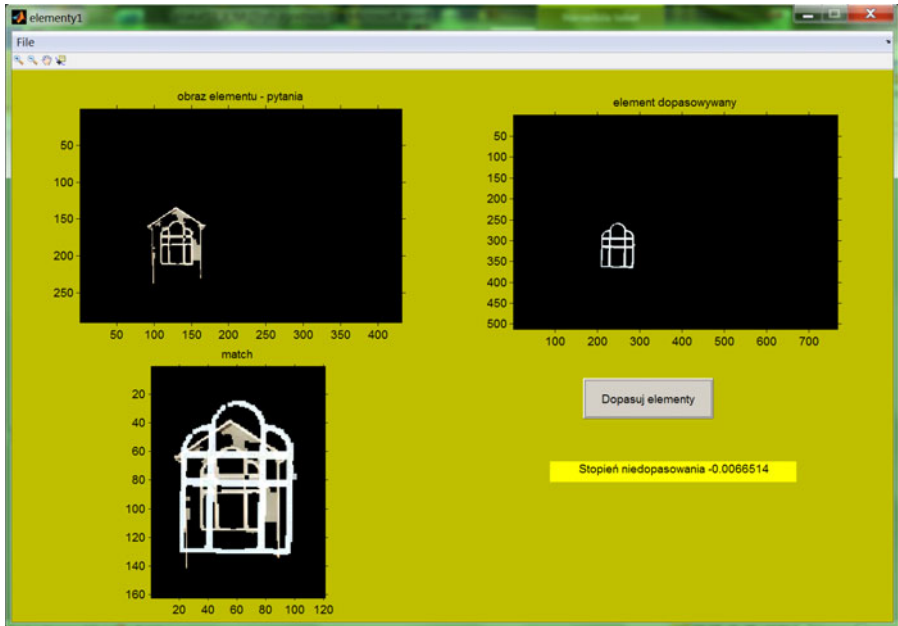


Fig. 2. An example of two matched objects based on the minimal Euclidean distance for the first 10 Zernike moments. These two elements were matched from the objects segmented from the images presented above.

Characteristic features of Zernike moments are:

1. The above-defined Zernike moments are only invariant to rotation.
2. The translation invariance is achieved by the location of the original image centroid in the centre of the coordinates.
3. The scale invariance is obtained by normalizing Z_{00} by the total number of image pixels.

3 Spatial Relationship of Graphical Objects

The feature vector F_o (cf. (2)) is further used for object classification. Therefore, we have to classify objects first in order to assign them to a particular class and second in order to compare objects coming from the same class [23].

In our system spatial object location in an image is used as the global feature. Firstly, it is easy for the user to recognize this spatial location visually. Secondly, it supports full identification based on rules for location of graphical elements. Let us assume that we analyse a house image. Then, for instance, an object which is categorized as a window cannot be located over an object which is categorized as a chimney. For this example, rules of location mean that all architectural objects must be inside the bounding box of a house. For an image of a Caribbean beach, an object which is categorized as a palm cannot grow in the middle of the sea, and so on. For this purpose, the mutual position of all objects is checked. The location rules are also stored in the pattern library [23]. Thirdly, object location reduces the differences between high-level semantic concepts perceived by humans and low-level features interpreted by computers.

For the comparison of the spatial features of two images an image I_i is interpreted as a set of n objects composing it:

$$I_i = \{o_{i1}, o_{i2}, \dots, o_{in}\} \quad (7)$$

Each object o_{ij} is characterized by a unique identifier and a set of features discussed earlier. This set of features includes a centroid $C_{ij} = (x_{ij}, y_{ij})$ and a label L_{ij} indicating the class of an object o_{ij} (such as window, door, etc.), identified in the process described in [23]. For convenience, we number the classes of the objects and thus L_k 's are just numbers.

Formally, let I be an image consisting of n objects and k be a number of different classes of these objects, $k \leq N$, because usually there are some objects of the same type in the image, for example, there can be four windows in a house.

Let us assume that there are, in total, M classes of the objects recognized in the database, denoted as labels L_1, L_2, \dots, L_M . Then, by the signature of an image I_i (7) we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \dots, \text{nobc}_{iM}] \quad (8)$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i , i.e. such objects o_{ij} .

Additionally, for an image I_i we consider a representation of spatial relationships of the image objects. The object's o_{ij} mutual spatial relationship is calculated based on

the algorithm below. Now, we consider one image; let C_p and C_q be two object centroids with $L_p < L_q$, located at the maximum distance from each other in the image, i.e.,

$$\text{dist}(C_p, C_q) = \max \{ \text{dist}(C_i, C_j) \forall i, j \in \{1, 2, \dots, k\} \text{ and } L_i \neq L_j \} \tag{9}$$

where: $\text{dist}(\bullet)$ is the Euclidean distance between two centroids (see fig. 3). The line joining the most distant centroids is the line of reference and its direction from centroid C_p to C_q is the direction of reference for computed angles θ_{ij} between other centroids. This way of computing angles makes the method invariant to image rotation.

Hence, we received triples (L_i, L_j, θ_{ij}) where the mutual location of two objects in the image is described in relation to the line of reference (see fig. 3 bottom). Thus, there are $T=m(m-1)/2$ numbers of triples, generated to logically represent the image consisting of m objects. Let S be a set of all triples, then we apply the concept of principal component analysis (PCA) proposed by Chang and Wu [20] and later modified by Guru and Punitha [21] to determine the first principal component vectors (PCVs).

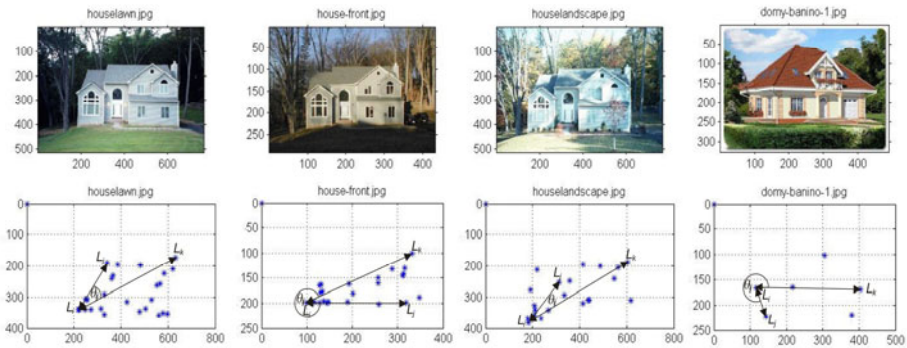


Fig. 3. Determination of angle relative to the reference direction for the construction of matrix S

First, we have to suppose that S is a set of observations for three variables. We construct a matrix of observations $X_{3 \times N}$ where each triple is one observation. Next, we count the mean value u of each variable, and we calculate the deviations from the mean to generate matrix $B = X - u \mathbf{1}$, where $\mathbf{1}$ - vector of all 1s. In the next step, we compute the covariance matrix $C_{3 \times 3}$ from the outer product of matrix B by itself as:

$$C = \mathbb{E} [B \otimes B] = \mathbb{E} [B B^*] = 1/N [B B^*]. \tag{10}$$

where: \mathbb{E} is the expected value operator, \otimes is the outer product operator, and $*$ is the conjugate transpose operator. Eventually, we find eigenvectors, which diagonalises the covariance matrix C :

$$V^{-1} C V = D \tag{11}$$

where: D is the diagonal matrix of the eigenvalues of C .

Using the Matlab procedure $V = \text{princomp}(X)$, we receive three component vectors (PCVs). For further analysis we use the first of them, which is the “spatial component” of the representation of an image I_i , and is denoted PCV_i .

For example, we use centroid coordinates from our CBIR to find angle θ_{ij} (see fig. 3 bottom). Thus, we construct set S of our observations, where N is combinations of the centroid numbers. For example, $N_{I_1} = C_2^{26} = 325$ and $N_{I_2} = C_2^{21} = 210$, respectively. The obtained results are shown in table 1.

Table 1. Representative principal component vectors for the images shown in fig. 3

Image name	First component	Second component	Third component
House-front	-0,001786	-0,003713	0,999992
Domy-banino-1	0,000206	0,003988	0,999992
Houselawn I_1	0,000388	0,001869	0,999998
Houselandscape I_2	0,004109	0,001557	0,999990

4 Construction of Search Engine

4.1 Graphical User Interface

Graphical User Interface (GUI) is a crucial element of our system as the area of human-computer interaction [24]. Hence, we have made an effort to create a useful tool for the user who is interested in designing their own image. This design is treated as a query by image. Fig. 4 presents the main GUI window entitled “Query_menu”. In the left window the user can choose the image outlines which become visible in an enlarged form in the main window.

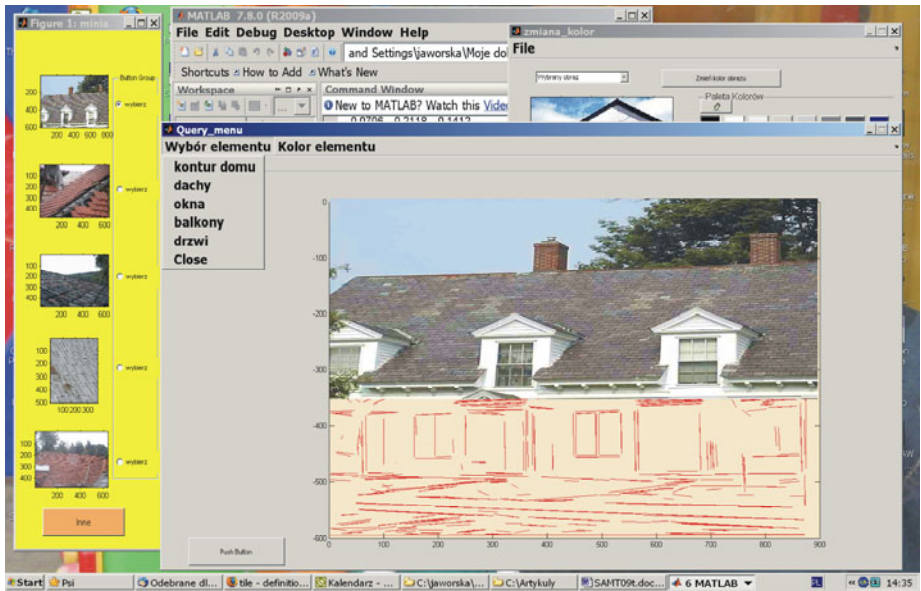


Fig. 4. The user menu applied by the system to design a query by image. The left window is used to present graphical elements, for example house roofs. It is easy to notice that the first roof at the top of the list of miniatures on the left is chosen and located in the house outline.

Next, the user chooses particular graphical elements from subsequent menus and places them on the appropriate location in the chosen outline. These elements can be scaled in a limited range. In most query-by-example systems, the features for retrieval and their importance are estimated by the system. Even in systems where such information can be provided by the user, users cannot always communicate unambiguously what they are looking for. In our system, these constraints are overcome by the user's selection of specific features (for example, the colour and texture of an object) from numerous menus. After the designing process, the image is sent as a query to the DB; it means that we have feature vectors F_{qi} (where $i=1, \dots, N$) for all objects used to form either query image I_q and PCV $_q$.

4.2 Similarities

So far, we have described how images are represented in our system. Now, we will describe how the similarity between two images is determined and used to answer a query. Let a query be an image I_q , such as $I_q = \{o_{q1}, o_{q2}, \dots, o_{qn}\}$ (cf. (7)). An image in the database will be denoted as $I_b, I_b = \{o_{b1}, o_{b2}, \dots, o_{bm}\}$. In order to answer the query, represented by I_q , we compare it with each image I_b in the database in the following way.

First of all, we determine a first similarity measure sim_{sgn} between I_q and I_b computing the Hamming distance $d_H(x,y) \in \mathbb{F}_{10}^{(M)}$ between the vectors of their signatures (8), i.e.:

$$\text{sim}_{\text{sgn}}(I_q, I_b) = d_H(\text{nobc}_q, \text{nobc}_b) \quad (12)$$

If the similarity (12) is smaller than a threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q . Otherwise, we proceed to the next step and we find the spatial similarity sim_{PCV} of images I_q and I_b computing the Euclidean distance between their PCVs as:

$$\text{sim}_{\text{PCV}}(I_q, I_b) = 1 - \sqrt{\sum_{i=1}^3 (PCV_{bi} - PCV_{qi})^2} \quad (13)$$

If the similarity (13) is smaller than the threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q . Otherwise, we proceed to the final step, namely, we compare the similarity of the objects representing both images I_q and I_b . For each object o_{qi} present in the representation of the query I_q , we find the most similar object o_{bj} of the same class, i.e., $L_{qi} = L_{bj}$. If there is no object o_{bj} of the class L_{qi} , then $\text{sim}_{\text{ob}}(o_{qi}, o_b)$ is equal to 0. Otherwise, similarity $\text{sim}_{\text{ob}}(o_{qi}, o_b)$ between objects of the same class is computed as follows:

$$\text{sim}_{\text{ob}}(o_{qi}, o_{bj}) = 1 - \sqrt{\sum_l (Fo_{qil} - Fo_{bjl})^2} \quad (14)$$

where l indexes the set of features F_O used to represent an object, as described in (2). When we find highly similar objects (for instance, $\text{sim}_{\text{ob}} > 0.9$), we eliminate these

two objects from the following process of comparison [25]. The process is realized according to the algorithm presented below:

```

Algorithm: Pair matching algorithm with elimination
k=0;
i=1;
j=1;
for j=j:Lqi %number of objects in a particular class
  for i=i:Lbj %number of objects in a particular class
    if sim(i,j)>.9
      match(i,j)=sim(i,j);
      row(i)=i;
      col(j)=j;
      j=j+1;
      i=i+1;
    end;
  end;
end;
while k==0
  [k,R]=min(row);
  [k,C]=min(col);
  match(R,C)=sim(R,C);
  row(R)=R;
  col(C)=C;
end;

```

Thus, we obtain the vector of similarities between the query I_q and an image I_b .

$$\text{sim}(I_q, I_b) = \begin{bmatrix} \text{sim}_{\text{ob}}(o_{q1}, o_{b1}) \\ \vdots \\ \text{sim}_{\text{ob}}(o_{qn}, o_{bn}) \end{bmatrix} \quad (15)$$

where n is the number of objects present in the representation of I_q .

In order to compare images I_b with the query I_q , we compute the sum of $\text{sim}_{\text{ob}}(o_{qi}, o_{bi})$ and then use the natural order of the numbers. Thus, the image I_b is listed as the first in the answer to the query I_q , for which the sum of similarities is the highest.

5 Conclusion

The construction of a CBIR system requires combining different functional systems, linked together and cooperating with each other. For this purpose, object classification and identification procedures have been established and the GUI prototype has been constructed.

We have prepared a model of image similarity as a three-step procedure. This is, of course, a preliminary model of a three-step procedure to answer a query. There are many other possible ways to compute the similarity between the images, e.g. using different metrics. Intensive computational experiments are under way in order to

come up with some conclusions as to the choice of the parameters of the model, including the choice of the above-mentioned metrics. However, the preliminary results we have obtained so far using the simplest configuration are quite promising.

References

1. Deb, S. (ed.): *Multimedia Systems and Content-Based Image Retrieval*, ch. VII and XI. IDEA Group Publishing, Melbourne (2004)
2. Flickner, M., Sawhney, H., et al.: *Query by Image and Video Content: The QBIC System*. *IEEE Computer* 28(9), 23–32 (1995)
3. Niblack, W., Flickner, M., et al.: *The QBIC Project: Querying Images by Content Using Colour, Texture and Shape*. In: *SPIE*, vol. 1908, pp. 173–187 (1993)
4. Ogle, V., Stonebraker, M.: *CHABOT: Retrieval from a Relational Database of Images*. *IEEE Computer* 28(9), 40–48 (1995)
5. http://en.wikipedia.org/wiki/List_of_CBIR_engines
6. Pons, O., Vila, M.A., Kacprzyk, J.: *Knowledge management in fuzzy databases*. *Studies in Fuzziness and Soft Computing*, vol. 39. Physica –Verlag, Heidelberg (2000)
7. Lee, J., Kuo, J.-Y., Xue, N.-L.: *A note on current approaches to extent fuzzy logic to object oriented modeling*. *International Journal of Intelligent Systems* 16, 807–820 (2001)
8. Berzal, F., Cubero, J.C., Kacprzyk, J., Marin, N., Vila, M.A., Zadrozny, S.: *A General Framework for Computing with Words in Object-Oriented Programming*. In: Bouchon-Meunier, B. (ed.) *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.*, vol. 15(Supplement), pp. 111–131. World Scientific Publishing Company, Singapore (2007)
9. Cubero, J.C., Marin, N., Medina, J.M., Pons, O., Vila, M.A.: *Fuzzy Object Management in an Object-Relational Framework*. In: *Proceedings of the 10th International Conference IPMU, Perugia, Italy*, pp. 1775–1782 (2004)
10. Candan, K.S., Li, W.-S.: *On Similarity Measures for Multimedia Database Applications*. *Knowledge and Information Systems* 51(3), 30–51 (2001)
11. Chang, S.K., Shi, Q.Y., Yan, C.W.: *Iconic indexing by 2D strings*. *IEEE Trans. Pattern Anal. Machine Intell.* 9(5), 413–418 (1987)
12. Chang, S.K., Jungert, E., Li, Y.: *Representation and retrieval of symbolic pictures using generalized 2D string*. In: *SPIE Proc. on Visual Comm. and Image Process*. Philadelphia, pp. 1360–1372 (1989)
13. Chang, C.C., Wu, T.C.: *Retrieving the most similar symbolic pictures from pictorial databases*. *Informat. Process. Manage.* 28(5), 581–588 (1992)
14. Wu, T.C., Chang, C.C.: *Application of geometric hashing to iconic database retrieval*. *Pattern Recognition Letters* 15, 871–876 (1994)
15. Zhou, X.M., Ang, C.H., Ling, T.W.: *Image retrieval based on object's orientation spatial relationship*. *Pattern Recognition Letters* 22, 469–477 (2001)
16. Jaworska, T.: *Object extraction as a basic process for content-based image retrieval (CBIR) system*. In: *Opto-Electronics Review*, Association of Polish Electrical Engineers (SEP), Warsaw, vol. 15(4), pp. 184–195 (2007)
17. Jaworska, T.: *Database as a Crucial Element for CBIR Systems*. In: *Proceedings of the 2nd International Symposium on Test Automation and Instrumentation*, vol. 4, pp. 1983–1986. World Publishing Corporation, Beijing (2008)
18. Smith, J.R., Chang, S.-F.: *Integrated spatial and feature image query*. *Multimedia Systems* 7, 129–140 (1999)

19. Chang, C.C.: Spatial match retrieval of symbolic pictures. *J. Informat. Sci. Eng.* 7, 405–422 (1991)
20. Chang, C.C., Wu, T.C.: An exact match retrieval scheme based upon principal component analysis. *Pattern Recognition Letters* 16, 465–470 (1995)
21. Guru, D.S., Punitha, P.: An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis. *Pattern Recognition Letters* 25, 73–86 (2004)
22. Teague, M.R.: Image analysis via the general theory of moments. In: *JOSA*, 8th edn., vol. 70, pp. 920–930 (1980)
23. Jaworska, T.: Graphical Object Classification and Query by Image as an Aspect of Content-Based Image Retrieval System. In: Owskiński, J. (ed.) *Studies and Materials of Polish Operational and Systems Research Society*, Bydgoszcz, Poland, vol. 32, pp. 269–282 (2010)
24. Newman, W.M., Lamming, M.G.: *Interactive System Design*. Addison-Wesley, Harlow (1996)
25. Mucha, M., Sankowski, P.: Maximum Matchings via Gaussian Elimination. In: *Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS 2004)*, pp. 248–255 (2004)

Exploiting Class-Specific Features in Multi-feature Dissimilarity Space for Efficient Querying of Images

Turgay Yilmaz, Adnan Yazici, and Yakup Yildirim*

Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey
{turgay,yazici}@ceng.metu.edu.tr

Abstract. Combining multiple features is an empirically validated approach in the literature, which increases the accuracy in querying. However, it entails processing intrinsic high-dimensionality of features and complicates realizing an efficient system. Two primary problems can be discussed for efficient querying: representation of images and selection of features. In this paper, a class-specific feature selection approach with a dissimilarity based representation method is proposed. The class-specific features are determined by using the representativeness and discriminativeness of features for each image class. The calculations are based on the statistics on the dissimilarity values of training images.

1 Introduction

CBIR systems aim to retrieve pictures from large image repositories according to the needs of the users [6]. In CBIR systems, images are usually modelled with a set of low level features, such as color, texture or shape, from which underlying similarity functions are used to perform queries [1]. The ultimate goal of designing CBIR systems is to achieve the best possible retrieval accuracy. To achieve high accuracy on a retrieval task, traditional approaches prefer creating superior low level features than the currently available ones, or optimization of them [5, 14]. However, the noise in sensed data, non-universality of any single low level feature and performance upper bounds prevent relying on a single feature [22]. In the information fusion literature, fusing multiple features is an empirically validated approach for increasing the retrieval performance [8, 14, 17, 27].

Dealing with multiple features entails processing intrinsic high-dimensionality of each feature and handling heterogeneous dimensions / scales of different features. Modelling the CBIR system to operate in feature space (storing image features in the database) makes the system struggle with the heterogeneity of different features and prevents it from being fast and flexible [3]. Such a system is not fast since similarity calculation is done at query-time. Also, it cannot be

* This work is supported in part by a research grant from TÜBİTAK EEEAG with grant number 109E014.

flexible either, considering that handling a new feature requires renewing the system for processing the dimensionality and scale of the new feature. Therefore, an alternative approach, that regards the fastness and the flexibility issues, is modelling the system in dissimilarity space. In accordance with the ideas of [20, 21] for representing images with dissimilarities, Bruno et al. [4] present fusing multiple features in dissimilarity space. In dissimilarity space, the images in the database are represented with the dissimilarity values to prototype objects of the particular image categories. Thus, both the retrieval operation is faster and adding new features to the system is easier as long as the distance function is available at once for processing the images in the database.

Beyond the representation problem of images, another crucial issue is to find out the features that are more beneficial for fusion. This problem, namely the feature selection problem, tries to determine which subset of features yield to an optimal result. In [12], Jain et al. group widely-used techniques with a general aspect of view: exhaustive search, branch-and-bound search, best individual features, sequential forward/backward selection, sequential forward/backward floating search. The methods except the exhaustive search provide computationally better ways of finding an optimal set, however exhaustive search guarantees to find the optimal solution. For each of these methods, selection criteria during forward/backward selection operations can differ; information gain, previously-defined quality metrics or the complexity can be considerations. With a more specific view on the problem, some of the recent approaches in the information fusion literature can be listed as: Finding principal/independent components [16, 26], selecting the most coherent and less complex features according to the heterogeneity issue [15], calculating the information gain obtained [2, 13] and defining quality and reliability metrics on features [22, 23].

Although there are many different approaches for the selection of features, all of them have a common preference: The selection process is independent of the category (semantic meaning) of the images. However, considering the idea that different features can be more effective, representative and discriminative for different image categories, using a category dependent feature selection approach can be more beneficial.

In this study, we propose a class-specific feature selection approach for the fusion of multiple features. In order to eliminate the high-dimensionality of multiple features and provide efficient querying over the images, we prefer a dissimilarity based approach. To learn the class-specific features, we carry out a training phase. During the training, the class-specific features are determined by using the representativeness and discriminativeness of features for each image class. The calculations of representativeness and discriminativeness are based on the statistics on the dissimilarity values of training images.

The remainder of this paper is organized as follows: First, the multi-feature modelling in dissimilarity space is introduced in Section 2. Then, the class-specific feature selection approach is given in detail, in Section 3. In Section 4, the empirical results and the evaluations are presented. Lastly, in Section 6, some conclusions are drawn and further study is discussed.

2 Multi-feature Modelling in Dissimilarity Space

The literature of information fusion agrees on the idea that combining multiple features enhances the efficiency. However, how to combine such information is still studied. One of the discussed issues is the representation of images. In feature based representation, an image is usually represented with a multi-dimensional feature vector and having multiple features causes dealing with multiple of such multi-dimensional feature vectors, each having different dimensions and scales. Handling the complexity of different dimensions and scales of different features makes the CBIR system more dependent on the currently available features and less flexible to new features. In [3], Bruno et al. discuss these issues in detail. Still, a more crucial flaw for feature-based representation is the inefficiency of the fast querying capabilities. Having features in the database requires calculating the similarities of related images for every query task.

A more convenient way is the dissimilarity based representation [3,7,20,21]. In dissimilarity based representation, feature values are not stored in the database; instead the dissimilarity values of images are stored. Thus, the CBIR system does not need to deal with the intrinsic dimensionality of features to combine them. In addition, a query task is simpler; it does not require similarity calculations for each query. The dissimilarity values of images are calculated once, before including the image into the CBIR system. To calculate the dissimilarity values, the dissimilarity functions of each feature are utilized. Hence dissimilarity-based representation is a more flexible and fast way of representing the images in a CBIR system employing multiple features.

In dissimilarity based representation, the dissimilarities between each image couple is not necessary. Instead, the dissimilarities of the images in the image database with prototype images of the system are enough. The number of prototype images is quite smaller than the size of the image database. Usually, the prototype images are grouped according to their image classes (semantic meanings of images) in order to meet semantic query requirements. In a multi-feature CBIR system, such distance values between the images in the image database and the prototype images should be stored separately for each feature.

More formally, assuming that $F = \{f_1, f_2, ..f_k\}$ is the set of features available for the CBIR system having k number of features, $C = \{c_1, c_2, .., c_m\}$ is the image database having m number of images, $P = \{P_1, P_2, ..P_n\}$ is the set of prototype image classes containing n number of image classes, each prototype image class is $P_i = \{p_1^i, p_2^i, ..p_t^i\}$ where number of prototype images is t and t is not necessarily the same in all prototype image classes; the multi-feature CBIR system owns following distance-based representation for each image class i and feature f :

$$D_f^i = \begin{pmatrix} d_f(c_1, p_1^i) & d_f(c_1, p_2^i) & \cdots & d_f(c_1, p_t^i) \\ d_f(c_2, p_1^i) & d_f(c_2, p_2^i) & \cdots & d_f(c_2, p_t^i) \\ \vdots & \vdots & \ddots & \vdots \\ d_f(c_m, p_1^i) & d_f(c_m, p_2^i) & \cdots & d_f(c_m, p_t^i) \end{pmatrix} \quad (1)$$

where $d_f(x, y)$ is the dissimilarity between the database image x and the prototype image y for feature f .

A semantic query (for instance “Find pictures of cars”) executed in this kind of CBIR system is handled as follows: The distance matrices of D_f^i s are evaluated, where i is the class of ‘car’ images and $f \in F$. First, for each matrix, prototype aggregation with a predefined algorithm is performed and an aggregated distance vector that represents the distances of all images in the image database to the ‘car’ semantic image class is obtained. Then k number of distance vectors, each representing a different feature, are combined with a feature selection algorithm. The combination of k number of distance vectors results with a single distance vector which shows the distances of all database images to the ‘car’ class.

In this study, we propose a class-specific feature selection approach for the feature selection problem stated above. The prototype aggregation problem is beyond the scope of this paper. However two different basic aggregation methods (minimum and average) are utilized during the empirical study in order to see the effect of using different aggregation techniques.

3 Exploiting Class-Specific Features

In CBIR systems, as mentioned in Section 1, a particular feature or a common set of features is usually used to compare the query image with the database images. In these systems, the features are selected to represent the problem domain. However, if the size of the database and/or the diversity of image collection is increased, these methods fail to give satisfactory results. Specifically, using the same features for different domains and types of objects yields unsatisfactory results. Finding a solution to the problem is quite simple: using different features for different object types. For example, shape features are more important than color features for a ‘car’ object whereas a ‘sea’ object can be defined with color and texture features.

To describe the approach more formally, assume an image database having images from 2 semantic classes. It is assumed that class C_1 contains n_1 number of images and C_2 contains n_2 number of images in the database. Also, it is assumed that the images of class C_1 can be defined better with color features and the images of C_2 can be defined better with shape features. If this database is used in a CBIR system that compares images according to only color features or shape features, the performance of the system is nearly 50% in terms of accuracy. If color features are used, the performance of the system is satisfactory for C_1 , but not for C_2 . To obtain a satisfactory performance for the whole system, different features should be used for different classes.

By considering this idea, in [25], Uysal et al. utilized an approach identifying the Best Representative Feature (BRF) for each object class, which maximizes the correct match in a training set. Similarly, in [24] Swets et al. propose to use Most Expressive Features and Most Discriminating Features. However, these approaches lacks the advantages of fusing multiple features since they select only one feature for each class.

Besides, Jain et al. [10] apply the idea in biometrics domain. They propose combining multiple traits by selecting person-specific traits for recognition. However, they do not propose a feature selection methodology. They obtain the person-specific traits after an exhaustive search process on the training data.

In this study, we propose a class-specific feature selection mechanism by finding out the representative and discriminative features for each image class. Representative characteristics of features are calculated according to the dissimilarities of images within the same class, and discriminative characteristics are calculated according to the ability of features to distinguish between different image classes. Using these characteristics, the importance values of features for each image class are calculated as detailed below. The importance values of features for each category is also called the Class-Specific Features (CSF) index. The mechanism is based on statistical calculations over the dissimilarity values of all prototype images. Providing such prototype images can be considered as the training phase of the CBIR system. The CSF indices are used as the weights of the features during feature combination process.

3.1 Calculation of CSF Indices

To calculate the CSF indices, firstly the dissimilarity values of prototype images to each other is calculated and a dissimilarity matrix is obtained as $D_f^i(P)$ for each f , similar to the one given in Section 2. Differently, $D_f^i(P)$ includes dissimilarities of prototype images in image class i to all prototype images of all image classes. $D_f^i(P)$ contains $n \cdot t$ rows and t columns.

$$D_f^i(P) = \begin{pmatrix} d_f(p_1^1, p_1^i) & d_f(p_1^1, p_2^i) & \cdots & d_f(p_1^1, p_t^i) \\ \vdots & \vdots & \vdots & \vdots \\ d_f(p_t^1, p_1^i) & d_f(p_t^1, p_2^i) & \cdots & d_f(p_t^1, p_t^i) \\ \vdots & \vdots & \vdots & \vdots \\ d_f(p_1^n, p_1^i) & d_f(p_1^n, p_2^i) & \cdots & d_f(p_1^n, p_t^i) \\ \vdots & \vdots & \vdots & \vdots \\ d_f(p_t^n, p_1^i) & d_f(p_t^n, p_2^i) & \cdots & d_f(p_t^n, p_t^i) \end{pmatrix} \quad (2)$$

After obtaining the dissimilarity matrices $D_f^i(P)$ for each feature and image class, dissimilarity values of each image category in each matrix are aggregated both column-wise and row-wise. Thus, the mean and standard deviation vectors are obtained as follows;

$$\mu(D_f^i(P)) = [\mu_f^{i,1} \mu_f^{i,2} \cdots \mu_f^{i,n}]^T \quad (3)$$

$$\sigma(D_f^i(P)) = [\sigma_f^{i,1} \sigma_f^{i,2} \cdots \sigma_f^{i,n}]^T \quad (4)$$

Here, $\mu_f^{i,j}$ denotes the mean of dissimilarities from all images in class i to all images in class j for feature f . Also, $\sigma_f^{i,j}$ denotes the corresponding standard deviation.

To obtain the CSF indices, four important parameters are extracted from the above given vectors of $\mu(D_f^i(P))$ and $\sigma(D_f^i(P))$:

- Mean of Class ($\mu_f^{i,i}$): $\mu_f^{i,i}$ is the average dissimilarity value of a class to itself, for a particular feature f . Mean of Class is a representative characteristic for features. For a selected class, the features with lower dissimilarity values represent the image class better. Thus, the CSF index is inversely proportional to the mean of the category.
- Standard Deviation of Class ($\sigma_f^{i,i}$): $\sigma_f^{i,i}$ is another important representative property. For any class, a feature with small standard deviation entails close image-to-image dissimilarity values within the class. Such a feature can be considered as a better feature. Thus, the CSF index is inversely proportional to the standard deviation of an image class.
- Standard Mean Distance to Other Classes (δ_f^i): Standard mean distance to other classes is a discriminative feature which is calculated by using the dissimilarities of a class to other classes. It is calculated as follows:

$$\delta_f^i = \sqrt{\frac{\sum_{j=1}^n (\mu_f^{i,i} - \mu_f^{i,j})^2}{n}} \quad (5)$$

where n is the number of image classes. This calculation gives us the average dissimilarity of an image class i to all other classes. Thus, having a greater dissimilarity means better discrimination among all categories, which means that the CSF index is directly proportional to δ_f^i .

- Correctness Ratio (ω_f^i): Although the three parameters given above are important and provide good representation and discrimination, the issue of correctness of the feature is not considered. It is important for a feature to give the lowest dissimilarity values for the images in a class which is the same with the class of the query images. Correctness ratio of a particular feature f can be defined as what percentage of the means in a $\mu(D_f^i(P))$ vector are larger than the mean value of the class i ($\mu_f^{i,i}$). As the correctness ratio decreases, the representation ability decreases, which means that the CSF index is directly proportional with the correctness ratio.

Considering the effects of the above parameters, the CSF index of a particular feature f on a particular image class i is calculated using the formula below:

$$CSF_f^i = \frac{(1 - \mu_f^{i,i}) \cdot \delta_f^i \cdot \omega_f^i}{\sigma_f^{i,i}} \quad (6)$$

3.2 Normalization on Dissimilarities

As mentioned before, CBIR system having dissimilarity-based representation does not need to deal with the intrinsic dimensionality of features to combine them. However, different scales of different features are still a problem to be solved. Different scales of the values contained in the features causes dissimilarity values to be in different scales.

In the literature, there are several normalization methods to handle the different scales of multiple features [11]: Min-max, decimal scaling, z-score, median, double sigmoid, tanh estimators, biweight estimators. In [11], Jain et al. empirically show that min-max, z-score and tanh estimators methods are superior. Also they note that the simplest method (min-max) would suffice when the minimum and maximum values are known. Min-max normalization transforms values from a known (or estimated) range $[min, max]$ into $[0, 1]$ range with the following basic formulation: $x' = (x - min)/(max - min)$. Considering that we have the prototype images and dissimilarity values of prototype images to themselves, it is easy to find the minimum and maximum dissimilarity values for each feature. Thus the min-max normalization approach is preferred in this study.

4 Evaluation

To demonstrate the validity of the proposed approach, a number of experiments are carried out. For the experiments, the CalTech 101 image dataset [9] is used. It contains pictures of objects belonging to 101 categories. During the tests, all of the 101 classes in the dataset are used. Randomly selected 10 images for each class, hence a total of 1010 images, are treated as the prototype images. For the query purposes, randomly selected 20 images for each class and a total of 2020 images are employed the image database. In addition, as the features to be combined, 8 visual features of MPEG-7 [18] in three types are utilized: Color descriptors of Color Layout(CL), Color Structure(CS), Dominant Color(DC), Scalable Color(SC); Shape descriptors of Contour Shape(CSh), Region Shape(RS); Texture descriptors of Edge Histogram(EH), Homogeneous Texture(HT). The dissimilarities of the images for these features are calculated by using the MPEG-7 reference software (eXperimentation Model, XM) [19].

The tests are mainly performed on semantic retrieval of images; the semantic classes are queried over the image database. The images are fetched and sorted according to the dissimilarity values. To measure the retrieval accuracy, *Precision*, *Recall*, *Average Precision(AP)* and *Mean Average Precision(MAP)* metrics are used. *Precision* is the fraction of retrieved images that are relevant to the search, whereas *Recall* is the ratio of the number of relevant images retrieved to the total number of relevant images in the collection. The *AP* is the sum of the precision at each relevant hit in the retrieved list, divided by the minimum between the number of relevant documents in the collection and the length of the list. Considering that image collection in our test contains 2020 images, *AP* is measured at 2020. *MAP* is the *AP* averaged over several image classes.

As the primary test, the accuracy of the proposed method on semantic retrieval is measured. In order to perform a detailed comparison, this test is executed in four steps. As the first step, the retrieval accuracies of each single feature is calculated. For the second step, following simple combination approaches are tested: Minimum Distance(MD), Average Distance(AD), Euclidian Distance(ED). The combined dissimilarity is obtained by selecting the minimum dissimilarity (distance) in MD, averaging all available dissimilarities in AD and

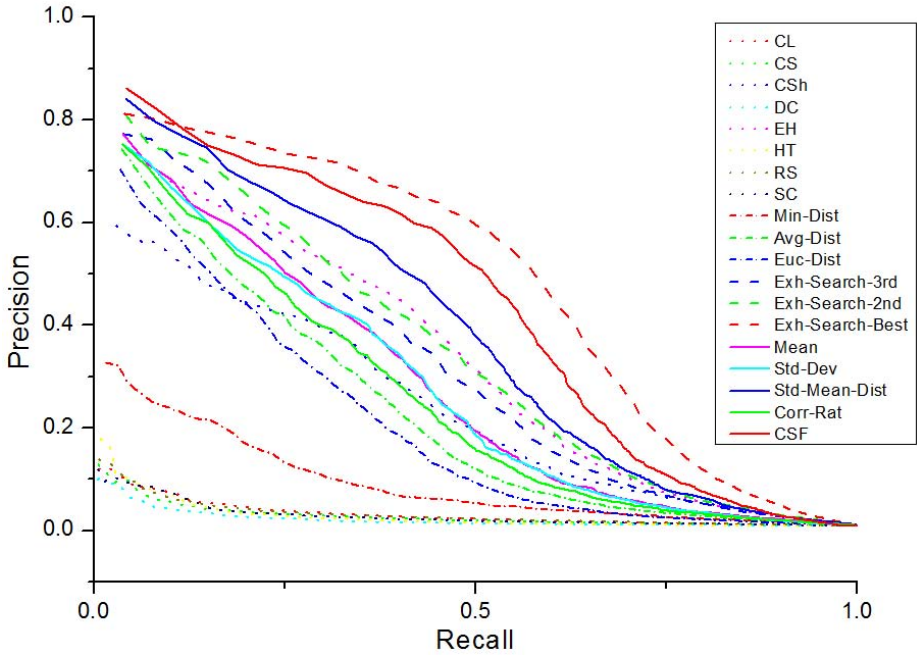


Fig. 1. Precision-Recall Graph for Semantic Retrieval

calculating an Euclidian distance on the available dissimilarities in ED. For the third step, feature selection by an exhaustive search approach is applied and the combined dissimilarity is calculated by averaging the dissimilarities of resultant features from the feature selection. An exhaustive search for feature selection requires calculating all combinations of available features, 2^8 cases in total for our test. Considering that performing an exhaustive search during each query is not applicable due to the time cost, the selection process is executed once on the prototype images. Then, 10 best selections (ES[1-10]) are found and semantic retrieval test is performed for each of these 10 feature selections. As the last step, the approach proposed in this study is performed for feature selection. Calculated CSF indices are used to combine the dissimilarity values with a weighted-sum approach. Not only the CSF index, but also the four parameters of the CSF are tested separately in order to see which one is more influential. In Figure 1, the Precision-Recall graphs of these methods are given. In addition, the AP of some sample categories, MAP of Best 10, 20, 50 and all 101 categories are presented in Table 1. Also, how many times each method has the best score and mean ranks of each method are included in the table.

Considering the test results, it is observed that obtaining an increase in the accuracy requires a good selection on the features. Simple methods like MD, AD and ED are not enough for selection. MD lacks the advantages of combining multiple features whereas AD and ED always combine all of the features and are affected by the unfavorable features. Besides, the exhaustive search

Table 1. Semantic Query Results

	electric guitar	saxophone	inline skate	stop sign	revolver	Best-10	Best-20	Best-50	All-101	Number of Best Scores	Mean Rank	
Single Features	CL	0.013	0.035	0.127	0.376	0.023	0.406	0.259	0.129	0.073	0	22.6
	CS	0.028	0.135	0.037	0.388	0.021	0.361	0.241	0.119	0.066	0	23.0
	CSh	0.865	0.766	0.725	0.253	0.361	0.841	0.743	0.542	0.339	3	13.9
	DC	0.020	0.033	0.055	0.427	0.019	0.258	0.171	0.088	0.050	0	23.6
	EH	0.895	0.874	0.633	0.827	0.928	0.924	0.855	0.667	0.424	6	10.1
	HT	0.006	0.063	0.159	0.235	0.029	0.304	0.210	0.112	0.063	0	23.0
	RS	0.097	0.120	0.153	0.624	0.114	0.354	0.233	0.121	0.070	0	22.4
	SC	0.016	0.065	0.057	0.654	0.064	0.352	0.229	0.116	0.066	1	22.8
Simple	MD	0.401	0.253	0.190	0.253	0.733	0.703	0.550	0.318	0.176	0	19.1
	AD	0.029	0.614	0.734	0.863	0.615	0.813	0.716	0.493	0.310	1	14.2
	ED	0.014	0.563	0.704	0.792	0.542	0.766	0.677	0.457	0.284	0	15.9
Exh. Search	ES1	0.958	0.964	0.870	0.856	0.970	0.963	0.927	0.806	0.563	36	5.0
	ES2	0.841	0.919	0.763	0.917	0.830	0.895	0.820	0.630	0.418	3	9.6
	ES3	0.565	0.951	0.831	0.985	0.898	0.923	0.828	0.616	0.400	1	11.1
	ES4	0.928	0.960	0.806	0.880	0.797	0.923	0.862	0.693	0.459	5	8.0
	ES5	0.934	0.916	0.844	0.910	0.973	0.927	0.872	0.704	0.484	7	6.9
	ES6	0.815	0.885	0.720	0.811	0.794	0.867	0.780	0.609	0.405	4	10.5
	ES7	0.641	0.968	0.911	0.981	0.959	0.932	0.855	0.663	0.441	6	8.7
	ES8	0.587	0.916	0.785	0.935	0.854	0.896	0.797	0.594	0.387	2	11.3
	ES9	0.578	0.972	0.844	0.979	0.852	0.927	0.845	0.634	0.409	3	10.6
	ES10	0.746	0.942	0.886	0.981	0.841	0.926	0.864	0.714	0.482	8	7.1
Proposed	μ	0.583	0.700	0.762	0.893	0.653	0.834	0.747	0.556	0.359	2	12.3
	σ	0.174	0.878	0.786	0.959	0.803	0.876	0.787	0.567	0.362	1	11.5
	δ	0.867	0.887	0.835	0.961	0.959	0.942	0.866	0.683	0.458	4	7.4
	ε	0.315	0.617	0.734	0.862	0.640	0.817	0.722	0.518	0.333	1	13.1
	CSF	0.955	0.981	0.889	0.987	0.957	0.970	0.928	0.769	0.521	24	5.5

guarantees to find the optimal feature selection by evaluating all possible combinations. Therefore ES1 outperforms the other methods. However the ES[1-10] ranking obtained at the training phase is not the same during the querying. For instance, ES5 performs better than ES2, ES3 and ES4. Such situation is caused by difference between training and query images. Although it is not observed in this test conditions, it could be possible that the best combination obtained during the training phase do not give best results during querying. It is possible to handle such incompliance by executing the exhaustive search during each query, but it causes a time inefficiency.

On the other hand, our proposed method of CSF gives successful accuracy results that are very close to the best selection in total and even better for one fourth of the image classes. Regarding that the results of the best selection in ES can be considered as the upper-bound for the retrieval task, the CSF method can be qualified as a robust and successful approach. In addition, our claim of exploiting class-specific features can be supported by the results of ES method. Different feature combinations in ES selections perform better in different image classes, which results different classes requires the use of different features.

Another observation on the results is the superiority of δ parameter of CSF approach among other parameters. Therefore, it can be stated that the discriminativeness characteristics of features are more effective than the representativeness.

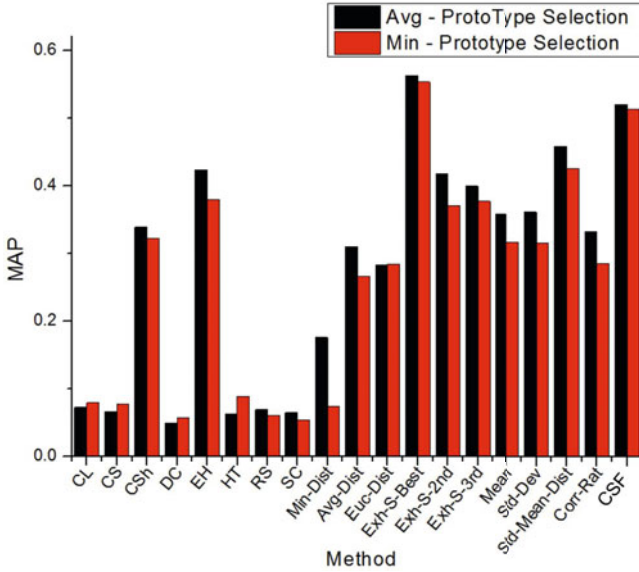


Fig. 2. Comparison of Average and Minimum Aggregation Methods

An important discussion for combining multiple features is the independency of features. Using complementary features with the methods requiring independent inputs can cause a decrease in the accuracies. Therefore, many studies exist in the information fusion literature that performs an independence analysis [16]. In this empirical study, the features utilized are not fully independent. It is previously stated that simple methods like MD, AD and ED are not successful enough for the selection task. One important reason in their inefficiency is the fact that they can not eliminate complementary information and the violation of independence assumption decreases their performance. However, the ES and CSF approaches enables selecting different combinations and eliminates complementary features.

As mentioned in Section 2, a prototype aggregation is necessary to combine the dissimilarities of multiple prototypes. Although prototype aggregation is beyond our scope, a secondary test is performed to show the effect of prototype aggregation. During the first test, *averaging* is used for aggregation. In this test, the previous test is repeated with a *minimum* aggregation method. The comparison of two methods are given in Figure 2. It is clearly shown that *averaging* is superior than *minimum*. However, these two are very simplistic methods and there are better ways of exploiting the information included in the prototypes.

As the last test, the time complexities of our proposed method and exhaustive search are compared. The query execution times of these two approaches are quite the same since querying includes only a weighted/unweighted summation of several features. However, the execution times for the training phases, which are carried out in order to find out the optimal set of features, differ much. Time

complexity of exhaustive search is $O(m^2 \cdot 2^n)$ where m is the total number of prototype images and n is the number of features. Whereas, time complexity of our proposed method is $O(m^2 \cdot n)$. Time-measurements obtained in this test validated these theoretical definitions. Results are given in Table 2. The results show us that CSF approach is 50 times better than the ES approach, in our case. If the number of features increases, execution time for ES could be worse.

Table 2. Execution Times for Training Phases

	Total Execution Time
Exhaustive Search	1,049,652 msec
CSF Calculation	19,802 msec

5 Conclusion

In this paper, a class-specific feature selection approach for the fusion of multiple features is presented. In order to eliminate the high-dimensionality of multiple features and provide efficient querying over the images, a dissimilarity based approach is utilized. The class-specific features are determined by using the representativeness and discriminativeness of features for each image class. The calculations of representativeness and discriminativeness are based on the statistics on the dissimilarity values of training images. The approach is tested on Cal-Tech 101 dataset by using 8 MPEG-7 features and compared with the single features, simple combination approaches and exhaustive search approach. Test results showed that proposed class-specific feature selection approach is a timely-efficient, accurate and robust way of feature selection.

Some further research direction can be as follows: Employing prototype selection and aggregation methods within the proposed approach, utilizing proposed approach with a dissimilarity based classification mechanism and performing multi-modal feature selection obtained from video data.

References

1. Arevalillo-Herráez, M., Domingo, J., Ferri, F.J.: Combining similarity measures in content-based image retrieval. *Pattern Recogn. Lett.* 29, 2174–2181 (2008)
2. Atrey, P.K., Kankanhalli, M.S., Oommen, J.B.: Goal-oriented optimal subset selection of correlated multimedia streams. *ACM Trans. Multimedia Comput. Commun. Appl.* 3 (February 2007)
3. Bruno, E., Marchand-Maillet, S.: Multimodal preference aggregation for multimedia information retrieval. *Journal of Multimedia* 4 (5), 321–329 (2009)
4. Bruno, E., Moënné-Loccoz, N., Marchand-Maillet, S.: Design of multimodal dissimilarity spaces for retrieval of multimedia documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(9), 1520–1533 (2008)
5. Chibelushi, C., Deravi, F., Mason, J.: Adaptive classifier integration for robust pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 29(6), 902–907 (1999)

6. Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: approaches and trends of the new age. In: MIR 2005: Proc. of the 7th ACM SIGMM Workshop on Multimedia Information Retrieval, pp. 253–262. ACM Press, New York (2005)
7. Duin, R.P.W., Loog, M., Pekalska, E., Tax, D.M.J.: Feature-based dissimilarity space classification. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 46–55. Springer, Heidelberg (2010)
8. Duin, R.P.W., Tax, D.M.J.: Experiments with classifier combining rules. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 16–29. Springer, Heidelberg (2000)
9. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(4), 594 (2006)
10. Jain, A., Ross, A.: Learning user-specific parameters in a multibiometric system. In: Proceedings of 2002 International Conference on Image Processing (2002)
11. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. *Pattern Recognition* 38(12), 2270–2285 (2005)
12. Jain, A.K., Duin, R.P., Mao, J.: Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 4–37 (2000)
13. Kankanhalli, M., Wang, J.: Experiential sampling on multiple data streams. *IEEE Transactions on Multimedia* 8(5), 947–955 (2006)
14. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 226–239 (1998)
15. Kludas, J., Bruno, E., Marchand-Maillet, S.: Information fusion in multimedia information retrieval. In: Boujemaa, N., Detyniecki, M., Nürnberger, A. (eds.) AMR 2007. LNCS, vol. 4918, pp. 147–159. Springer, Heidelberg (2008)
16. Kludas, J., Bruno, E., Marchand-Maillet, S.: Can feature information interaction help for information fusion in multimedia problems? *Multimedia Tools Appl.* 42, 57–71 (2009)
17. Kuncheva, L.I.: Switching between selection and fusion in combining classifiers: an experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 32(2), 146–156 (2002)
18. Martínez, J.: Mpeg-7 overview (version 10). Requirements ISO/IEC JTC1 /SC29 /WG11 N6828, International Organisation For Standardisation (October 2003)
19. MPEG: Mpeg-7 reference software experimentation model (2003), [http://standards.iso.org/ittf/PubliclyAvailableStandards/c035364_ISO_IEC_15938-6\(E\)_Reference_Software.zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c035364_ISO_IEC_15938-6(E)_Reference_Software.zip) (Online; accessed February 01, 2011)
20. Nguyen, G.P., Worring, M., Smeulders, A.W.M.: Similarity learning via dissimilarity space in cbir. In: Proc. of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR 2006, pp. 107–116. ACM, New York (2006)
21. Pekalska, E., Paclík, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research* 2, 175–211 (2001)
22. Poh, N., Kittler, J.: Multimodal Information Fusion: Theory and Applications for Human-Computer Interaction. 8, pp. 153–169. Academic Press, London (2010)
23. Snidaro, L., Niu, R., Foresti, G., Varshney, P.: Quality-based fusion of multiple video sensors for video surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37(4), 1044–1051 (2007)
24. Swets, D.L., Weng, J.J.: Shoslif-o: Shoslif for object recognition and image retrieval (phase ii). Tech. Rep. CPS 95-39, Michigan State University, Department of Computer Science (1995)

25. Uysal, M., Yarman-Vural, F.T.: Selection of the best representative feature and membership assignment for content-based fuzzy image database. In: CIVR, pp. 141–151 (2003)
26. Wu, Y., Chang, E.Y., Chang, K.C.C., Smith, J.R.: Optimal multimodal fusion for multimedia data analysis. In: Proc. of the 12th Annual ACM International Conf. on Multimedia, MULTIMEDIA 2004, pp. 572–579. ACM, New York (2004)
27. Xu, L., Krzyzak, A., Suen, C.: Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics* 22(3), 418–435 (1992)

Generalised Fuzzy Types and Querying: Implementation within the Hibernate Framework

Jose Enrique Pons, Ignacio Blanco Medina, and Olga Pons Capote

Department of Computer Science and Artificial Intelligence
University of Granada

C/Periodista Daniel Saucedo Aranda s/n E-18071, Granada-Spain
{jpons,opc}@decsai.ugr.es

Abstract. Fuzzy databases manage imprecision in its schema and offer tools for flexible querying. A true standard does not exist. Relational databases are usually the base for the implementation of fuzzy databases. In this paper we propose a general model to represent and querying fuzzy types in any relational database. The model is implemented within the Hibernate Framework.

Keywords: fuzzy database, fuzzy querying, open source.

1 Introduction

A fuzzy database (*FDB*) is a database that manages imprecision in its schema by means of fuzzy sets. Almost every implementation is an extension to some existent relational database management systems (*DBMS*). The portability is the main problem for these implementations. Our goal is to define and implement a model that manages imprecision (both representation and querying) in any underlying relational database. The model selected for the representation is GEFRED [10]. This model is a synthesis among different proposals [3, 12, 15, 17] to deal with the problem of the representation and management of fuzzy information on relational databases. Several proposals for fuzzy querying relational databases can be found in [13, 2].

The model that we propose is inspired on the ability of the Hibernate Framework to abstract the applications from the specific database vendors. Therefore, any relational DBMS may be extended into a fuzzy database with fuzzy types and flexible querying.

The structure of the paper is organized as follows: in section 2 some general concepts about fuzzy databases and a short introduction for Hibernate framework are both introduced. Section 3 is the proposal and the implementation of the general model for the fuzzy types representation and querying. A sample of use with the results is explained in section 4. In section 5, we compare the proposed implementation with respect to other implementations. Finally, section 6 contains the conclusions and the future research work.

2 Context

The main characteristics of a FDB are explained in section 2.1. We will make a short introduction on the Generalized model of Fuzzy Relational Databases (*GEFRED*) [10]. Section 2.2 describes the main features of Hibernate and its architecture. We will explain and compare the different querying methods provided by the framework.

2.1 Fuzzy Relational Databases

As mentioned above, a fuzzy database is a database that manages vagueness or imprecision in its schema. The model selected for the representation is *GEFRED* [10]. This main model takes into account the most important proposals, like Kacprzyk [7] and Buckles models [4], [3]. The relational representation is defined in the Fuzzy Interface for Relational SysTEms (*FIRST*) [5], [6].

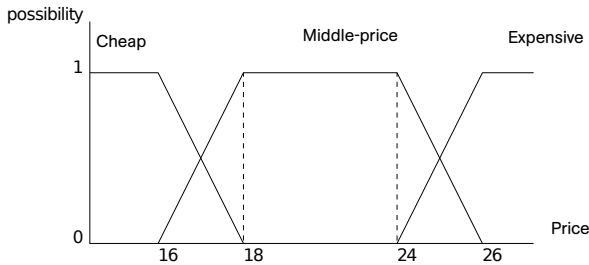


Fig. 1. Price possibility distributions. Labels for 'Cheap', 'Middle-price' and 'Expensive' are associated to each possibility distribution

Fuzzy representation and flexible querying are the two main features that a FDB should provide. The following points explain these properties.

Fuzzy Datatypes. The *GEFRED* model defines three main types of fuzzy data types. Each fuzzy data type has its own representation and restrictions. The model defines the well known fuzzy constants UNKNOWN, UNDEFINED and NULL. The datatypes are:

1. Data type 1: Crisp data with fuzzy querying capabilities. The underlying domain for type 1 is a numerical type. E.g. The distance between two cities.
2. Data type 2: Fuzzy data types over ordered underlying domains (numerical domains). These types represent possibility distributions of several forms (triangular, trapezoidal and interval). Table 1 shows the relational representation in 5 columns. The following are the main sub-types (field FT in table 1):
 - (a) *CRISP*: The field stores a crisp value.
 - (b) *LABEL*: A label is a string linked to a fuzzy value (or a possibility distribution). In example, in figure 1 a label 'middle-price' is linked with the second possibility distribution.

Table 1. Relational representation for fuzzy attributes type 2. Note that '-' is the abbreviation for *NULL* constant.

Fuzzy Type	FT	F1	F2	F3	F4
UNKNOWN	0	-	-	-	-
UNDEFINED	1	-	-	-	-
NULL	2	-	-	-	-
CRISP	3	d	-	-	-
LABEL	4	ID	-	-	-
INTERVAL	5	n	-	-	m
APPROX	6	d	d-m	d+m	m
TRAPEZ	7	α	β	γ	δ

- (c) *INTERVAL*: Two values represent a fuzzy interval, usually written as $[n, m]$.
- (d) *APPROXIMATE*: A crisp value (written as d) is stored with a *margin* (m) value. Therefore, a triangular possibility distribution is built with the core equal to d and the support equals to the interval $[d - m, d + m]$.
- (e) *TRAPEZOIDAL*: The tuple $[\alpha, \beta, \gamma, \delta]$ represents a trapezoidal possibility distribution. The support is the interval $[\alpha, \delta]$ and the core is the interval $[\beta, \gamma]$.

Table 2. Relational representation for fuzzy attributes type 3. Note that '-' is the abbreviation for *NULL* constant.

Fuzzy Type	FT	FP1	F1	...	FPn	Fn
UNKNOWN	0	-	-	...	-	-
UNDEFINED	1	-	-	...	-	-
NULL	2	-	-	...	-	-
SIMPLE	3	p	d	...	-	-
POSS.DIST	4	p_1	d_1	...	p_n	d_n

3. Data type 3: Fuzzy data with an underlying non-ordered domain. The user must define each domain value for this data type. A label identifies each value. Finally, it is necessary to define a similarity relationship between each pair of values. The only applicable operator is the fuzzy equality operator because of the underlying domain is non-ordered. Table 2 shows the representation in a relational database. This datatype supports two main sub-types:
 - (a) *SIMPLE*: This type needs two values: A label identifier (stored as F1) and the possibility degree for this label, stored as FP1.
 - (b) *POSSIBILITY DISTRIBUTION*: This type needs a list of pairs of values. The first value (stored as Fn) stores the label identified. The second value stores the possibility degree for the mentioned label (stored as FPn). E.g. a possibility distribution for the hair colour. $\{0.75/\text{Brown}, 0.2/\text{Blond}, 0.01/\text{Red}\}$.

Table 3. Fuzzy operators

Possibility	Necessity	Possibly / Necessarily
FEQ	NFEQ	Fuzzy =
FGT	NFGT	Fuzzy >
FGEQ	NFGEQ	Fuzzy ≥
FLT	NFLT	Fuzzy <
FLEQ	NFLEQ	Fuzzy ≤
MGT	NMGT	Much >
MLT	NMLT	Much <

Fuzzy Querying. Fuzzy operators perform comparisons by means of possibility distributions. Table 3 shows the operators defined in *FIRST*. It is also possible to compare values with fuzzy constants. Each operator implements comparison between the possibility distributions represented by the fuzzy datatypes. E.g., the fuzzy equals operator is implemented as follows:

$$FEQ(p1, p2) = \sup_{d \in U} \min(\pi_{p1}(d), \pi_{p2}(d)) \tag{1}$$

Where U is the underlying domain, $p1, p2$ are two values of a given fuzzy type (e.g. fuzzy type 2) and $\pi_{p1}(d), \pi_{p2}(d)$ are the associated possibility distributions.

2.2 Hibernate Framework

The Hibernate Framework [8] is a collection of open source projects that enable developers to make object-relational mapping. The framework needs an object-oriented language (Java) and a relational DBMS. A query language called Hibernate Query Language (*HQL*) is also provided, which is an object-oriented extension to SQL.

Architecture. An application working with Hibernate has 3 layers (fig. 2):

1. **Application layer:** It is the top layer. The application makes CRUD (Create, Update and Delete) operations by means of persistent objects: an object in the application representing a table in the database. There exist two types of persistent objects:

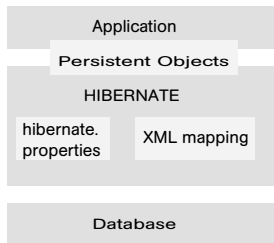


Fig. 2. Minimal Hibernate architecture

- (a) **Persistent Objects:** These objects represent a current database state: the object state joins the database state by means of the database session.
 - (b) **Transient Objects:** These objects are no longer attached to a database session. Thus, their value does not represent a current database state.
2. **Hibernate layer:** This layer acts as an abstraction layer between the DBMS and the application. The framework introduces the concept of *dialect*: A dialect is an abstraction for the specific DBMS. Thus, the application will work with any DBMS by changing the dialect.
 3. **The database:** The lower layer is the DBMS: MySQL, PostgreSQL, Oracle and many other DBMS supported by the framework.

Querying in the Hibernate Framework. Four ways are defined for querying in the Hibernate framework. Ordered from the more object-oriented way to the more relational way:

1. Query by criteria: Selects objects that fulfill a set of criterion. E.g.

```
createCriteria(Person.class).
Add(Restrictions.eq("login", "lmmn"));
```

2. Query by example: Selects objects similar to a given object. E.g.

```
Person p.login = "lmmn";
```

3. HQL: An object-oriented language based on SQL. E.g.

```
Select p from Personal p where p.login = "lmmn";
```

4. SQL: A SQL sentence. E.g.

```
Select * from Person as p where p.login = 'lmmn';
```

3 Proposal

In this section we will define the architecture of a general framework for fuzzy representation [3.1](#) and fuzzy querying [3.2](#) in any relational database. The implementation of the model is done within the Hibernate framework.

3.1 Fuzzy Representation

To represent a fuzzy type in any relational DBMS, we will analyze the constraints that the classical approach has. Then, the architecture of the general model will be presented and finally, the implementation within Hibernate.

Conditions for portability: To represent a (fuzzy) type in any relational database, the following assumptions should hold:

1. The interface between the framework and the database is the SQL standard.
2. The extended types (fuzzy types) are represented in the database with basic SQL types.
3. The meta-data for managing properly the extended types relies outside the database catalog. This means that outside the framework, the (fuzzy) datatypes will not be treated properly.
4. Each DBMS has its own implementation for the SQL standard. This implies that for each DBMS an interface between the framework and the database is needed.

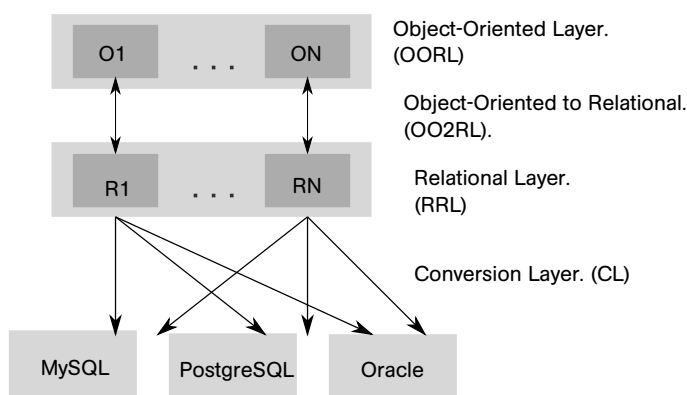


Fig. 3. Abstract layer model

Architecture for the General Model. The architecture for a generalized object-oriented model needs the following elements (fig. 3):

- **Object-Oriented Representation Layer.** *OORL*: The representation of the (fuzzy) types in an object-oriented way.
- **Object-Oriented to Relational Representation Layer.** *OO2RRL*: This layer is a mapping between the objects in the higher layer and the representation in the layer below. Note that the mapping may not be trivial and a conversion function should be given.
- **Relational Representation Layer.** *RRL*: The (fuzzy) types are represented by basic types. In the GEFRED model, *FIRST* is the specification for this layer.
- **Conversion Layer.** *CL*: This layer customizes the SQL representation from *RRL* for the concrete database implementation. This customization process must be done because of the different implementations of the SQL standard on each DBMS.

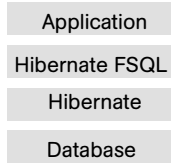


Fig. 4. Minimal Hibernate architecture for fuzzy representation

Implementation. The proposed general framework relies between the application and the DBMS. The implementation of the model in the Hibernate Framework is the following (see fig. 4):

- **FSQL layer:** Is the implementation for OORL: the representation for the fuzzy objects in the Java programming language.
- **Transformation layer:** This layer bring together the implementation of the OO2RRL and RRL layers.
- **Conversion Layer:** Hibernate supports this layer by means of the dialect. It has mapping types between SQL and the concrete implementation for these types in the database. Thus, each DBMS, has its own dialect. Hibernate provides dialects for the major DBMS, and it is easy to develop new dialects for new DBMS.

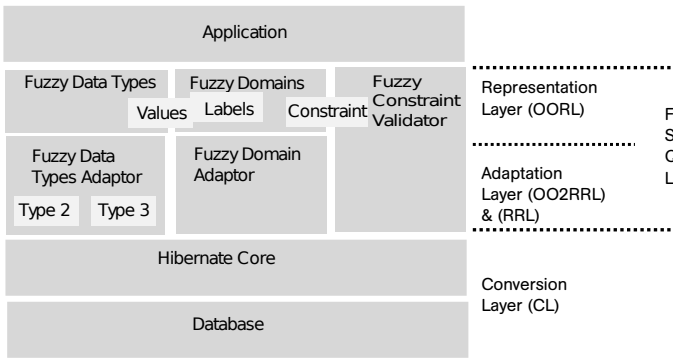


Fig. 5. Detailed Hibernate architecture for fuzzy representation

In a more detailed view (fig. 5), the *FSQL layer* is composed by several sub-elements:

1. **Representation:**
 - (a) **Fuzzy data types:** The three fuzzy data types mentioned above are represented. This representation is based on Fuzzy Knowledge Representation Ontology (*FKRO*) [9] since it suits our object-oriented representation to a large extent.

- (b) **Fuzzy domains:** To create fuzzy domains of types 2 and 3, two main fuzzy meta-domains are defined.
 - (c) **Fuzzy constraint validator:** Each fuzzy domain may be associated with a set of fuzzy constraints. The validator checks the constraints.
2. **Adaptation layer:** This layer transforms from the object-oriented representation to the relational representation in the database (see section 2.1).

3.2 Fuzzy Querying

To generalize fuzzy querying into any relational database, we should take into account that the interface between the relational DBMS and the framework is the SQL standard. Therefore, the model we propose uses the following elements:

- **Declarative implementation** for each fuzzy operator. This implementation should be done in the SQL language.
- **Abstract syntax tree (AST) representation** for the query. These representation allows a customization process done by the conversion layer (*CL*).
- **Conversion Layer:** This layer customizes the *AST* for the running database.

The implementation of the fuzzy querying is done by modifying the HQL language. The fuzzy operators are implemented in a declarative way in the SQL language. The HQL language has the following features:

- **Object-oriented representation** for queries.
- **Customization:** The HQL code is analyzed and translated into SQL sentences. Hibernate customizes SQL sentences for the running DBMS through the dialect. E.g. The application is running against MySQL, then Hibernate generates the SQL code customized for it.

The Hibernate core deals with the HQL translation. By modifying this code, implementing fuzzy operators should result in a HQL with fuzzy querying capabilities. This extension will work on any database supported by the system also. The following steps are how Hibernate processes a HQL query:

- The framework builds an abstract syntax tree (AST) once the query passed lexical and syntactical analysis.
- The AST represents tokens as nodes. The semantic analyzer renders the tree in the into SQL sentences. Then the dialect customizes the SQL sentence.

The following example explains the translation workflow. It is a HQL sentence that search for all the restaurants that have an average price around 15 euro:

```
SELECT r FROM Restaurant r WHERE r.PriceAvg FEQ $[15,10,20,5];
```

Figure 6 shows the AST tree for this sentence. Then, the FEQ node is mutated in the rendering process to its implementation in SQL. Through the dialect, the sentence is customized to execute for instance, in MySQL.

Consider the following two triangular possibility distributions notated by $(\alpha_2, \beta_2, \gamma_2, \delta_2)$ for the user input and $(priceAvg_\alpha, priceAvg_\beta, priceAvg_\gamma, priceAvg_\delta)$ for the *priceAvg* field (see section 2.1). The implementation in SQL is the following:

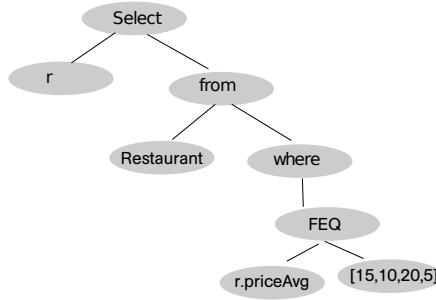


Fig. 6. Abstract Syntax Tree for the sentence

```

SELECT * FROM Restaurant as r WHERE
1 < CASE WHEN (r.priceAvg.gamma <= beta2)
    OR (r.priceAvg.beta >= gamma2) THEN 0
    WHEN (r.priceAvg.alpha = alpha2) THEN 1
    WHEN (r.priceAvg.gamma > beta2) AND (r.priceAvg.alpha < alpha2)
    THEN (r.priceAvg.gamma - beta2) / (r.priceAvg.delta - delta2)
    ELSE (gamma2 - r.priceAvg.beta) / (r.priceAvg.delta + delta2);
    
```

4 A Practical Case of Use: Selection of Customer’s Preferences

Consider a touristic recommendation system that works with restaurants. The each restaurant is stored as in table 4. The field ID is the primary key of the table and the unique identifier for the restaurant. The average price is stored as a fuzzy triangular distribution $(d, d + m, d - m, m)$. (see section 2.1).

Table 4. Restaurant table, with the average price expressed as triangular possibility distribution

ID	Name	AVG	Quality
001	Amadeus	[15,10,20,5]	5
002	Atlantis	[11,5,17,6]	3
003	Cafe Theatre	[20,10,30,10]	2
004	De Graslei	[23,18,28,5]	5
005	Pakhuis	[13,12,15,2]	3
006	De 3 Biggetjes	[45,25,65,20]	4

The user wants to obtain information about the restaurants in the city which have an average price around 15 euro (with a margin of 5 euro). The query in the HQL language uses the fuzzy equality FEQ operator and compares the triangular possibility distribution stored in the field Avg w.r.t. the triangular possibility distribution specified by the user:

SELECT R from Restaurant R WHERE R.AVG FEQ #[15, 10, 20, 5];

This query is translated as explained in section 3.2. Table 1 shows the resultset for the query and the compatibility degrees for each restaurant.

Table 5. Resultset of the query

Compatibility	ID	Name	AVG	Quality
1	001	Amadeus	[15,10,20,5]	5
0.6363	002	Atlantis	[11,5,17,6]	3
0.6667	003	Cafe Theatre	[20,10,30,10]	2
0.2	004	De Graslei	[23,18,28,5]	5
0.714	005	Pakhuis	[13,11,15,2]	3
0	006	De 3 Biggetjes	[45,25,65,20]	4

5 Comparison

In this section we will discuss the differences with respect to the portability among several implementations. The proposals analyzed are:

- FSQL server [5]: The reference implementation of the FIRST interface on the GEFRED model. The first implementation works with Oracle database, although there is an implementation in PostgreSQL.
- SQLfi [14]: The implementation for the SQLf language [21].

Table 6. Comparison among different fuzzy DB implementations

Fuzzy DB	Catalog	Interface	Query language	Query Processor
FSQL	Inside DB.	FSQL client	FSQL	Procedural.
SQLfi	Inside DB.	Client app.	SQLf	Procedural.
FDBLL	Inside DB.	Client app.	Fuzzy SQL	Procedural.
PFSQL	Inside DB.	JDBC client	PFSQL	Procedural.
H. FSQL	Outside DB.	Entity Classes	Fuzzy HQL	Declarative.

Table 7. Changes to migrate the implementation to another DBMS

Fuzzy DB	Changes for representation	Changes for querying
FSQL	Create the metadata tables (FMB). Develop a client application.	Implement in a procedural way the FSQL operators. Develop a client application.
SQLfi	Create the metadata tables. Develop a client application.	Implement the fuzzy operators and the query translator.
FDBLL	Create the fuzzy data definitions.	Implement the fuzzy SQL processor in the DBMS.
PFSQL	Create the metadata tables.	No changes
H. FSQL	No changes.	No changes.

- FDBLL [11]: Fuzzy database language and library . A fuzzy SQL implementation in C language over a relational DBMS.
- PSQL [16]: An extension of the FSQL model. The main features are the use of priority fuzzy logic and the portability: The implementation is done by a JDBC driver. The driver acts as interface between any Java program and the fuzzy database. The user may change the running database by just adding the fuzzy meta tables to the catalog and keeping the same program.
- Hibernate FSQL: The proposed implementation. The main difference is that the fuzzy meta data is not stored in the database, therefore, to change the running database is as easy as changing some parameters in the Hibernate configuration file. There is no need to create or modify fuzzy meta tables in the DBMS catalog.

Table 6 shows the main differences in the implementations and in the portability among each approach. Table 7 shows the changes that must be done in order to change the running DBMS for an application.

6 Conclusions and Future Work

In the presented work we have introduced a general model for the representation of (fuzzy) types and for fuzzy querying. The main advantage with respect to other implementations is the portability, as have been discussed in last section. The drawback for the portability is the dependency between the application and the framework. This means that, outside the framework, the DBMS is not able to manage the fuzzy types nor to make fuzzy queries. This is not such a big issue. Over the last few years the trend is to develop the business layer outside the DBMS too. further research work will include new approaches for fuzzy querying like bipolarity specification in the queries and practical applications with the proposed framework.

Acknowledgements. The researchers are supported by the grant BES-2009-013805 within the research project TIN2008-02066: *Fuzzy Temporal Information treatment in relational DBMS*, and the project P07-TIC-03175: *Representation and Handling of Imperfect Objects in Data Integration Problems*.

References

1. Bosc, P., Galibourg, M., Hamon, G.: Fuzzy querying with sql: extensions and implementation aspects. *Fuzzy Sets Syst.* 28, 333–349 (1988)
2. Bosc, P., Pivert, O., Farquhar, K.: Integrating fuzzy queries into an existing database management system: An example. *International Journal of Intelligent Systems* 9(5), 475–492 (1994)
3. Buckles, B.P., Petry, F.E.: A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems* 7(3), 213 (1982)
4. Buckles, B.P., Petry, F.E.: Extending the fuzzy database with fuzzy numbers. *Information Sciences* 34(2), 145–155 (1984)

5. Galindo, J., Medina, J., Pons, O., Cubero, J.: A server for fuzzy SQL queries. In: Andreassen, T., Christiansen, H., Larsen, H.L. (eds.) FQAS 1998. LNCS (LNAI), vol. 1495, pp. 164–174. Springer, Heidelberg (1998)
6. Galindo, J., Urrutia, A., Piattini, M.: Fuzzy Databases: Modeling, Design and Implementation. Idea Group (2006)
7. Kacprzyk, J., Ziółkowski, A.: Database queries with fuzzy linguistic quantifiers. *IEEE Trans. Syst. Man Cybern.* 16, 474–479 (1986)
8. King, G., et al.: Hibernate Reference Documentation, 3.6.0.cr2 edn., <http://www.hibernate.org/docs>
9. Martínez Cruz, C.: Sistema de gestión de bases de datos relacionales difusas multipropósito. Ph.D. thesis, Universidad de Granada (2008)
10. Medina, J., Pons, O., Cubero, J.: a generalized model of fuzzy relational databases. *Information Sciences* 76(1-2), 87–109 (1994)
11. Nakajima, H., Sogoh, T., Arao, M.: Fuzzy database language and library-fuzzy extension to sql. *Fuzzy Systems* 1, 477–482 (1993)
12. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences* 34(2), 115–143 (1984)
13. Tahani, V.: A conceptual framework for fuzzy query processing - a step toward very intelligent database systems. *Inf. Process. Manage.* 13(5), 289–303 (1977)
14. Tineo, L., Goncalves, M., Eduardo, J.C.: A fuzzy querying system based on sqlf2 and sqlf3. In: Solar, M., Fernandez-Baca, D., Cuadros-Vargas, E. (eds.) CLEI 2004, pp. 845–851 (September 2004)
15. Umano, M., Fukami, S.: Retrieval processing from fuzzy databases. *Technical Reports of IECE of Japan* 80(204), 45–54 (1980)
16. Škrbic, S., Takaci, A.: An interpreter for priority fuzzy logic enriched sql. In: BCI 2009, pp. 96–100. IEEE Computer Society, Washington, DC, USA (2009)
17. Zemankova-Leech, M., Kandel, A.: Fuzzy relational databases - a key to expert systems (1984)

Data Reliability Assessment in a Data Warehouse Opened on the Web

Sébastien Destercke, Patrice Buche, and Brigitte Charnomordic

UMR IATE and MISTEA, 2 place Viala, F-34060 Montpellier Cedex 1, France
LIRMM, CNRS-UM2, F-34392 Montpellier, France
destercke@cirad.fr, {buche,bch}@supagro.inra.fr

Abstract. This paper presents an ontology-driven workflow that feeds and queries a data warehouse opened on the Web. Data are extracted from data tables in Web documents. As web documents are very heterogeneous in nature, a key issue in this workflow is the ability to assess the reliability of retrieved data. We first recall the main steps of our method to annotate and query Web data tables driven by a domain ontology. Then we propose an original method to assess Web data table reliability from a set of criteria by the means of evidence theory. Finally, we show how we extend the workflow to integrate the reliability assessment step.

1 Introduction

The huge amount of technical and scientific documents available on the Web include many data tables. In addition to local data sources, they represent big potential external data sources for the data warehouse of a company dedicated to a given domain of application. To lighten the burden laid upon domain experts when selecting data from the data warehouse for a particular application, it is necessary to give them indicative reliability evaluations. In this paper, we present a framework to estimate the reliability of data tables collected from the Web. Compared to more *ad-hoc* estimation, the presented generic method can give insights to the expert as to why a particular data table is tagged as reliable or not reliable. Due to its generic nature, this method can be reused in other data warehouses using the semantic web recommended languages.

Reliability estimation is an essential part of the Semantic Web architecture, and many research works [1] focus on issues such as source authentication, reputation, etc. For example, [2] advocates a multi-faceted approach to trust models. They propose an OWL based ontology of trust related concepts. The idea is to provide systems using the annotation power of a user community to collect information about reliability. Our approach is different, as we do not rely on users but rather on information about the Web data table origins to compute a reliability estimations. Among methods proposing solutions to evaluate trust or data quality in web applications, the method presented in [3] is close to the method presented in the paper. It uses possibility theory evidence theory, whereas we base our method on evidence theory. Another difference is that in our approach global information is obtained by a fusion of multiple uncertainty models, while in [3] global information results from the propagation of uncertainty models through a aggregation function. Each method has its pro and cons: it is easier to integrate interactions

between criteria in aggregation functions, while it is easier to retrieve explanations of the final result in our approach.

In this paper, we details our method and its integration in @Web, along with the whole workflow used in @Web. The current version of @Web (see [4,5]), a Web-enabled data warehouse, has been implemented using the W3C recommended languages (see [6] for details about these languages): OWL to represent the domain ontology, RDF to annotate Web tables and SPARQL to query annotated Web tables.

We first recall in Section 2 the purpose and architecture of the data warehouse. Section 3 details the proposed method to assess Web data table reliability. In Section 4, we show how this reliability assessment is presented and explained to the user. Finally, in Section 5, we explain how @Web is extended to implement the reliability management.

2 @Web Presentation

@Web is a data warehouse opened on the Web [4,5] centered (in its current version) on the integration of heterogeneous data tables extracted from Web documents. The focus has been put on Web tables for two reasons: (i) experimental data are often summarized in tables, (ii) table structured data are easier to integrate than, e.g., in text or plots. The main steps of Web table integration are summarised in Fig. 1. A central role in data integration in @Web is played by the domain ontology. This ontology describes the concepts, their relations and the associated terminology of a given application domain. @Web can therefore be instantiated for any application domains (e.g., food predictive microbiology, food chemical risks, aeronautics [5]), provided a proper domain ontology is defined.

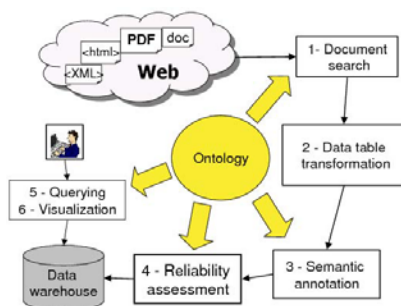


Fig. 1. Main steps of the document workflow in @Web

Once the ontology is built, @Web workflow includes the different steps shown in Fig. 1 to integrate new data in the warehouse. Concepts found in a data table and semantic relations linking these concepts are automatically identified. Data tables are then annotated with the identified concepts, allowing users to interrogate and query the data warehouse in an homogeneous way.

Our case study uses the @Web instance implemented in the Sym'Previus [7] decision support system whose aim is to simulate the growth of a pathogenic microorganism

in a food product. Semantic relations in this system include for instance the *GrowthRate* that links a given microorganism within a given food product to a specific growth rate and its associated parameters. Data retrieved from tables can then be used to define the parameters of numerical growth oriented simulated models.

2.1 @Web Generic Ontology

The current OWL ontology representation used in the @Web system is composed of two main parts: a generic part, called *core ontology*, which contains the structuring concepts of the Web table integration task, and a specific part, commonly called *domain ontology*, which contains the concepts specific to the considered domain. The *core ontology* is composed of symbolic concepts, numeric concepts and relations between these concepts. It is separated from the definition of the concepts and relations specific to a given domain, i.e., the *domain ontology*. All the ontology concepts are materialized by OWL classes. For example, in the microbiological ontology, the respectively symbolic and numeric concepts *Microorganism* and *pH* are represented by OWL classes, respectively subclass of the generic classes *SymbolicConcept* and *NumericConcept*. An excerpt of an OWL class organization for symbolic concepts is given in Figure 2.

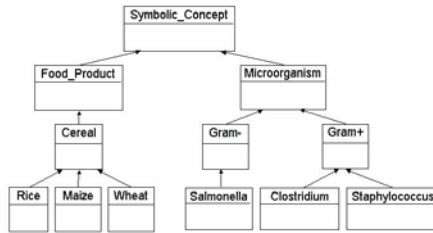


Fig. 2. Excerpt of OWL class hierarchy for symbolic concepts in the microbial domain

2.2 @Web Workflow

The first three steps of @Web workflow (see Fig. 1) are as follows. The first task consists in retrieving relevant Web documents for the application domain, using key-words extracted from the domain ontology. It does so by defining queries executed by different crawlers. In the second task, data tables are extracted from the retrieved documents and are semi-automatically translated into a generic XML format. The Web tables are then represented in a classical and generic way – i.e., a set of lines, each line being a set of cells. In the third task, the Web tables are semantically annotated according to the domain ontology. The semantic annotation process of a Web table consists in identifying which semantic relations of the domain ontology can be recognized in each row of the Web table (see [5] for details). This process generates RDF descriptions.

Example 1. Fig. 3 presents an example of a Web table in which the semantic relation *GrowthParameterAwMin* has been identified. The domain of this relation is a kind of Microorganism and its range is food product water activity (a_w). The first row indicates that *Clostridium* requires a minimal food product a_w of 0.943 to be able to grow.

Organism	a_w minimum	a_w optimum	a_w maximum
Clostridium	0.943	0.95-0.96	0.97
Staphylococcus	0.88	0.98	0.99
Salmonella	0.94	0.99	0.991

Fig. 3. Example of a Web table

Example 2. Figure 4 presents the main part of the RDF descriptions corresponding to the recognition of the relation *GrowthParameterAwMin* in the first row (denoted *uriRow1*) of the Web table given by Fig. 3. Starting from the left part of the figure, we see that the row is annotated by the relation *GrowthParameterAwMin*, abbreviated as *GPaw1*. The domain of the relation *GrowthParameterAwMin* is an instance of the symbolic concept *Clostridium*. The range of the relation is an instance of the numerical concept *Aw* and has for value 0.943.

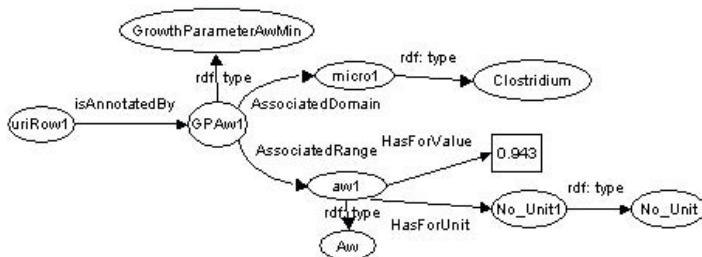


Fig. 4. Example of RDF annotations generated from the Web table of Figure 3

2.3 SPARQL Querying of RDF Graphs

In the XML/RDF data warehouse, the querying is done through MIEL++ queries. We briefly recall how MIEL++ queries are executed in the current version of @Web (For details, see [4]). A MIEL++ query is asked in a view that corresponds to a relation of the ontology (e.g., the relation *GrowthParameterAwMin*). A MIEL++ query is an instantiation of a view by the end-user, who specify among the set of queryable attributes of the view what are the selection attributes and their searched values, and what are the projection attributes (with the meaning of the relational model). An important specificity of a MIEL++ query is that searched values may be expressed as fuzzy sets (see [8,9,10]), which use allows end-users to represent their preferences in a gradual way.

Definition 1. A fuzzy set μ defined on a space \mathcal{A} is a function $\mu : \mathcal{A} \rightarrow [0, 1]$ with $\mu(x)$ the membership degree of x . The support $S(\mu)$ and the kernel $K(\mu)$ of a fuzzy sets are the sets $S(\mu) = \{x \in \mathcal{A} | \mu(x) > 0\}$ and $K(\mu) = \{x \in \mathcal{A} | \mu(x) = 1\}$.

Example 3. Let us define a MIEL++ query Q expressed in the view *GrowthParameterAwMin* as follows:

$$Q = \{ \text{Microorganism, aw} \mid (\text{GrowthParameterAwMin}(\text{Microorganism, aw}) \wedge (\text{Microorganism} \approx \text{MicroPreferences}) \wedge (\text{aw} \approx \text{awPreferences})) \}.$$

The discrete fuzzy set *MicroPreferences*, which is equal to $\{(Gram+,1.0), (Gram-,0.5)\}$, means that the end-user is firstly interested in microorganisms which are Gram+ and secondly Gram-. The trapezoidal fuzzy set *awPreferences* that has the characteristic points $[0.9, 0.94, 0.97, 0.99]$, means that the end-user is first interested in *aw* values in the interval $[0.94, 0.97]$ (the kernel of the fuzzy set), but that he/she accepts to enlarge the querying till the interval $[0.9, 0.99]$ (the support of the fuzzy set).

Since fuzzy sets are not supported in a standard SPARQL query, a complete solution to translate a MIEL++ query into a standard SPARQL query is presented in detail in [4]. In this paper, we only recall how is measured the satisfaction of a MIEL++ query. The satisfaction of a selection criterion $att \approx attPref$ is measured by the membership degree $\mu_{attPref}(x)$ of the corresponding value x expressed in the RDF graph (x is supposed to be a crisp value in this paper). As selection criteria are considered to be conjunctive, a global adequation degree, denoted ad , is computed using the t-norm *min*.

Example 4. The answers to the SPARQL query associated with the MIEL++ query of Example 3 compared with the Web table presented in Figure 3 is given below:

ad	$\mu_{MicroPref}(x)$	$\mu_{AwPref}(x)$	<i>Microorg</i>	<i>aw</i>
1.0	1.0	1.0	<i>Clostridium</i>	0.943
0.5	0.5	1.0	<i>Salmonella</i>	0.94
0.0	1.0	0.0	<i>Staphilococcus</i>	0.88

3 Reliability Evaluation

This section describes the method we propose to evaluate the reliability of Web tables.

3.1 A Model for Reliability Evaluation

We assume that reliability takes its value on a finite ordered space $\Theta = \theta_1, \dots, \theta_N$ such that $\theta_i < \theta_j$ iff $i < j$. θ_1 corresponds to total unreliability, while θ_N corresponds to total reliability. We denote by $I_{a,b} = \{\theta_a, \dots, \theta_b\}$ a set such that $a \leq b$ and $\forall c$ s.t. $a \leq c \leq b$, $\theta_c \in I_{a,b}$. Such sets include all values between their minimum value θ_a and maximum value θ_b , and using a slight stretch of language we call them *intervals*.

The evaluation will be based on the values taken by S groups A_1, \dots, A_S of criteria. Note that a group may be composed of multiple criteria, e.g., number of citation \times publication date. The group constitution ensures that the impact of each group A_i on the reliability evaluation can be judged (almost) independent of the impact of any other group A_j . Each group A_i can assume C_i distinct values on spaces $\mathcal{A}_i = \{a_{i1}, \dots, a_{iC_i}\}$.

For each possible value of each criteria group A_1, \dots, A_S , a domain expert is asked to give its opinion about the corresponding data reliability. To facilitate expert elicitation, a linguistic scale with a reasonable number of terms is used, for instance the five terms *very unreliable*, *slightly unreliable*, *neutral*, *slightly reliable* and *very reliable*. These opinions are modeled as fuzzy sets that describes some ill-known value of reliability.

Denote by $\mathcal{F}(\Theta)$ the set of all fuzzy sets defined over a domain Θ . For each group A_i , we define a mapping $\Gamma_{A_i} : \mathcal{A}_i \rightarrow \mathcal{F}(\Theta)$ according to the expert opinions, such that

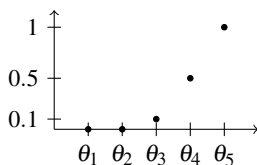


Fig. 5. Fuzzy set corresponding to the term *very reliable* defined on Θ with $N = 5$

$\Gamma_{A_i}(a)$ with $a \in \mathcal{A}_i$ is the interpretation on Θ of the information provided by $A_i = a$ about the reliability. We denote by μ_a the fuzzy set $\Gamma_{A_i}(a)$.

The expert may select from a limited number of linguistic terms as well as combination of them, using "or" disjunctions^[1]. An additional term allows to express total ignorance. A fuzzy set on Θ is then associated to each term. Fig. 5 provides an illustration of a fuzzy set corresponding to the term *very reliable*.

Example 5. Consider the two groups $A_1 = \text{source type}$ and $A_2 = \text{experience repetition}$ such that

$$A_1 = \{a_{11} = \text{journal paper}, a_{12} = \text{governmental report}, a_{13} = \text{project report}, a_{14} = \text{other}\}$$

$$A_2 = \{a_{21} = \text{repetitions}, a_{22} = \text{no repetition}\}$$

The expert then provides his opinion about the reliability value for the different values of these two criteria. These opinions are summarised below

$$\mu_{a_{11}} = \text{very reliable}, \mu_{a_{12}} = \text{slightly reliable}, \mu_{a_{13}} = \text{neutral}, \mu_{a_{14}} = \text{slightly unreliable};$$

$$\mu_{a_{21}} = \text{very reliable}, \mu_{a_{22}} = \text{slightly unreliable}.$$

3.2 Global Reliability Information through Merging

For a given data table each group A_i takes a particular value, hence S different fuzzy sets are provided as pieces of information. We propose to use evidence theory [11] to merge these information in a global reliability assessment. Indeed, this theory comes with a rich choice of merging rules [12], together with a good compromise between expressiveness and tractability. It encompasses fuzzy sets and probability distributions as special cases. We recall here the basics of the theory and its links with fuzzy sets.

A basic belief assignment (*bba*) m on a space Θ is a mapping from the power set $2^{|\Theta|}$ of Θ onto the unit interval $[0, 1]$, such that $\sum_{E \subseteq \Theta} m(E) = 1$ and $m(\emptyset) = 0$. Sets E such that $m(E) > 0$ are called **focal elements**. We denote by \mathcal{F}_m the set of focal elements of m . The mass $m(E)$ can be interpreted as the probability that the most precise description of what is known about a particular situation is of the form " $x \in E$ ". From this mass assignment, Shafer [11] defines two set functions, called *belief and plausibility functions*, for any event $A \subseteq \Theta$:

$$Bel(A) = \sum_{E, E \subseteq A} m(E); \quad Pl(A) = 1 - Bel(A^c) = \sum_{E, E \cap A \neq \emptyset} m(E),$$

¹ In this case, fuzzy sets are combined by the classical t-conorm max.

where the belief function measures the certainty of A (i.e., sums all masses that cannot be distributed outside A) and the plausibility function measures the plausibility of A (i.e., sums all masses that it is possible to distribute inside A).

A fuzzy set μ with M distinct membership degrees $1 = \alpha_1 > \dots > \alpha_M > \alpha_{M+1} = 0$ defines a *bba* m having, for $i = 1, \dots, M$, the focal elements E_i with masses $m(E_i)$ [13]:

$$\begin{cases} E_i = \{\theta \in \Theta \mid \mu(\theta) \geq \alpha_i\} = A_{\alpha_i}, \\ m(E_i) = \alpha_i - \alpha_{i+1}. \end{cases} \quad (1)$$

Therefore, each fuzzy set provided by experts during information collection can be mapped into an equivalent *bba*.

Example 6. Consider the fuzzy set depicted in Fig. 5. Its equivalent *bba* m is such that

$$m(E_1 = \{\theta_5\}) = 0.5, \quad m(E_2 = \{\theta_4, \theta_5\}) = 0.4, \quad m(E_3 = \{\theta_3, \theta_4, \theta_5\}) = 0.1.$$

When S groups of criteria (called sources in the sequel) provide pieces of information modelled as *bbas* m_1, \dots, m_S over a same space Θ , it is necessary to merge them into a global model. Two main issues related to merging rules are the handling of (i) dependence [14] and of (ii) conflict [12] between sources.

Here, sources are selected to remain as independent as possible, therefore tackling the first issue. We are thus left with the problem of properly handling conflicting information. Given the fact that sources are independent, the merging of *bbas* m_1, \dots, m_S can be written,

$$\forall E \subseteq \Theta \quad m(E) = \sum_{E_i \in \mathcal{F}_i}^{\oplus_{i=1}^S (E_i) = E} \prod_{i=1}^S m_i(E_i), \quad (2)$$

with \mathcal{F}_i the focal elements of m_i , and $\oplus_{i=1}^S (E_i) = E$ an aggregation operator on sets. Note that TBM conjunctive rule and the disjunctive rule [12] are retrieved when $\oplus = \cap$ and $\oplus = \cup$, respectively. However, the former is not adapted to the case of conflicting information, while the latter often results in a very imprecise model.

To deal with the problem of conflicting information, we propose a merging strategy based on maximal coherent subsets (MCS). Given a set of conflicting sources, MCS consists in applying a conjunctive operator within each non-conflicting subset of sources, and then using a disjunctive operator between the partial results [15]. Consider $N = \{I_{a_1, b_1}, \dots, I_{a_k, b_k}\}$ a set of k intervals. Using the MCS method on such intervals consists in taking the intersection over subsets $\overline{K}_j \subset N$ s.t. $\cap_{i \in \overline{K}_j} I_{a_i, b_i} \neq \emptyset$ that are maximal with this property, and then in considering the union of these intersections as the final result (i.e. $\cup_j \cap_{i \in \overline{K}_j} I_{a_i, b_i}$). We denote by \oplus_{MCS} the MCS aggregation operator. In general, detecting MCS is NP-hard, however in the case of intervals over an ordered space (our case here), the algorithm proposed in [15] reduce this complexity drastically.

An application of MCS on four (real valued) intervals I_1, I_2, I_3, I_4 is shown in Fig. 6. The two MCS are (I_1, I_2) and (I_2, I_3, I_4) and the final result is $(I_1 \cap I_2) \cup (I_2 \cap I_3 \cap I_4)$. Note that, if all intervals are consistent, conjunctive merging is retrieved, while disjunction is retrieved when every pair of intervals conflicts. As we shall see, the groups of intervals forming maximal coherent subsets may be used as elements explaining the result. Applying MCS in our case comes down to apply Eq. (2) with $\oplus = \oplus_{MCS}$ to the fuzzy sets $\mu_{a_{ij}}, a_{ij} \in A_i$ once they have been transformed into *bbas* (thanks to Eq. (1)).

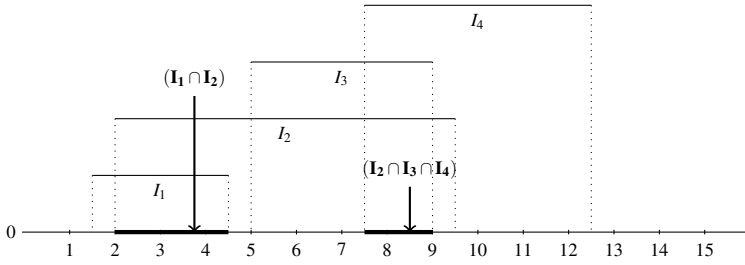


Fig. 6. Illustration of maximal coherent subsets merging

Example 7. Consider the two groups of Example 5. Now, assume that the retrieved data come from a journal paper ($A_1 = a_{11}$) but that the experiment has not been repeated ($A_2 = a_{22}$). Value a_{11} corresponds to "very reliable", while a_{22} corresponds to "slightly unreliable". Each group A_i thus provides an individual *bba* corresponding to the criterion value. These *bbas* are given in the following table:

$$\frac{E_{11} = \{\theta_3, \theta_4, \theta_5\}}{m_{a_{11}} \quad 0.1} \quad \frac{E_{12} = \{\theta_4, \theta_5\}}{0.4} \quad \frac{E_{13} = \{\theta_5\}}{0.5}$$

$$\frac{E_{21} = \{\theta_1, \theta_2, \theta_3\}}{m_{a_{22}} \quad 0.1} \quad \frac{E_{22} = \{\theta_2, \theta_3\}}{0.4} \quad \frac{E_{23} = \{\theta_2\}}{0.5}$$

We denote by j_i the index of the criterion value for a given data item, and m_g the *bba* obtained by merging $m_{a_{1j_1}}, \dots, m_{a_{2j_2}}$ through Equation (2) with $\oplus = \oplus_{MCS}$. The merging result of the *bbas* given in Example 7 is summarised in Table 1.

Table 1. Example of merging independent information using MCS

Criteria in MCS	MCS focal sets	Focal set	Mass of focal set
$\{A_1, A_2\}$	$E_{11} \cap E_{21} + E_{11} \cap E_{22}$	$\{\theta_3\}$	0.05
$\{A_1\}$ and $\{A_2\}$	$E_{11} \cup E_{23} + E_{12} \cup E_{22}$	$\{\theta_2, \dots, \theta_5\}$	0.21
$\{A_1\}$ and $\{A_2\}$	$E_{12} \cup E_{21}$	$\{\theta_1, \dots, \theta_5\}$	0.04
$\{A_1\}$ and $\{A_2\}$	$E_{12} \cup E_{23}$	$\{\theta_2, \theta_4, \theta_5\}$	0.20
$\{A_1\}$ and $\{A_2\}$	$E_{13} \cup E_{21}$	$\{\theta_1, \theta_2, \theta_3, \theta_5\}$	0.05
$\{A_1\}$ and $\{A_2\}$	$E_{13} \cup E_{23}$	$\{\theta_2, \theta_5\}$	0.25
$\{A_1\}$ and $\{A_2\}$	$E_{13} \cup E_{22}$	$\{\theta_2, \theta_3, \theta_5\}$	0.2

4 Reliability Presentation and Explanation

A look at Table 1 tells us that the merging result is hard to read, and that it is necessary to provide tools that summarize this information in a digestible representation. Given a set $D = \{e_1, \dots, e_d\}$ of d data, we propose three complementary means to summarise their reliability evaluations: by ordering them, by providing a summarising (quantitative) interval and by explaining the main reasons for the reliability evaluation.

In this section, we use the notion of lower and upper expectations of a function $f : \Theta \rightarrow \mathbb{R}$ induced by a *bba* m_g . These lower and upper expectations are defined as

$$\underline{\mathbb{E}}_g(f) = \sum_{A \subseteq \Theta} m(A) \min_{\theta \in A} f(\theta) \quad \text{and} \quad \overline{\mathbb{E}}_g(f) = \sum_{A \subseteq \Theta} m(A) \max_{\theta \in A} f(\theta). \quad (3)$$

They correspond to the infimum and supremum values of all expectations of f w.r.t. probability measures dominating the belief function induced by m_g .

4.1 Comparing, Evaluating and Ordering Data

Let m_{g_1}, \dots, m_{g_d} be the global *bbas* representing our knowledge about the reliability of e_1, \dots, e_d . We propose to induce an order between them by using numerical comparison of interval-valued estimations, using a particular function in Eq. (3). We propose to consider $f_\Theta : \Theta \rightarrow \mathbb{R}$ such that $f_\Theta(\theta_i) = i$ (each θ_i receives its rank as value), and to summarize the reliability of data item e_i by the interval $[\underline{\mathbb{E}}_{g_i}(f_\Theta), \overline{\mathbb{E}}_{g_i}(f_\Theta)]$ (obtained by using Eq. (3)).

Example 8. Consider the three *bbas* $m_{g_1}, m_{g_2}, m_{g_3}$ respectively representing the reliability of e_1, e_2, e_3 (e.g. resulting from the merging process illustrated in Example 7), defined over $\Theta = \{\theta_1, \dots, \theta_5\}$ such that

$$m_{g_1}(\{\theta_1, \theta_2, \theta_3\}) = 0.3, m_{g_1}(\{\theta_2, \theta_3\}) = 0.7; \quad m_{g_2}(\{\theta_3, \theta_4\}) = 0.5, m_{g_2}(\{\theta_4, \theta_5\}) = 0.5; \\ m_{g_3}(\{\theta_1\}) = 0.4, m_{g_3}(\{\theta_5\}) = 0.4, m_{g_3}(\{\Theta\}) = 0.2.$$

Corresponding reliability intervals are:

$$[\underline{\mathbb{E}}_{g_1}(f_\Theta), \overline{\mathbb{E}}_{g_1}(f_\Theta)] = [1.7, 3]; \quad [\underline{\mathbb{E}}_{g_2}(f_\Theta), \overline{\mathbb{E}}_{g_2}(f_\Theta)] = [3.5, 4.5]; \\ [\underline{\mathbb{E}}_{g_3}(f_\Theta), \overline{\mathbb{E}}_{g_3}(f_\Theta)] = [2.6, 3.4].$$

Partial order: We propose to order the *bbas* according to the (partial) order $\leq_{\mathbb{E}}$ s.t. $m_g \leq_{\mathbb{E}} m_{g'}$ iff $\underline{\mathbb{E}}_g(f_\Theta) \leq \underline{\mathbb{E}}_{g'}(f_\Theta)$ and $\overline{\mathbb{E}}_g(f_\Theta) \leq \overline{\mathbb{E}}_{g'}(f_\Theta)$. In Example 8, we have $e_1 <_{\mathbb{E}} e_3 <_{\mathbb{E}} e_2$ (further on, we make no difference between a datum e_i and its *bba* m_{g_i}), obtaining in this case a complete order among the objects. However, as $\leq_{\mathbb{E}}$ is in general a partial order, we propose an algorithm allowing to build from it a complete (pre-)order, so that users are provided with an ordered list, easier to understand and interpret.

Building groups: The next step is to order data by groups of decreasing reliability according to the order $\leq_{\mathbb{E}}$, i.e., to build an ordered partition $\{D_1, \dots, D_O\}$ of D , where D_1 corresponds to the most reliable data. Given a subset $F \subseteq \{e_1, \dots, e_d\}$, denote by $opt(\mathbb{E}, F)$ the set of optimal data in the sense of reliability, i.e. not dominated w.r.t. $\leq_{\mathbb{E}}$

$$opt(\mathbb{E}, F) = \{e_i \in F \mid \nexists e_j \in F, \text{ such as } e_i \leq_{\mathbb{E}} e_j\}.$$

The partition $\{D_1, \dots, D_O\}$ can now be defined recursively as follows:

$$D_i = opt(\mathbb{E}, (\{e_1, \dots, e_d\} \setminus \bigcup_{j=0}^{i-1} D_j)) \text{ with } D_0 = \emptyset. \quad (4)$$

4.2 Explaining the Results

Another interest of MCS is that they give insights about the reasons that have led to a particular reliability assessment, providing the user with some possibly useful explanations. Indeed, according to our method, the more often a subgroup F of MCS appears in $\oplus_{i=1}^S (E_i) = E$ (see Eq. (2)), the more important its impact is on the global reliability score m_g . Therefore, we propose to measure the importance $w(F)$ of a MCS F in m_g by summing all the masses of m_g for which it has been a maximal coherent subset, that is

$$w(F) = \left\{ \sum \prod_{i=1}^S m_i(E_i) \mid F \text{ is an MCS of } \oplus_{i=1}^S (E_i) \right\}$$

Example 9. In Table 11 the impact w of the different encountered MCS is evaluated as follows: $w(\{A_1\}) = 0.95$, $w(\{A_2\}) = 0.95$, $w(\{A_1, A_2\}) = 0.05$, from which it can be inferred that the two criteria $\{A_1\}$ and $\{A_2\}$ appear often alone and do not agree with each other. This means that the imprecision in the final reliability representation can be explained by the conflict between criteria A_1 and A_2 .

In this example, the analysis is straightforward. However, when dealing with thousands of data and half a dozen of criteria groups, such tools may help users to perform a quick analysis and retain the data that best serve their purposes.

5 Extending @Web for Data Reliability Management

This section describes the change made to @Web to add a reliability estimation to each Web table and to use them in the display of a user query result. As data from a same table often come from a same experiment, table level has been retained to model reliability.

5.1 Extending the Ontology to Include Reliability Criteria

Some criteria retained for reliability estimation are part of the domain knowledge. For example, measurements methods to count micro-organisms all roughly have the same precision, while the accuracy of methods to appreciate wheat grain size greatly varies. Therefore, it is natural to include criteria in the domain ontology. This solution allows designers to adapt the choice of the criteria associated with each domain of application, preserving the @Web generic approach at the same time.

In the extended version of the ontology integrating reliability criteria, the *core ontology* is enriched with corresponding symbolic and numeric criteria. The *domain ontology* is completed by the definition of the criteria selected to evaluate the reliability, together with their possible values. For example, the respectively symbolic and numeric criteria *SourceType* and *CitationNumber* are represented by OWL classes and belong to the *domain ontology*. They are subclasses of the generic classes *SymbolicCriterion* and *NumericCriterion*, respectively, which belong to the *core ontology*. As for symbolic concepts, the possible values associated with a symbolic criterion are represented by OWL classes that are subclasses of the OWL class representing the criterion.

5.2 Storing Data Reliability Criteria in RDF Graphs

In extended @Web, an additional fourth task concerns the reliability management (see Figure 1). Users manually enter the values associated with the reliability criteria for each Web table. This information is stored in a RDF graph associated with the table.

Example 10. Fig. 7 presents the RDF descriptions representing the reliability criteria and values associated with the Web table of Fig. 3. They express that the table (having the *uriTable1* identifier within the XML document) has for associated criteria the same values than in Example 7: journal paper and no repetitions of experiments.

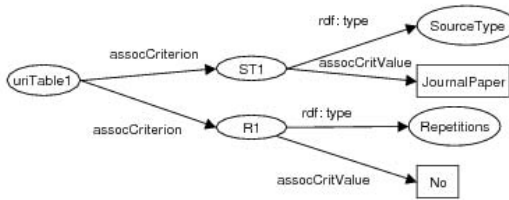


Fig. 7. Example of RDF annotations associated with the Web table of Figure 3

The fourth task output of the extended @Web system is an XML/RDF data warehouse composed of a set of XML documents which represent Web tables, together with the RDF annotations corresponding to the recognized semantic relations and the reliability criteria values.

5.3 SPARQL Queries and Data Reliability

To evaluate the reliability of the answers associated with a MIEL++ query, the following post-processing is executed. Reliability criteria associated to a Web table are retrieved thanks to SPARQL queries (generated by using the ontology). Each answer associated with a given row of a Web table is then associated to its reliability interval thanks to its URI which links it to its original table. Answers are then compared and ordered according to methods of Sec. 4.

6 Conclusion and Perspectives

In this paper, we have proposed a method that evaluates reliability of Web data tables by using sets of criteria concerning the data origins. This method, based on evidence theory, is generic and can be applied to any domain once proper criteria have been defined. Special attention has been given to tractability and ease of use. A first possible perspective of this work should be to take account of possible uncertainty in the criteria values. In the present paper, we have considered that criteria were known. It would be desirable to consider the case where some criteria are ill-known (using *bbas* to describe this uncertainty). A second possible perspective would be to extend our approach to cope with multiple experts providing (possibly) different opinions about the same criteria.

References

1. Gil, Y., Artz, D.: Towards content trust of web resources. In: WWW 2006: Proceedings of the 15th International Conference on World Wide Web, New York, NY, USA, pp. 565–574 (2006)
2. Quinn, K., Lewis, D., O’Sullivan, D., Wade, V.: An analysis of accuracy experiments carried out over a multi-faceted model of trust. *Int. J. of Information Security* 8, 103–119 (2009)
3. Denguir-Rekik, A., Montmain, J., Mauris, G.: A possibilistic-valued multi-criteria decision-making support for marketing activities in e-commerce: Feedback based diagnosis system. *European Journal of Operational Research* 195(3), 876–888 (2009)
4. Buche, P., Dibie-Barthélemy, J., Chebil, H.: Flexible SPARQL querying of web data tables driven by an ontology. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 345–357. Springer, Heidelberg (2009)
5. Hignette, G., Buche, P., Dibie-Barthélemy, J., Haemmerlé, O.: Fuzzy annotation of web data tables driven by a domain ontology. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 638–653. Springer, Heidelberg (2009)
6. Antoniou, G., van Harmelen, F.: A semantic Web primer. The MIT Press, Cambridge (2008)
7. Buche, P., Couvert, O., Dibie-Barthélemy, J., Hignette, G., Mettler, E., Soler, L.: Flexible querying of web data to simulate bacterial growth in food. *Food Microbiology* 28(4), 685–693 (2011)
8. Zadeh, L.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
9. Buche, P., Haemmerlé, O.: Towards a unified querying system of both structured and semi-structured imprecise data using fuzzy views. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS(LNAI), vol. 1867, pp. 207–220. Springer, Heidelberg (2000)
10. Thomopoulos, R., Buche, P., Haemmerlé, O.: Different kinds of comparison between fuzzy conceptual graphs. In: Ganter, B., de Moor, A., Lex, W. (eds.) ICCS 2003. LNCS(LNAI), vol. 2746, pp. 54–68. Springer, Heidelberg (2003)
11. Shafer, G.: A mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
12. Smets, P.: Analyzing the combination of conflicting belief functions. *Information Fusion* 8, 387–412 (2006)
13. Dubois, D., Prade, H.: On several representations of an uncertain body of evidence. In: Gupta, M., Sanchez, E. (eds.) Fuzzy Information and Decision Processes, pp. 167–181. North-Holland, Amsterdam (1982)
14. Denoeux, T.: Conjunctive and disjunctive combination of belief functions induced by non-distinct bodies of evidence. *Artificial Intelligence* 172, 234–264 (2008)
15. Dubois, D., Fargier, H., Prade, H.: Multi-source information fusion: a way to cope with incoherences. In: Cepadues (ed.) Proc. of French Days on Fuzzy Logic (LFA), pp. 123–130 (2000)

Propagation of Question Waves by Means of Trust in a Social Network

Albert Trias Mansilla and Josep Lluís de la Rosa Esteva

Universitat de Girona, Agents Research Lab, Escola Politècnica Superior, PIV
17071 Girona, Spain
{albert.trias,jluis.rosa}@udg.edu

Abstract. This paper proposes a model of behavior for agents who answer questions; the model works similarly to the way in which people interact in social networks and the agents behave differently depending on who asks a question; this behavior modulates the effort utilized in finding better answers for a given question. Our model also avoids consulting all acquaintances fact that can overload or overburden the contacts. However, since reducing the number of recipients might result in a poorer answer, we propose a behavior of consulting a small set of contacts and adding more recipients only if no relevant answer is found. The most promising result is that the first *answer-in* is probably the most relevant. The ordering the answers simply as they arrive gives the best ranking of answers. The new ranking is well-suited for real time question answering and avoids costly methods associated with re-ranking results.

Keywords: P2P, Question Answering, Trust, Social Networks, Query Routing, Multi-agent Systems.

1 Introduction

Centralized search engines are designed to satisfy the needs of the most of users and have been progressively made more personalized and context aware. Although they generally provide good results, centralized search engines are less effective when dealing with atypical searches [1], and the relevance of their results decreases due to the effects of search engine optimization techniques (SEO)¹. At the same time, public interest in social networking sites has grown. For example, in early 2011 Facebook had 550 million active users, according to its own website. Researchers [2, 3, 4, 5, 6] and companies are showing increasing interest in “the village paradigm” [2] or social search versus the “library paradigm”. There are new examples of Q&A portals such as Aardvark and Quora, with stronger social or 2.0 features; there are new web browsers such as Rockmelt, which integrates Facebook with the browser, allowing users to ask questions of their online acquaintances or to share relevant information. The most popular online social network sites are interested in question answering, as demonstrated by Facebook’s creation of Facebook Questions. Most of these new

¹ <http://dashes.com/anil/2011/01/threes-a-trend-the-decline-of-google-search-quality.html>

waves of Q&A portals connect users with their acquaintances, who probably are more motivated to help the user than a complete stranger.

In a social search, or in the village paradigm, the search problem is reduced to finding people who can cover the information need. That is why this paradigm's most important aspect is finding people with expertise in the question's domain who are willing to answer the question.

In this paper, Section 2 introduces the social search, while Section 3 contains an exposition on why multi-agent systems can be used in Q&A. Section 3 also shows that multi-agent systems constitute a possibility for automating Q&A. In Section 4, an initial attempt to automate social Q&A with query routing is shown. Section 5 covers our proposed model, and Section 6 shows our simulations and results. Our conclusions are in Section 7.

2 Social Search

Social Search is a type of search that uses social interactions, implicit or explicit, to obtain results. Chi [3] proposes the following classification for social search engines:

- *Social Feedback Systems* use social data to sort the results. This information can be obtained directly (ratings, tags or bookmarks) and indirectly (logs).
- *Social Answering Systems* are systems that use peoples' expertise or opinions to answer questions in a particular domain; the *answerers* can be friends, colleagues or strangers.

According to Chi's classification, Social Feedback Systems are unable to address new questions when the information is not available. Although Social Answering Systems solve this problem, the experts can be requested to answer the same question multiple times, and the stakeholders do not receive the content immediately.

Within Social Answering Systems, we consider knowledge exchange portals (Q&A). Knowledge and information exchanges in the form of questions and answers have emerged and grown along with the development of the Internet [7]. Likewise, 15% of the queries to the web search engines are completely formulated questions [8], despite the fact that a search using keywords does not always return relevant results.

Q&A portals are the result of attempts to address the limitations of search engines [9]. Some users use them when search engines do not return satisfactory results or when they are searching for opinions or customized advice. Personal knowledge exchanges are defined as market-based arrangements used to trade knowledge assets. Personal refers to the fact that the entities involved in the exchange are individuals and reflects the often "personal" nature of knowledge [10]. Market-based institutions offer promising mechanisms to evaluate knowledge and to design incentives to motivate and support valuable knowledge transfer [11]. Gosain [7] studied the key challenges of supporting the exchange of personal knowledge within web-based marketplaces and studied how personal knowledge exchanges can overcome this challenge [10].

Overall, the knowledge exchanges suffer from a number of drawbacks [8], with the most important drawbacks being:

- Lack of Answers.
- Users do not know either how to phrase their questions to match the answers or who to ask.

The **lack of answers** is caused by **not having** the right people **available** to provide the answers, or by having no knowledge available at all. People could be unavailable for several reasons: because **they are not willing to answer** or because they are **not accessible due to their low answer bandwidth** or because the question **did not get through to them**. Since the burden of answering questions is a consequence of low bandwidth, questions may be answered slowly.

One of the interesting problems of Social Answering Systems is predicting when the questioner will be satisfied with an answer, as satisfaction is subjective. This problem was studied between others by Agichtein and Liu [12]. They showed that humans had problem predicting questioners' satisfaction, but previous interactions were useful for predicting satisfaction, and that is why their system outperformed human classifiers by using classification algorithms. Another key step is the matching between answerers and questions [13], one option is recommending the questions to the potential answerers [2], usually based on the expertise and past performance. These approaches are top-down. A bottom-up approach is considerably more modular, scalable and synergic with the collective (2.0) approaches that are now at the core of Internet: let every agent be associated to a user collect his/her knowledge and roles within a number of communities. One of the key differences between our work and the ones above is that our approach is bottom-up and p2p, while the other work is centralized and top-down. To do so, we have an intensive use of agents, though it is not the first attempt of using them in Social Answering Systems, as for example the approach proposed by Galitsky and Pampapathi [14], and other examples like MARS and 6Search can be found in section 3.

There are mainly three options to provide askers good results minimizing the answering time [13]: the first one is reusing existent content in the Social Answering System [15, 17]; the second one consists on routing questions to experts on the topic of the question [18,19,20]; and the third one is about trying to understand the answering behavior to encourage more answers [13,20].

Having an agent able to receive the questions, qualify them, or relocate them to other valuable users, and even in certain cases try to answer the questions, would add high value to the automation of social answering systems. Such an agent could deliver answers faster, possibly nearly as well or in as relevant a manner as could most people in most cases, agents can reuse previous answers avoiding that users answering the same questions once and again and motivating them to answer new questions. Furthermore agents can help routing the questions in a subjective way. Due to these important considerations, we next analyze what properties of intelligent agents are necessary to be used for social answering systems.

In the AgentLink Roadmap [21], Luck et al. claim that agent technology can be considered from three perspectives: as a design metaphor, as a source of technologies or as a simulation. We consider agent technology as a design metaphor, where agents provide a way of structuring the application by means of autonomous and communicative entities.

Features of the intelligent agents that are relevant for social answering are reactivity, sociability, proactivity, and autonomy. Agents' reactivity is defined as their reaction after receiving a question, that they will decide what actions to perform (ignore it, try to answer it, show it to its owner, forward it); agents' sociability is defined based on their capability of asking the question to other agents; agents'

proactivity is defined based on their taking initiative without being explicitly asked to have their users maintain updated knowledge bases; and finally, agents' autonomy defined as the ability of the agents to work independently of any other entity.

Furthermore, agents have enabled support representation, and coordination and cooperation among heterogeneous users and their processes. Internet and software agents enable the construction of information systems from multiple heterogeneous sources and contribute to improving the relationship between suppliers and consumers of knowledge to give the agents better control of the interactions [22]. These agents appear to offer the best approach for the automation of social networks for knowledge exchanges and are a good match for P2P systems.

On the other hand, the oldest Q&A automation was provided by listing FAQs and their answers on web pages or text documents. This system was a popular way to provide answers to common questions [10].

Using FAQs as a way of automating Q&As is not simply uploading an FAQ page to a personal web page. Instead, FAQs as a way of automating of Q&As is like sharing one's knowledge sources, monitoring the questions coming in and being answered, proactively asking for Q&A updates, creating new pairs of Q&As, and much more. These actions are combined with privacy management, as not everything is meant to be public or indexable in its first instance. These tasks are the job of the Q&A agent.

Our idea is based on each agent containing its user's personal FAQ list. When an agent receives a question with an unknown answer, the agent can proactively decide to ask the user to complete the question with further explanations and optionally it could add the question to its user's FAQ. With this approach, our attempt is to reuse previous pairs of questions and answers. In a prior study [23], 30% of the time that a query was performed, it had been carried out before by the same user, and 70% of the time it was searched before by an acquaintance of the user. When an agent receives a query that it cannot answer, and the question is not appropriate for its owner, the agent can forward the question to other agents.

3 First Attempts in the Automation of Social Q&A

Sixearch (6S) [5] proposed to use distributed systems and social networks to address the problem of the web search. In their work, queries are sent through the social network, and the contacts that receive the query can forward the question or answer, as in the breadth first search used by Gnutella protocol. Also these researchers use the TTL (Time to Live), which determines when a question can be forwarded. The answer, which consists of a set of bookmarks and documents, follows the inverse path of the question.

In [1], Walter proposed a recommender system in which agents exploit the social network to obtain information filtered through trust based on a breadth first search. Walter [1] proposed to compute the Trust Path T_{a_i, a_j} in the graph to rank the answers. In the literature, there are discussions about trust transitivity; its usage depends on the scenario and on whether trust can be used in recommender systems with a discounting [1]. The answer is chosen randomly with probabilities assigned by a logit function (equation 1).

$$T_{a_i, a_j}(t + 1) = \begin{cases} \gamma T_{a_i, a_j}(t) + (1 - \gamma)r_k & \text{for } r_k \geq 0 \\ (\gamma - 1)T_{a_i, a_j}(t) + \gamma r_k & \text{for } r_k < 0 \end{cases} \quad (1)$$

Where:

- r_k is the experience that a_i has made following the recommendation about o_k transmitted by a_j .
- $T_{a_i, a_j}(t) \in [-1, 1]$ is the trust value and $T_{a_i, a_j}(0) = 0.5$.
- $\gamma \in [0, 1]$, indicates the dynamics of the trust. For values of $\gamma > 0.5$ trust is increased slowly but can decrease fast, which is usually a desired property for the dynamics of trust.

MARS (Multi-Agent Referral) [6, 24, 25] is a P2P social network that uses agents to help their users obtain referrals to find experts who might answer the user's question. Agents determine to which contacts to send each question. Also, when an agent receives a question, the agent decides whether to show the question to its owner and whether to provide referrals to the questioning agent.

Michlmayr [26] proposed a model for query routing in P2P networks based on Ant Colony Optimization. In her proposal, an ant represents a query and when it reaches a peer with documents that satisfies the query, then a backward ant, which follows the inverse path, dropping pheromone, represents the answer message. In the case of the selection of the contact using pheromones, the most recent experiences are most important, as pheromones evaporate over time. $\tau_{cu} \leftarrow (1 - \rho)\tau_{cu}$, where $\rho \in [0, 1]$ is the parameter that determines the amount of pheromone evaporated. This model can be similar to the model proposed by Walter with a $\rho = \gamma = 0.5$; the difference is that in the model of Walter, the "evaporation" happens when there is a new evaluation, and the decision is done by the acquaintance instead of by the query.

4 Model: Behavior of Agents in Social Q&A and Question Waves

We base our model in the ASKNEXT protocol [27], in which each user is represented by an agent who can forward the question, as in a BFS [1, 5, 26]. If the agent finds that a question is of interest to its user, then the agent can show the question to her [6]. In our model, an agent can try to answer using the knowledge that it has indexed, or, alternatively, can try to obtain new knowledge from its user, or, finally, can ask acquaintances. In this protocol, for each question the agents can play three roles:

- **Questioner:** The questioner is the agent who started the question.
- **Mediator:** A mediator is an agent who receives a question and forwards it to another.
- **Answerer:** An answerer is an agent who answers a question with its knowledge or with its user's knowledge.

4.1 Behavior of Agents: Give and Take

We claim that in most of the cases, the agent's behavior has to be based on *reciprocity*. In [28], the authors explain that social exchange theory assumes that people try to have balanced relationships (reciprocity); people prefer relationships where they give and receive a similar amount of support. We think that agents need to

have a *give & take* behavior. Reference [28] points out that if there is a discrepancy between giving and receiving, then the continuation of the relationship is threatened; furthermore, there are some cases in which there are unbalanced relationships, as can be the case of family or close friends. In close relationships, people feel responsible for each other's well-being and do not consider the balance. For these reasons we think that a Q&A agent will have contacts classified into two groups: one group for whom the agent feels responsible and does not consider the balance in the relationship, and another group for whom the agent expects the relationship to be balanced. We will call the first group **close acquaintances** and the second group **convenient acquaintances**.

In this paper we will use the term *reciprocity* to refer to the effort that an agent makes on answering a question from an acquaintance. We believe that there are at least three factors that would affect the result of the task given to the agent. The first factor is the expertise of the answerer (E); the second factor is the answerer's implication (I) and the third factor consists of external factors (R). By *implication* we mean the amount of effort that an agent makes when performing a task; the agent can be motivated by rewards, by the task itself, or by who asks for the task. The external factors (R) can include considerations such as the personal situation of the answerer, whether she has free time, whether she is in a good or bad mood, whether she has work overload, whether she is tired, etc. As the result of a question will depend on the difficulty of the question, we model the result as a threshold θ for the implication, expertise, and other factors to be considered in relevant answers. The answer quality (ϑ) is denoted in equation 2, where $\{\vartheta, \theta, I, E, R\} \in [0,1]$.

$$\vartheta = f(I, E, R, \theta) \tag{2}$$

We can express ϑ in logic (equation 3) that can be implemented by applying a strong conjunction with Łukasiewicz (equation 4).

$$\vartheta = I \wedge E \wedge R > \theta \tag{3}$$

$$\vartheta = F_{\otimes}(I, E, R) = F_{\otimes}(F_{\otimes}(I, E), R) = \max(0, \max(0, I + E - 1) + R - 1) \tag{4}$$

We think that an agent will put its best effort into helping close acquaintances; in the case of convenient acquaintances, the amount of effort that the agent will extend will depend on the benefit that the agent takes from the relationship while trying to maintain its balance. We will compute trust as the benefit that one agent a_1 takes from an agent a_2 , meaning the quality of the answers that a_1 received from a_2 for the benefit of agent a_1 .

4.2 Question Waves

A question wave is an attempt to find an answer to a question. In every attempt, the same question is sent to a subset of acquaintances. The expectancy of finding appropriate answers decays after every attempt. The wave propagates through the network of agents, which amplify or attenuate it. The advantage of the question wave is that multiple agents are committed to finding the answers, with diversified options to find them, resulting in more relevant², faster³, and robust⁴ answers.

² Relevant in the sense that answers come ranked by trust.

³ Faster in the sense of reducing the burden of questions, and the agents are less overwhelmed.

⁴ Robust in the sense of finding answers persistently.

Question waves try to solve the following problem: in P2P, to request a question to all possible peers is not feasible because it can overload the system. However, reducing the number of recipients too far can provide the worst results. Furthermore, we believe that it is not feasible to request one agent after another when a response (with or without answers) is obtained, because this peer can be disconnected, can ignore the query, or can wait forever for the reply of an acquaintance. Deciding which peers are to receive requests is not any easy task. We think that the agents can decide to request this task to new recipients, as humans do, as a function of the current outcome for a task over time.

In each attempt, the question sender (questioner or mediator) selects the most reliable acquaintances who have not been selected before for this question. Adding new recipients implies that the agent is not sure that it will receive any answer from the current recipients and tries to obtain an answer from less trusted recipients. Furthermore, this approach can be used when useful answers are received, enabling the user to read them in real time (when the search process is not complete) as a heuristic from the most to the least relevant.

We think that this scheme can be implemented, dividing the possible recipients into different groups from most trusted to least trusted and programming the message sending. At some time, the message sending would be canceled when the agent decided that the question was resolved with the current answers.

5 Simulations

Our simulations consist of a set of agents $A=\{a_0, a_1, \dots, a_i\}$, that, at each simulation step, perform the following algorithm:

Method Step

```

For each Received Answers
If Own Question, Update result and Trust
    Else If not answer forwarded yet
        Forward it and update trust
If I have a new Own question
    Select contacts in contact waves; Program messages
For each received question
    If I received it before, ignore it
    Else If I am good enough for answering,
        Generate Answer Value; Send answer
    Else Select contacts in contacts waves; Program messages
Send programmed messages
  
```

One agent decides whether it has enough knowledge to answer a question in the function of the sender evaluation $Ev(a_i)$. As part of the reciprocity, the higher the evaluation an acquaintance has, the more the agent will try to give a better answer. The agent will answer if its $e_j > Ev(a_i) - \sigma$. We used $\sigma = 0.1$.

In our simulations the implication of agents are computed by the evaluation of the requesting contact as an information source ($Ev(a_i)$), and the distance d from the questioner agent, because we believe that people use less effort helping friends of friends instead of direct friends. We computed ($Ev(a_i)$) as the mean of trust value for each domain (mean) or as the maximum trust value in any domain (max). The implication value is denoted by equation 3, where TTL_{MAX} is the maximum distance that a question can reach.

$$I = \frac{Ev(a_i)(TTL_{MAX} - d + 1)}{TTL_{MAX}} \quad (5)$$

In these simulations we compute ϑ with equation 6, because we want to give the most importance to some variables over others. ϑ is the answer quality, α is the weight of the Implication (I), β the weight of the expertise (E), δ the weight of the external factors (R).

$$\vartheta = \alpha I + \beta E + \delta R \quad (6)$$

We updated trust values based on a hybrid between the models proposed by Walter [1] and Michlmayr [18]; the trust updated for each node is about its neighbor as in [18] instead of the path [1], but we used the equations from [1], with a $\gamma=0.8$ (equation 7).

$$T_{a_i,j}(t+1) = \begin{cases} \gamma T_{a_i,j}(t) + (1-\gamma)r_k \text{ for } r_k \geq 0.7 \\ (\gamma-1)T_{a_i,j}(t) + \gamma r_k \text{ for } r_k < 0.7 \end{cases} \quad (7)$$

We modeled the question waves consisting of 4 waves. The 1st wave arrives after 1 simulation step; the 2nd wave arrives after 5 simulation steps, the 3rd after 20 simulation steps and the last after 40 simulation steps.

Agents are sorted into the different waves (1st, 2nd, 3rd and 4th) by the trust of the questioner. In our implementation, wave 1 goes in the 1-to- t_1 partition, while wave 2 goes in the t_1 -to- t_2 partition, and so forth. The vector $T=\{t_1, t_2, t_3\}$ defines the trust partitions for the different waves. For experimentation, we used the following classifications: $T = \{[0.8, 0.7, 0.6], [0.85, 0.8, 0.7], [0.85, 0.75, 0.5], [0.85, 0.7, 0.5]\}$.

When a mediator receives an answer after it already has forwarded an answer, the mediator can use two strategies: the first strategy is to ignore the new answer as the task is complete for this question; the second strategy consists of forwarding the answer if the evaluation of the answer is better than the previous answer. In this case, the new answers are used to update the trust.

We modeled our simulations using some points in common with [25]:

- The expertise vector E of each agent has dimension 5. The value of $e_j \in [0,1]$ of an expertise vector $E = \{e_1, e_2, e_3, e_4, e_5\}$ means the expertise level in domain j . E values are set randomly.
- Agent a_i will generate an answer from his expertise vector E when there is a good match between the query and its expertise vector.
- There is a random value that affects the answer quality. Its value is between 0 and 1. In our case, this Randomness has a weight of δ and represents R .

- The querying agents rate the services from P_i as ϑ'_i and $\vartheta'_i = \vartheta_i$.
- The queries correspond to vectors of length 5 that are 1 in one dimension and 0 in all other dimensions. For each query q , $\exists! i \mid q_i = 1$ and $\forall j, j \neq i \mid q_j = 0$.

Also we added the following points:

- At each step, each agent has a question probability of having its own question (Ψ). In our simulations, $\Psi = 0.05$.
- The interest vector $It = \{i_1, i_2, i_3, i_4, i_5\}$ with the same dimension of vector E denotes the probability that a question will belong to domain j . $\sum_j i_j = 1$.
- Simulations have 3,000 simulation steps and 200 agents.
- The graph that represents the social network is undirected (but connections can have different weights as trust values). The number of friends that each agent added is 10, with a mean of 20 acquaintances per agent. We only consider acquaintances by convenience.
- The initial trust from each agent to its acquaintances is 0.75 for each domain.
- We repeated each configuration 10 times, setting as the random seed the values from 0 to 9.
- In the equation (6) we used the parameters: $\alpha = 0.2$, $\beta = 0.7$ and $\delta = 0.1$.

5.1 Evaluation

As answers are qualified by ϑ using equation 6 as the answer quality then, to evaluate results, we used the Spearman correlation⁵ between the set of answers sorted by ϑ and the set of answers sorted by some heuristics explained below:

- Answer Distance (D): The idea is that with answers with more close answers, it is more possible that the answerer agents know the questioner; also the answerer will be more motivated to help a friend than a friend of a friend.
- Trust of the last sender (Tr): The idea with this ranking is that the important concern is the information source that we asked. From where that source takes the information does not matter; Tr is implicit on the trust of the information source.
- Receiving Order (H): The idea is that all the contributors are searching first in the most relevant sources of information, so we expect that the best results are found first.
- Answer Distance and Trust (DT); Receiving Order and Trust (HT). As the distance evaluation and the order receiving can have several items with the same value, we decided to break the tie with the trust of the last sender.

5.2 Results

Table 1 contains the results from the simulations. We computed a mean comparison test for each configuration, observing that the results from each heuristic were different. D has the worst ranking. We suspect that, in our implementation, as the agent's implication (I) decreases with distance then this heuristics has higher significance. Perhaps its weak correlation to relevance is due to the random

⁵ We used the java library jsc (Java Statistical Classes) <http://www.jsc.nildram.co.uk>

generation of the social networks used in our experiments, where trust relations among peers were not built on the basis of their knowledge needs and, moreover, agents do not have the ability to start new relationships other than the ones of its user, and neither to cut some of the current relationships. We believe that if agents had the chance of modifying their contact list regarding their needs, there would be an increase in the relevance of this heuristics.

The second worst correlation is shown by the Tr heuristics. We believe that, at least with these input data, the trust on the answerer is more important than that on the mediator. If instead of Tr we had used the trust of the last mediator (TLM) or the trust transitivity (TT) as heuristics, we think we would have improved the correlation to the set of answers ranked by ϑ . In the case we had used the TLM heuristics then the agents should inform their contacts about the trust on the last mediator, though it would not be desirable in some cases where privacy should be observed. The Tr heuristics (trust of the last sender) has poor correlation to ϑ , but seems to contribute a little the improvement of the H and D heuristics.

The heuristics H shows the best performance, specifically with $T = [0.8, 0.7, 0.6]$. The best answer quality is obtained with $Ev(a)_{max}$, because the implication factor will always be greater than with the mean. As in this paper we do not consider the cost of obtaining a good answer, as overrating the agents will bring the best results.

Table 1. Simulation Results

$Ev(a)$	T	D	H	DT	HT	Tr	ϑ
mean	.8,.7,.6	.14	.67	.17	.66	.14	.66
mean	.85,.8,.7	.10	.49	.16	.48	.17	.68
mean	.85,.75,.5	.11	.43	.16	.43	.19	.67
mean	.85,.7,.5	.12	.56	.16	.55	.16	.67
max	.8,.7,.6	.23	.7	.27	.69	.14	.72
max	.85,.8,.7	.13	.62	.2	.61	.2	.73
max	.85,.75,.5	.15	.6	.23	.59	.22	.72
max	.85,.7,.5	.19	.67	.24	.65	.16	.72

We highlight that with our algorithm the best answers are more likely to come up than the other answers. This is a very important claim, which is supported by the highest correlation of H to the quality of answers, much higher than the trusted ranking of answers. The average correlation of H goes from 0.43 to 0.7, while Trust is poorly correlated, around 0.14 and 0.22.

6 Conclusions

The method of question waves simplifies the estimation of how many recipients will receive a question. It will be useful for automating question answering.

Question waves add recipients dynamically when no relevant answer is found. Thus, having the “*perfect*” set of recipients is not critical, at least at first, because the agents will improve the list of recipients over time with respect to the search results. The state of the art requires having calculated the best recipients at the moment of processing a search, and this is not necessary for our approach any longer.

Furthermore, this method allows answers to be obtained with a *first-in-is-the-best-answer* approach; our simulations show a best correlation with our approach compared with using other heuristics as acquaintances’ trust or answerers’ distance.

Perhaps in a case in which we dealt with real opinions it would be harder for mediators to find an answering agent with a strong similarity to the questioner agent, and our *first-in-is-the-best-answer* approach would not work robustly. To solve it, we are interested in performing simulations with data related to opinions as future work.

Acknowledgments. This research is funded by the EU project Num. 238887 (iSAC6+), the ACCIÓ Catalan Government grant ASKS, the Spanish MCINN projects IPT-430000-2010-13 project (SAKE) and TIN2010-17903, the Universitat de Girona research grant BR09/10 and the Talència grant for the CSI-ref.2009SGR-1202.

References

1. Walter, F.E., Battiston, S., Schweitzer, F.: A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems* 16(1), 57–74 (2008)
2. Horowitz, D., Kamvar, S.D.: The anatomy of a large-scale social search engine. In: *Proceedings of the 19th Intl. Conf. on World Wide Web*, pp. 431–440. ACM, New York (2010)
3. Chi, E.H.: Information seeking can be social. *Computer* 42(3), 42–46 (2009)
4. Hendler, J.: Avoiding another AI winter. *IEEE Intelligent Systems* 23(2), 2–4 (2008)
5. Wu, L.S., Akavipat, R., Maguitman, A., Menczer, F.: Adaptive P2P social networks for distributed content based web search. In: *Social Inf. Retrieval Syst. Emergent Tech. and Appl. for Searching the Web Effectively*. IGI Global (2007)
6. Yu, B., Singh, M.P.: An agent-based approach to knowledge management. In: *CIKM 2002: Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pp. 642–644. ACM, New York (2002)
7. Gosain, S.: Issues in designing personal knowledge exchanges: First movers analyzed. *IT & People* 16(3), 306–325 (2003)
8. Liljenback, M.E.: ContextQA: Experiments in interactive restricted domain question answering, Master’s thesis, San Diego University (2007)
9. Harper, F.M., Moy, D., Konstan, J.A.: Facts or friends?: distinguishing informational and conversational questions in social Q&A sites. In: *CHI 2009 Proceedings of the 27th Intl. Conf. on Human Factors in Computing Systems*, pp. 759–768. ACM, New York (2009)
10. de la Rosa, J.L., Rovira, M., Beer, M., Montaner, M., Givobic, D.: Reducing the Administrative Burden by Online Information and Referral Services. In: *Citizens and E-Government: Evaluating Policy and Management* (2010)
11. Malhotra, Y.: Enabling knowledge exchanges for e-business communities. *Information Strategy* 18(3), 26–31 (2002)

12. Agichtein, E., Liu, Y.: Modeling Information-Seeker Satisfaction in Community Question Answering. *ACM Transactions on Knowledge Discovery from Data* 3(2), Article 10 (April 2009)
13. Liu, Q., Agichtein, E.: Modeling Answerer Behavior in Collaborative Question Answering Systems. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) *ECIR 2011*. LNCS, vol. 6611, pp. 67–79. Springer, Heidelberg (2011)
14. Galitsky, B., Pampapathi, R.: Can Many Agents Answer Questions Better than One? *FirstMonday* 10(1) (2005)
15. Bian, J., Liu, Y., Agichtein, E., Zha, H.: Finding the right facts in the crowd: factoid question answering over social media. In: *WWW*, pp. 467–476 (2008)
16. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: *CIKM*, pp. 919–922 (2007)
17. Jurczyk, P., Agichtein, E.: Discovering authorities in question answer communities by using link analysis. In: *CIKM*, pp. 919–922 (2007)
18. Liu, X., Croft, W.B., Koll, M.: Finding experts in community-based question-answering services. In: *CIKM*, pp. 315–316 (2005)
19. Zhou, Y., Cong, G., Cui, B., Jensenm, C.S., Yao, J.: Routing questions to the right users in online communities. In: *ICDE*, pp. 700–711 (2009)
20. Guo, L., Tan, E., Chen, S., Zhang, X., Zhao, Y.E.: Analyzing patterns of user content generation in online social networks. In: *KDD*, pp. 369–378 (2009)
21. Luck, M., McBurney, P., Shehory, O., Willmott, S.: *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink (2005)
22. Dignum, V.: An overview of agents in knowledge management. In: Umeda, M., Wolf, A., Bartenstein, O., Geske, U., Seipel, D., Takata, O. (eds.) *INAP 2005*. LNCS (LNAI), vol. 4369, pp. 175–189. Springer, Heidelberg (2006) ISBN 3-540-69233-9
23. Smyth, B., Balfé, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community based web search engine. *User Modeling and User-Adapted Interaction* 14(5), 383–423 (2005)
24. Yu, B., Singh, M.P.: Searching social networks. In: *AAMAS 2003: Proceedings of the 2nd Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*, pp. 65–72 (2003)
25. Yu, B., Singh, M.P., Sycara, K.: Developing Trust in Large-Scale Peer-to-Peer Systems. In: *First IEEE Symposium on Multi-Agent Security and Survivability*, pp. 1–10 (2004)
26. Michlmayr, E., Pany, A., Kappel, G.: Using Taxonomies for Content-based Routing with Ants. *Journal of Computer Networks* (2007)
27. Trias, A., de la Rosa, J.L., Galitsky, B., Drobocsi, G.: Automation of social networks with Q&A agents (extended abstract). In: *Proceedings of the 9th International Conf. on Autonomous Agents and Multiagent Systems*, pp. 1437–1438
28. Ikkink, K.K., van Tilburg, T.: Broken ties: reciprocity and other factors affecting the termination of older adults' relationships. *Social Networks* 21, 131–146 (1999)

Investigating the Statistical Properties of User-Generated Documents

Giacomo Inches¹, Mark James Carman², and Fabio Crestani¹

¹ Faculty of Informatics, University of Lugano, Lugano, Switzerland
{giacomo.inches, fabio.crestani}@usi.ch

² Faculty of Information Technology, Monash University, Melbourne, Australia
mark.carman@monash.edu

Abstract. The importance of the Internet as a communication medium is reflected in the large amount of documents being generated every day by users of the different services that take place online. In this work we aim at analyzing the properties of these online user-generated documents for some of the established services over the Internet (Kongregate, Twitter, Myspace and Slashdot) and comparing them with a consolidated collection of standard information retrieval documents (from the Wall Street Journal, Associated Press and Financial Times, as part of the TREC ad-hoc collection). We investigate features such as document similarity, term burstiness, emoticons and Part-Of-Speech analysis, highlighting the applicability and limits of traditional content analysis and indexing techniques used in information retrieval to the new online user-generated documents.

1 Introduction and Motivations

Communication is a primary need of the human being and the advent of the Internet amplified the possibilities of communication of individuals and the masses [1]. As result, many people every day use chat and instant messaging programs to get in touch with friends or family, or rely on online services such as blog or social networks to share their emotions and thoughts with the Internet community [2].

The increasing popularity of these online-based services (Twitter, Facebook, IRC, Myspace, blogs, just to mention few of them) results in a production of a huge number of documents generated by Internet users. It is therefore of great interest to study the properties of these online user-generated documents: from a commercial point of view we could identify new trends and hot topics by mining them [3,4], so as better focus advertisement or new online services; from a policing perspective, instead, it may allow us to detect misbehaviour [5,6,7]. Again, it is also interesting from a research point of view to understand the linguistic properties of such documents or their statistical properties to improve the current models and techniques used in Information Retrieval [8]. Since this kind of documents are more recent and less studied, compared to more consolidated collections (like e.g. the TREC ad hoc), we need to understand them as clearly

as possible in order to perform effective and valuable mining and/or retrieval on them [9].

We discuss the motivating related work in Section 2 and present the datasets used for our analysis in Section 3. In Section 4 we describe the analysis performed and the metrics used and conclude in Section 5 with a summary of the results of our study and an overview on the future work.

2 Related Work

Inches et al. [10] provided a preliminary analysis of the statistical properties of online user-generated documents. The authors show their “shortness” in terms of average document length and their “messy” nature due to spelling/mistyping errors as well as the fact that the terms occurrences followed a standard Zipfian distribution. Other works on online short documents focused more on clustering [11], on topic detection [12] or similarity measures [13], but without considering the general properties of the different collections analysed in each work.

Some work has already been done in trying to categorize the online documents based on their properties [7], leading to a distinction between chat- and discussion-like documents. More general properties of text can be found in the work of Serrano et al. [14], where the authors focus on different properties of “standard” written text with the purpose of developing a new and more complete model for the description of written text. More general purpose introductions to textual analysis can be found in [15] and [16].

This work aims at extending the analysis in [10] with the study of standard text properties such as the ones presented in [14] and to integrate them with a Part-Of-Speech (POS) study. To the best of our knowledge, neither of these analysis has been thus far performed on such collections. Instead, POS analysis has already been applied to queries [17], term weighting [18] and on text blocks [19], just to mention some topics.

3 Datasets

Our analysis aims at comparing user-generated documents and standard information retrieval ones, therefore we choose as representative of the first class four datasets containing different user-generated content, namely *Kongregate* (Internet Relay Chat of online gamers), *Twitter* (short messages), *Myspace* (forum discussions) and *Slashdot* (comments on news-posts). These datasets were first presented at the Workshop for Content Analysis in Web 2.0 [20] and are divided between training and testing data¹. Our analysis take into consideration only the train dataset for each collection, which is enough to our purposes.

As collections representative of standard information retrieval documents we employed three datasets of similar edited content: news articles from the *Associated Press* (AP, all years), the *Financial Times Limited* (FT, all years) and

¹ Datasets and details available at <http://caw2.barcelonamedia.org/>

the *Wall Street Journal* (WSJ, all years). These datasets form a representative subset of the standard TREC Ad-hoc collection² and, although they are similar in the type of content, they cover different topics: *AP* and *WSJ* report news in general, while *FT* focuses on markets and finance.

We notice that these collections show a similar topicality to the particular *Myspace* and *Slashdot* datasets we use: The *Myspace* dataset covers the themes of campus life, news & politics and movies, while the *Slashdot* dataset is limited to discussions of politics. The fact that the themes are similar to the news articles is important in order to make statistical comparison between the collections meaningful. As for the topicality of the *Twitter* and *Kongregate* datasets, due to their conversational and more unpredictable nature, we cannot state precisely what their topicality is [21,3,12].

We report in Table 1 some basic statistics about these datasets. The difference in the average document length is evident: the user-generated document collections contain documents that are remarkably short compared to the news articles. We will examine in Section 4 the implications of this property in terms of the document self-similarity and burstiness, where we will explain also the role of common and rare words.

Table 1. Statistics of datasets

	avg. doc. length (# words)	avg. word length	# Common words (% of the vocabulary)	# Rare words (% of the vocabulary)
Kongregate	4.50	7.55	489 (1.39)	29'805 (84.65)
Twitter	13.90	7.30	716 (0.20)	354'131 (97.19)
Myspace	38.08	8.11	743 (0.39)	179'757 (96.10)
Slashdot	98.91	7.88	560 (0.45)	118'276 (95.88)
WSJ	452.00	7.57	1003 (0.44)	219'332 (96.85)
AP	464.23	7.53	1217 (0.40)	298944 (97.34)
FT	401.22	7.26	1017 (0.36)	271'055 (97.23)

4 Analysis of the Datasets

4.1 Similarity

The first property that we study is the self-similarity between documents, which we compute using the cosine similarity between td-idf document vectors.

The comparison was performed between each pair of documents in the collection for a total of $\frac{N(N-1)}{2}$ comparisons for each collection (where N is the number of documents in the collection, available in [20,10]). We choose the WSJ to represent the TREC collections and display the values of the self-similarity computed after having removed the stopwords³ from the documents. The most evident difference between the user-generated documents (*Kongregate*, *Twitter*,

² Datasets and details available at http://trec.nist.gov/data/test_coll.html

³ Standard Terrier stopwords list.

Myspace and *Slashdot*) and the standard ones (represented by the *WSJ*) can be observed at the extremes of the similarity scale. For this reason, in Fig. 1 we zoom in to show only the percentage of document pairs with the lowest (left) and highest (right) similarity scores.

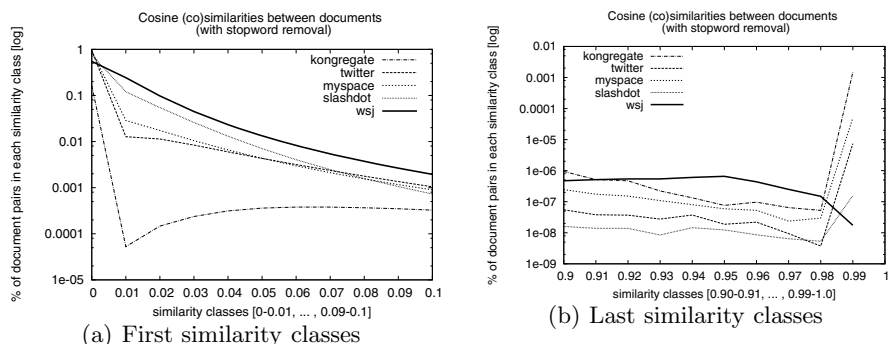


Fig. 1. Self-similarity between documents after stopword removal. We normalized the count for document in each similarity class by the total number of comparisons.

In the first case, we observe that user-generated documents appear less frequently with lower similarity values (0.01-0.09), as they become shorter. To the contrary, they appear more frequently with higher similarity values (0.9-1.00), contrasting the behaviour of the standard documents. The latter, in fact, drop down when we consider only the last similarity range (0.99-1.00).

This means that shorter documents seem to be more similar across themselves than the longer ones. This can be explained with the length of the documents itself: short documents contain less words (less “information”). Therefore, given two short documents, there is an higher probability that they appear to be similar even if they are unrelated, just because they are short.

To counteract this behaviour of the shorter user-generated documents we would need to enlarge the information they carry whenever we want to process them. Different solutions can be applied to this problem, which we leave for future study. We just mention two techniques which we could use: stream segmentation and document expansion. With stream segmentation we aim at merging documents together based on their temporal proximity, which raises the problem of setting proper boundaries for joining them, while with document expansion we could extract relevant information from the documents, such as internet links or tags, and retrieve from other sources new words to enlarge their topicality.

4.2 Burstiness

We present in this section the second analysis, where we study the burstiness of the terms in each collection. Plots in Fig. 2 show the percentage of documents

in each collection that contains a certain number of common or rare words. Common words are defined as the most frequent words in the vocabulary that account for more than 71% of the text in the collection, while rare words are the least frequent words in the vocabulary that account for 8% of the text, as computed also in [14] (Table II).

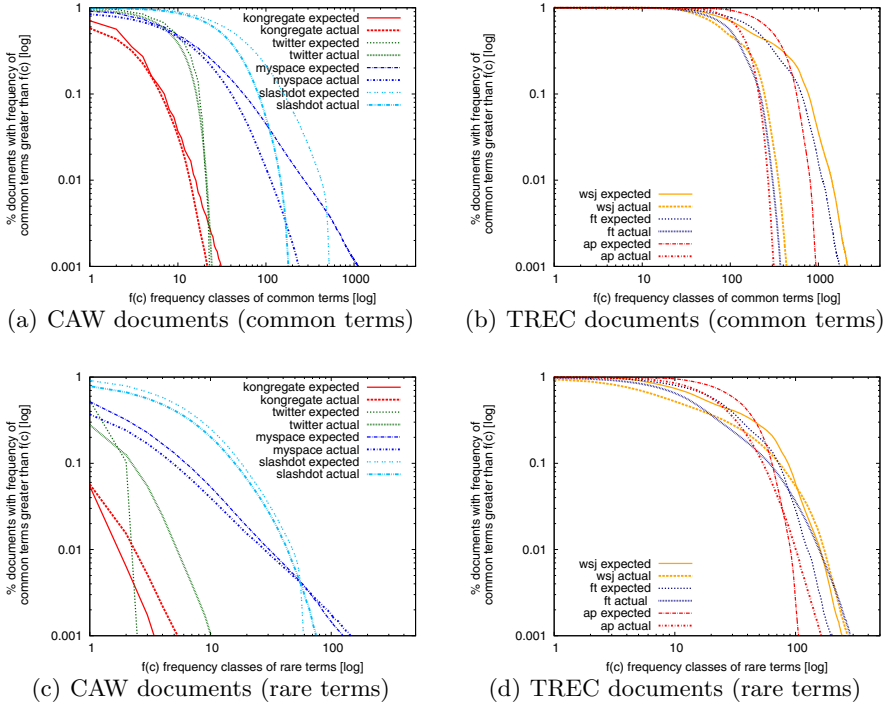


Fig. 2. Common and rare term burstiness for user-generated documents (CAW) and traditional ones (TREC)

In each plot we show also the expected number of such documents if the words in the vocabulary were uniformly distributed (according to their overall frequency in the collection) across the documents in the collection. Differences between the curves for actual and expected number of documents indicates burstiness in the collection, i.e. the phenomenon that a word observed once within a document is far more likely to re-occur within the same document than it is to occur in another document chosen at random.

Looking at the common terms plot for the three edited collections (*AP*, *FT* and *WSJ*), we see that the line denoting the actual number of documents with a certain number of common terms in them lies well below the expected number of such documents. This indicates that documents are bursty, since common terms are not spread evenly across the collection of documents, but are concentrated more in some documents than others. The same is true (although to a less

extent) for the rare terms in these collections: the actual number of documents containing a certain number of rare words lies below the expected curve, again indicating that documents are bursty, since the rare words are not uniformly distributed across documents.

Comparing the plots for user-generated content (*Kongregate*, *Twitter*, *Myspace* and *Slashdot*) with those for the edited collections, we see that the difference between the expected and actual number of documents is far less pronounced for the new collection (especially for the common terms) than it is for the traditional ones. This indicates that burstiness may not be an important issue for user-generated content as it is for traditional collections. This may have implications in document-length normalization for these collections: as we already noticed in Section 4.1 we should eventually pre-process them and expand their informative content through the use of document expansion or segmentation.

The fact that the expected/actual curves for the different user-generated collections differ greatly from one another is due to the large difference in average document length in the different collections. The curves for the edited collections (especially for the common terms) line up quite well due to the fact that the average document length is very similar.

4.3 Part-Of-Speech Distribution

In the third part of our work we employ GATE⁴ and its built-in tokenizer, sentence splitter and Part-Of-Speech (POS) analyser called ANNIE⁵ [22,23] to analyse the Part-Of-Speech (POS) tags distribution in the different datasets.

We report in Fig. 3 the results of the POS extraction through ANNIE of the full text on 30% of the documents in the collection, selected at random (since we did not find significant variation in the distributions with an higher subset). We used the ANNIE default settings for each component of the processing chain (tokenizer, sentence splitter and POS extractor) and report in Fig. 3 only the most significant categories⁶.

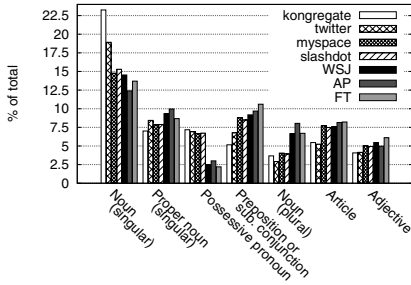
If we study in detail the results of Fig. 3 we can observe two different collection behaviors: first, we notice some inter-collection variations, between the user-generated datasets and the traditional datasets, then we perceive an intra-collection variation, inside the user-generated datasets, between chat-style and discussion-style documents.

Inter-collection differences can be seen in the usage of proper nouns, possessive pronouns and plural noun Fig. 3(a) as well as in the usage of verb and adverb Fig. 3(b). An explanation for this may be found in the nature of the documents contained in each collection: in the user-generated texts the user producing them is focused in expressing his/her point of view or emotions against the others (high usage of possessive pronouns), qualifying the amount of their sensations (high usage of adverb), addressing directly in first person (high usage of verb not in

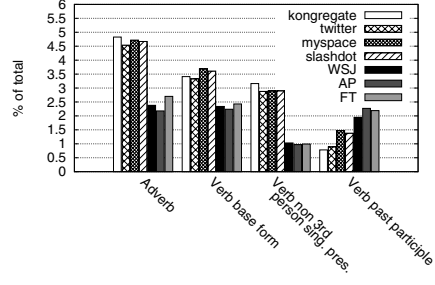
⁴ GATE: “General Architecture for Text Engineering”, <http://gate.ac.uk/>

⁵ ANNIE: “A Nearly-New Information Extraction System”, <http://gate.ac.uk/>

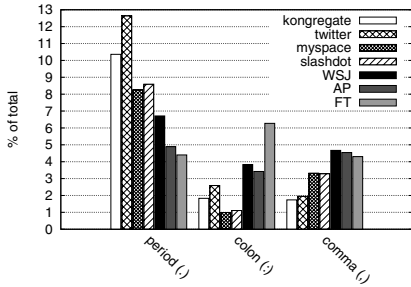
⁶ A complete list of the POS tag extracted by ANNIE can be found on <http://tinyurl.com/gate-pos>



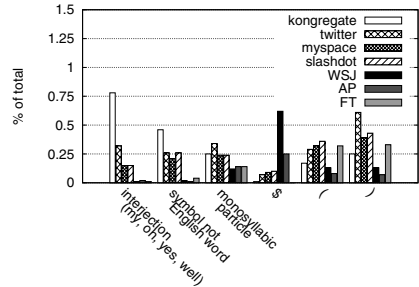
(a) nouns, pronouns, articles and adjectives



(b) verbs and adverbs



(c) high frequency punctuation



(d) interjections, symbols and low frequency punctuation

Fig. 3. POS analysis

the third person singular) and referring to action occurring mostly in the present time (verb in base form). To the contrary, texts that are edited in a professional way report events occurred in the past (high usage of verb in past participle), not occurring to the author itself (high usage of third person in the verb) or taking place in a particular location (higher use of singular proper noun). Again, if we take a detailed look at the punctuation, interjection and symbols in Fig. 3(c) and Fig. 3(d) we observe how user-generated documents consist of a more direct, personal and simple communication, given by a more extensive usage of interjection, symbols, monosyllabic particles and periods. Edited content, instead, is more descriptive, due to the usage of colons and commas, which generally link together different concepts inside the same sentence. A last observation regards the usage of brackets, which are more employed in the user-generated documents. We suppose they should be used in combination with colons and semicolons, building the so called emoticons, to enrich the expressiveness of the communication. We therefore analyse the usage of the emoticons in Section 4.4.

Intra-collection differences can be seen within the user-generated collection, where some datasets (*Myspace* and *Slashdot*) appear to be more related to the edited texts than the others (*Kongregate*, *Myspace*), which highlight different

properties. These properties are an high usage of proper singular nouns, periods, interjections and symbol, and a less usage of articles and adjectives, which becomes the least among all the collection for verbs in the past form and commas. They can be seen as attributes of an essential and immediate communication, such as the online-chat (*Kongregate*) or similar to chat (*Twitter*). On the other hand, for some POS categories the *Myspace* and *Slashdot* datasets are similar or just in-between to and with the *trec* datasets: this appear for preposition and subordinative conjunction, adjectives (Fig. 3(a)), verb in the past partiple form (Fig. 3(b)) as well as for periods, commas (Fig. 3(c)) and interjections (Fig. 3(d)). Following the approach proposed in [7], we label these documents as discussion-style documents.

These inter-collection and intra-collection differences can be used together with the measure of similarity and burstiness to give a preliminary classification of a dataset of unseen documents (standard/edited content or user-generate content, if user generated, chat- or discussion-style) as well as to help the retrieval of documents from a collection of a given type.

4.4 Emoticons and “Shoutings” Distribution

In this last part of our work we complement the POS analysis of Section 4.3 by investigating the distribution of emoticons and “shoutings” among the different collections. These features, in fact, can be very discriminative for identifying user-generated content [24] and in particular conversational data [3].

We collected a list of the most common emoticons (mostly through Wikipedia, see attachment A for a complete list) and parsed each document by comparing each token separately with a regular expression, thus identifying and counting only whitespace separated emoticons (such as :) and :P).⁷ In a similar way we counted so-called “shoutings”, that we define as whitespace separated tokens containing a succession of three-or-more consecutive instances of the same letter (e.g. *zzzz* and *mmmmaybe*). We did not include in this count tokens containing internet addresses (*www* and *WWW*) since they does not provide additional information on the collections being analysed.

In Fig. 4 we report the distribution of the emoticons and shoutings among the collections. The values represented are the relative collection frequency in both the linear and log scale. The behaviour of the distributions is similar and reflect the nature of the collections. User-generated collections (*Kongregate*, *Twitter*, *Myspace*, *Slashdot*) contain a large number of colloquial and informal tokens, such as emoticons and shoutings, that are used to improve the expressiveness of the communication. In the more standard and “professional” documents (*WSJ*, *AP*, *FT*), on the other hand, the communication remains on a formal and neutral level (having these collection almost zero counts for emoticons and shoutings and at least 1 order of magnitude less than the others).

⁷ We experimented also with matching emoticons within sequences of characters like *hello:)mum* but obtained too many false positives to consider those results valid. For the same reason, we did not count emoticons containing whitespaces such as *:)*.

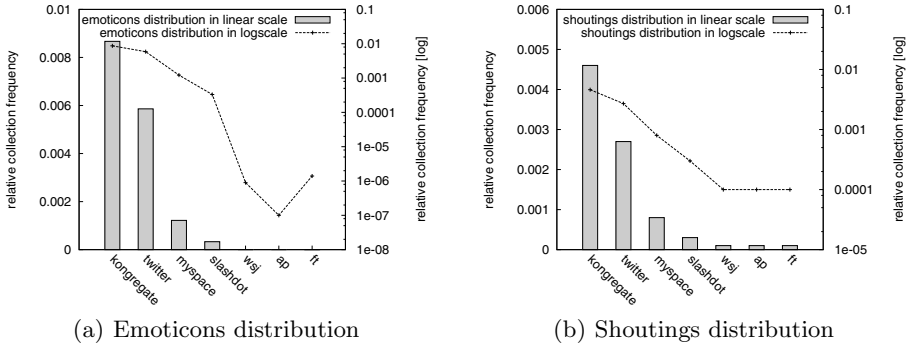


Fig. 4. Collection relative emoticons and shoutings distributions

Table 2. Top 10 emoticons in each dataset with their relative frequency as a percentage of all emoticon occurrences. We omit the few counts for WSJ,AP,FT since they are not informative. Emoticons in *italic* express a negative feeling (sadness), all the others a positive one (happiness, astonishment, smartness, tongue, smiley,...).

	Kongregate		Twitter		Myspace		Slashdot	
	emoticon	%	emoticon	%	emoticon	%	emoticon	%
1	:P	16.89	:)	43.35	:)	33.13	:)	37.26
2	XD	13.09	;))	11.12	;))	12.84	;))	17.75
3	:)	12.72	:~)	10.22	:P	10.84	:~)	14.92
4	:D	10.92	:D	8.78	:D	8.93	;~)	10.56
5	- . -	5.11	;~)	5.31	:]	4.61	:P	5.42
6	xD	4.62	:P	5.15	XD	3.47	:D	2.94
7	:0	3.45	:-(1.82	:p	2.84	B)	1.94
8	=D	2.95	XD	1.42	=P	2.39	:-(1.36
9	:p	2.84	:p	1.36	xD	2.37	:p	1.19
10	=P	2.72	:~D	1.10	:~)	1.61	:~P	1.04

As for the POS features analysed in Section 4.3, beside the inter-class differences highlighted above, it is also important to notice that we can highlight some intra-class differences among the user-generated documents: the more chat and colloquial documents (*Kongregate* and *Twitter*) contain more emoticons and shoutings occurrences (on the order of 1 or 2 levels of magnitude) than the documents that are more of a discussion-style.

These observations reinforce our conclusions about the usage of the features analysed in this work to improve the mining and retrieval of user-generated documents, which are nowadays of great interest for their novelty and popularity. To provide a practical example, we are currently applying these POS, emoticons and shoutings features to the TREC Blog08 collection [8, 25] to improve the

⁸ <http://trec.nist.gov/tracks.html>

results for the Faceted Blog Distillation Task, to distinguish for example between personal (more colloquial) and formal (more neutral) blogs. Another application of these features, especially the emoticons, can be found in the problem of detecting opinionated blog content (where we can search for emoticons expressing particular feelings, as in Table 2).

5 Conclusions and Future Work

In this work we analysed two different collections, a new sets of user-generated documents and a traditional one, containing edited documents.

In the first part of our study we computed the self-similarity between pairs of documents and analysed the term burstiness in each collection. We were able to identify one issue related to the length of the documents: for user-generated documents, in fact, their extreme shortness (compared to the traditional ones) makes them to be too little informative (when they are a lot) or too much informative (when they are not). To this purpose, we identified two techniques which could be used to address this problem but we left their implementation and evaluation to future studies.

In the second part of our study we performed a Part-Of-Speech (POS) analysis on a representative samples of our collections and we found that there exist some significant differences in the usage of the grammatical elements within the different datasets: noun, verbs, adverb as well as punctuation, interjections and symbols can be used to distinguish between new user-generated and traditional edited-content. They can also be applied to identify more chat-style content than discussion-style text inside the class of user-generated documents, since the latter are more traditional content likely, while the first ones show up new properties.

In the last part of the study we reinforce the conclusions drawn in the second part of our work, noticing how two particular features, the emoticons and the shoutings, also allow us to identify differences between “traditional” and user-generated content. Moreover they allow to distinguish between different types of user-generated documents: chat-like and discussion-like.

Taking into consideration the results of all parts of this works, we could extend it focusing on the document expansion or stream segmentation for each of the specific document classes: for chat-style and discussion-style documents we could consider the topic coherent portion of the entire chat or discussion as a document unit, instead of the single message or post (as we did in here). We are current applying the features discussed in this work to the TREC Blog Distillation Task and in future studies we plan to further refine the POS analysis, studying the occurrences of blocks of categories, similar to the work of Lioma and Ounis [19]. We also plan to tune the POS parser to better fit our collection, being able to detect categories (name-entity recognition) for the purpose of sentiment analysis [26,27] or polarity detection. To conclude, we could also like to use the statistical properties identified in this study to classify the content type of an unknown collection (user-generated vs edited and/or chat vs. discussion) and use this information for resource selection.

A List of Emoticons Used

```
(Z.Z) (-.-)Zzz Zzz :) :-) :-] :] :> :-> => ^_^ ^-^ (^_^) ^.^ ;) ;-) ;] ;-] ;> ;->
(^_^) ^_^ ^_* :wink: :( :-( :[ :-[ =( =[ :< =< :D :-D :D :D =D X-D XD
XD xD BD 8D X3 x3 :P :-P :-p :p =P =p :| :-| 8) 8-) B) B-) :'( :'-(:'[:
:~-[ :~( =~[ :~< :~< =~< T_T T.T (T_T) Y_Y Y.Y (Y_Y) _ . (.) ;~; ;~; ;~; :~:
_._ :S :-S =S @_@ :~? :? ?_? ?_? :~") :-~") :/) :-/) :~0 :0 :~o) :o :~o) :0
=0 =0 =o =_ =- -.- \ -.- o_o o.o o.o O_o O_o o_o - (-) _
(.) (oO) (Oo) . o_o O_o O.o o.o Oo oO >_< . _ U_U u_u (U_U) <3 ()(.)()
U.U u_u (U.U) U.u V_v v_v (V_V) V.V v.v (V.V) <_< >_> *_* (*_*) (*_**) (*.*
*.* :-* :-x :-X :-X :X ~*~ \o/ _ (.) x_x X_X (X_X) x.x X.X (X.X)
#.# (#.#) 0w0 (0w0) (***) *** :-Q_ :-Q__ :-Q___ :Q_ :Q__ :Q___ :Q_ (: (-:
:Q__ :Q___ :-Q :Q ==Q_ ==Q__ ==Q___ =Q_ =Q__ =Q___ =Q_ =Q__ =Q_Y i.i i_i
=Q___ ==Q =Q ** =3 :-3 :3 x3 :-@ :@ >:-@ >:@ >: >_< _ @>--
@:) @:-) @:-] @:] @:> @:-> @=> @=) @=] \\\(^_^)/ *<:o) ^o^ T.T Y.Y T_T
```

References

1. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: WebKDD/SNA-KDD 2007: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56–65. ACM, New York (2007)
2. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about twitter. In: WOSP 2008: Proceedings of the First Workshop on Online Social Networks, pp. 19–24. ACM, New York (2008)
3. Haichao Dong, S.C.H., He, Y.: Structural analysis of chat messages for topic detection. *Online Information Review* 30(5), 496–516 (2006)
4. Kucukyilmaz, T., Cambazoglu, B., Aykanat, C., Can, F.: Chat mining for gender prediction. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 274–283. Springer, Heidelberg (2006)
5. Medina, E.W.: Military textual analysis and chat research. In: International Conference on Semantic Computing, pp. 569–572 (2008)
6. Bache, R., Crestani, F., Canter, D., Youngs, D.: Mining police digital archives to link criminal styles with offender characteristics. In: Goh, D.H.-L., Cao, T.H., Sølvsberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 493–494. Springer, Heidelberg (2007)
7. Yin, D., Xue, Z., Hong, L., Davison, B.D., Kontostathis, A., Edwards, L.: Detection of harassment on web 2.0. In: CAW 2.0 2009: Proceedings of the 1st Content Analysis in Web 2.0 Workshop, Madrid, Spain (2009)
8. Qi, H., Li, M., Gao, J., Li, S.: Information retrieval for short documents. *Journal of Electronics (China)* 23(6), 933–936 (2006)
9. Wang, F., Greer, J.: Retrieval of short documents from discussion forums. In: *Advances in Artificial Intelligence*, pp. 339–343 (2002)
10. Inches, G., Carman, M., Crestani, F.: Statistics of online user-generated short documents. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 649–652. Springer, Heidelberg (2010)
11. Carullo, M., Binaghi, E., Gallo, I.: An online document clustering technique for short web contents. *Pattern Recognition Letters* 30(10), 870–876 (2009)
12. Tuulos, V.H., Tirri, H.: Combining topic models and social networks for chat data mining. In: WI 2004: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 206–213. IEEE Computer Society, Washington, DC, USA (2004)

13. Metzler, D., Dumais, S., Meek, C.: Similarity measures for short segments of text. In: Amati, G., Carpineto, C., Romano, G. (eds.) *ECiR 2007*. LNCS, vol. 4425, pp. 16–27. Springer, Heidelberg (2007)
14. Serrano, M., Flammini, A., Menczer, F.: Modeling statistical properties of written text. *PLoS ONE* 4(4), e5372 (2009)
15. Baeza-Yates, R.A., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
16. Manning, C.D., Schütze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge (1999)
17. Allan, J., Raghavan, H.: Using part-of-speech patterns to reduce query ambiguity. In: *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 307–314. ACM, New York (2002)
18. Lioma, C., Blanco, R.: Part of speech based term weighting for information retrieval. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) *ECIR 2009*. LNCS, vol. 5478, pp. 412–423. Springer, Heidelberg (2009)
19. Lioma, C., Ounis, I.: Examining the content load of part of speech blocks for information retrieval. In: *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pp. 531–538. Association for Computational Linguistics, Morristown (2006)
20. Codina, J., Kaltenbrunner, A., Grivolla, J., Banchs, R.E., Baeza-Yates, R.: Content analysis in web 2.0. 18th International World Wide Web Conference (April 2009)
21. Ramage, D., Dumais, S., Liebling, D.: Characterizing microblogs with topic models. In: *ICWSM* (2010)
22. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics* (2002)
23. Wilcock, G.: Introduction to linguistic annotation and text analytics. *Synthesis Lectures on Human Language Technologies* 2(1), 1–159 (2009)
24. Balog, K., Bron, M., He, J., Hofmann, K., Meij, E.J., de Rijke, M., Tsagkias, E., Weerkamp, W.: The university of amsterdam at trec 2009: Blog, web, entity, and relevance feedback. In: *TREC 2009 Working Notes*. NIST (November 2009)
25. Macdonald, C., Santos, R.L., Ounis, I., Soboroff, I.: Blog track research at trec. *SIGIR Forum* 44(1), 58–75 (2010)
26. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. In: *Proceedings of the International AAAI Conference on Weblogs and Social Media* (2010)
27. Ku, L.W., Ke, K.J., Chen, H.H.: Opinion analysis on caw 2.0 datasets. In: *CAW 2.0 2009: Proceedings of the 1st Content Analysis in Web 2.0 Workshop*, Madrid, Spain (2009)

Information Retrieval from Turkish Radiology Reports without Medical Knowledge

Kerem Hadımlı and Meltem Turhan Yöndem

Dept. of Computer Engineering,
Middle East Technical University,
Ankara, Turkey

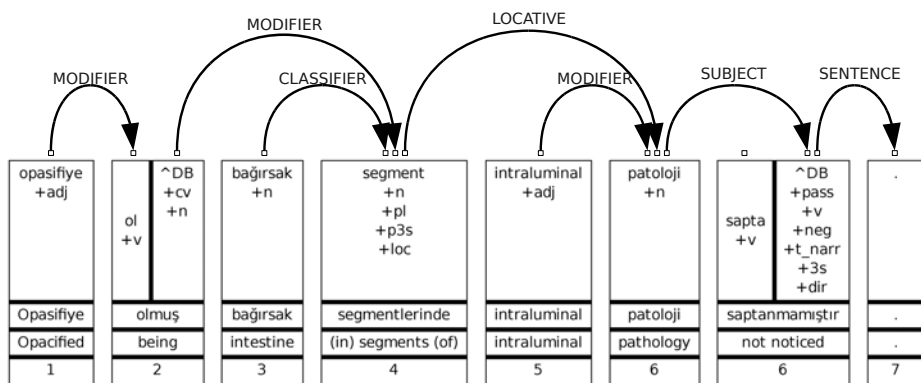
Abstract. It is observed that a person with no medical knowledge can still partially understand contents of Turkish radiology reports. Based on this observation, one rule based method and one data driven method for information retrieval from Turkish radiology reports are proposed. Both methods lack use of medical ontologies and medical lexicons in order to test the limits of the observation in isolation of other factors.

1 Introduction

Different imaging technologies are utilized by radiology departments to visualize patients' bodies, and these resulting images are examined by radiologists to write full text narrative reports. These reports contain a vast amount of medical information, and it is assumed that shared knowledge exists between writer and reader [12]. Although this assumption holds, we noticed that a person with no medical knowledge or education can still understand the basic relations within Turkish radiology report sentences, even if some of these might be incorrect due to missing knowledge. These basic relations may later be used to understand if a phrase mentions an anatomic location, a finding or a quality.

The reason for our observation lies in the properties of Turkish language. Turkish is characterized by a rich agglutinative morphology, free constituent order, and predominantly head-final dependencies [4]. Word structures in Turkish are formed by productive affixations of derivational and inflectional suffixes to root lexemes. Morphemes appended to a word can change the word's part of sentence tag (derivational morphemes), and these morphemes may be appended consecutively to form a series of different meanings [9]. Moreover, only the final POS tag is not sufficient to completely analyze a sentence, as syntactic relations may be formed between a word's earlier inflectional groups and other words in a sentence [13,10,11]. Also inflectional morphemes play an important role in syntactic analysis of a sentence. An example Turkish radiology report sentence may be seen in Figure 1. Different inflectional groups of single words are shown as separated by *DB* sign (derivational boundary).

Turkish radiology reports present even more challenges than English counterparts as there is no consensus on Turkish spelling of most Latin medical terms. Each radiologist tends to write the same terms differently. This is not a misspelling problem, rather a choice caused by differences in medical education.



Intraluminal(5) pathology(6) was not noticed(6) in opacified(1) segments(4) of intestine(3).

Fig. 1. A sample sentence, as a dependency graph and morphological features

The agglutinative nature of Turkish requires morphological analyses of not only common words, but also medical terms. This poses a problem for morphological analysis and disambiguation.

In this paper we propose two methods for information retrieval from Turkish radiology reports. Both methods lack the use of any medical lexicon or ontology, and rely solely on information on Turkish language (except for inherent medical information within training examples). Use of medical lexicons or ontologies is left out deliberately in order to see the results of our initial observation.

2 Rule-Based Method

Rule based method is based on hand crafted local templates and rules. These templates and rules are based on the subset of Turkish grammar which is used on our sample reports. These sample reports are from a variety of classes. A coarse distribution of the sample reports may be seen in Table 1.

Table 1. Distribution of sample reports

Report Type (Coarse grouping)	Number of Reports
Thoracic	8
Brain, neck	7
Abdominal	7
Bones, joints	7
Spinal	4
Angiography	2
Other	8
Total	43

The algorithm examines surface structures within a sentence, and extracts a set of relations between phrases. Phrase boundaries are determined automatically. The extracted relations are mostly between two phrases, although multi-phrase relations may also be extracted depending on matching rules. The extracted relations will be called as *individual meanings* in this paper.

Proposed method transforms a given sentence into a set of *individual meanings*. Each *individual meaning* has a type (from a finite set), and optional parameters that may map to phrases in the sentence. An example of a type of *individual meaning* is *LOCATION*, which would have *where* and *what* parameters. Finite set of types can be thought as the limited vocabulary of the transformation, although contents of parameters are unrestricted.

The algorithm used is multi-phased, and the first phase requires correct morpheme identification. Due to the simplicity of the method only the morphemes of the last inflectional group are necessary.

2.1 Morphological Analysis and Workarounds

For the rule based algorithm, morphological analysis is performed using Zemberek library [1]. Zemberek analyzes a given word string by trying to regenerate it from a known list of roots, morphemes, and rules governing agglutination. Zemberek, by design, may not try to generate all of the derived forms, but always generates the complete list of alternatives for inflections. Common derived forms of words were chosen to be included in its lexicon. This design choice makes its results less vulnerable to overly generating uncommon derivations. Zemberek is used as a syntax checker for Turkish and Turkic languages in many software packages.

In our case, regenerating the input word strings from a fixed dictionary of root words makes morphological analyzer vulnerable to medical terms that do not exist in its lexicon, but which are still used with agglutination in our reports. In order to overcome this limitation, a workaround is devised. Morphemes are made optional in the templates that match parts of sentences (first phase of algorithm).

Karaciğeri	konturları	, büyüklüğü	ve parankim
liver+POSS3S	contours+ACC	, size+ACC	andparenchymal
yapı		normal olarak	izlenmi stir.
structure+ACC		is normal.	
Contours, size and parenchymal structure of liver is normal.			

Hemaperitonium	,	mesane	lümeni	inde	hematom	ve hava.
Hemoperitoneum	,	urinary-bladder cavity+ACC+LOC	hematoma	and air.		
Hematoma and air in urinary bladder's cavity, hemoperitoneum.						

Fig. 2. Sample report sentences. In first example, ACC stands for accusative and POSS3S stands for 3rd singular person possession morphemes. In the second example, note that the locative suffix, *-de*, might extend to *hemoperitoneum* if it was an anatomical location.

Thus, for most rules, two alternative forms exist; one that matches a morpheme analyzed by Zemberek, and a second form that uses the surface representation of the morpheme. Although we may use surface forms of morphemes, still using a morphological analyzer provides us with a priority selection. If a word can be analyzed by the morphological analyzer, then only the rules with morphemes are put into effect. Otherwise, rules with surface forms allows us fallback to string suffix matching. Turkish has complex morphotactics and surface forms of even a single morpheme are affected by a variety of processes such as vowel harmony, vowel and consonant elisions or modifications in both morphemes and roots [11,9]. This prioritization of trusting a morphological analyzer over checking string suffixes allows us to be more error-tolerant in selecting surface forms to be included in the rules.

2.2 Algorithm

The algorithm is applied in two phases. Each phase uses its own rule set. Rule set of the first phase is actually a list of templates, which match to the target sentence locally, generating an initial set of *individual meanings*. The resulting *individual meanings* after the first phase are all local, i.e. they describe the relations between consecutive phrases. Boundaries of phrases are determined automatically using the keywords of matched templates as delimiters. In the second phase, the resulting *individual meaning set* of the first phase is expanded by applying another group of templates to the set itself. In the end an *individual meaning set* describing the whole sentence is obtained.

First phase templates match to the morphemes in the Turkish language and stopwords in the reports, whereas second phase rules extract nonconsecutive information such as effects of distributivity of a preposition.

First Phase. First phase rules are lists of templates, grouped into *individual meaning* types. Each template contains keywords that are matched into the sentence, and blanks that will be filled with phrases. Keywords may be morphemes, string suffixes, or whole words. Blanks will usually appear at the beginning and end of a template, and will have an attached parameter name, identifying the phrase that matches a blank area. The blank areas do not match to single words, but rather to all words that are in between keywords.

Sample templates for Location and Conjunction *individual meanings* can be seen in Table 2. If morphological analyzer is successful first template will match. Otherwise second template will match. Note that the blank areas are numbered. (a) and (b) represent *where* and *what* parameters. (c) and (d) represent *left* and *right* parameters. Keywords that align with input sentence are underlined.

When multiple templates match to the input sentence, blank parts between keywords are extended and concatenated. This assures that medical terms consisting of multiple words can be grouped together as a single phrase. The result of template-matching on the sample sentence can be seen in Table 3.

Table 2. Sample templates for some Individual Meanings

Type	Template	Matches to
LOC	..(a).. +LOC ..(b)..	lümen+POSS3SG+LOC hematom
	..(a).. + “de” ..(b)..	lümeninde hematom
CONJ	..(c).. , ..(d)..	hematom <u>ı</u> hava
	..(c).. ve ..(d)..	hematom <u>ve</u> hava
Hemaperitonium <u>ı</u> mesane lümeninde hematom <u>ve</u> hava		
CONJ <u>ı</u>	
LOC	de
CONJ	 <u>ve</u>
(I)..... <u>ı</u>(II).....de ... (III)... <u>ve</u> ..(IV)..	

Fig. 3. Application of first phase templates. Blanks are expanded between keywords

Table 3. Set of individual Meanings after first phase

Type	Parameter 1	Parameter 2
Conjunction	(left) Hemaperitonium	(right) mesane lümen
Location	(where) mesane lümen	(what) hematom
Conjunction	(left) hematom	(right) hava

Second Phase. The second phase of our algorithm works on a set of *individual meanings*. Second phase rules are responsible for expansion of the input set with *individual meanings* which exist between nonconsecutive phrases in the sentence.

Rules of the second phase specify two input *individual meanings*, and a resultant output *individual meaning*. In each iteration, any two *individual meanings* are selected from the input set, *meaning-A* and *meaning-B*. If both *A* and *B* satisfy the rule, then a third *individual meaning*, *meaning-C* is instantiated and added to the set. This process continues until no more expansions are possible.

Currently three rules are implemented. First two rules handle the left-distributivity and right-distributivity of *LOCATION individual meanings* over *CONJUNCTION individual meanings*.

The third rule handles proper expansion of adjective phrases. These phrases take the form of a qualifier word followed by a phrase, and can be chained. When faced with these phrases, first phase templates will generate *individual meanings* that are formed by the qualifier word and the right hand side phrase. Any *individual meaning* that is in left hand side of this *individual meaning* will be bound to the qualifier word instead of the r.h.s. phrase. These phrases are endocentric constructions [6], in which the rightmost phrase can be replaced for the entire phrase. Third rule will apply if *meaning-B* has a *QUALIFIER* parameter, and *meaning-A* has any parameter that is the same with *B*'s *QUALIFIER* – and it will instantiate a new *meaning-C* which is a copy of *meaning-A*, but the *QUALIFIER* replaced by the r.h.s. parameter of *B*.

Resulting set of *individual meanings* after second phase may be seen in Table 4. Last two *individual meanings* (marked with *) are deduced incorrectly due to missing medical information (“*Hemaperitonium*” is not an anatomical location).

Table 4. Set of Individual Meanings after second phase

Type	Parameter 1	Parameter 2
Conjunction	(left) Hemaperitonium	(right) mesane lümen
Location	(where) mesane lümen	(what) hematoma
Conjunction	(left) hematoma	(right) hava
Location	(where) mesane lümen	(what) hava
Location*	(where) Hemaperitonium	(what) hematoma
Location*	(where) Hemaperitonium	(what) hava

2.3 Application in Information Retrieval

Proposed algorithm generates a set of *individual meanings* from a given input sentence. Thus it extracts some of the information into structured form. In order to use these *individual meaning sets* in a retrieval scenario, a method similar to vector-space model is used.

The first step is parsing every document with the algorithm, and recording the extracted *individual meaning sets* into a database. The query strings are free text, written in a style similar to report sentences. In order to find relevant documents to a query, the query strings (which are similar to partial sentences) are assumed as a sentence on their own, and parsed with the algorithm. The resulting *individual meaning set* is the query set. Intersection of this set with the sets stored in the database is used to retrieve relevant documents.

There are various methods for deciding if a document should be retrieved based on the intersection. The first method would be checking if at least one *individual meaning* intersects between the whole document’s set and the query set. A second method is calculating the intersection with the whole document, but requiring the size of the intersection set to be greater than a threshold ratio of the search set. The third method is calculating the intersection with only a single sentence in the whole document (thus retrieving relevant sentences instead of documents), and requiring a threshold.

3 Data Driven Method

Creating hand-coded rules has its own problems such as labor and output quality. Instead, an alternative data driven method is devised. This method is constructed around dependency links.

After the availability of Turkish Treebank [11], different dependency parsers for Turkish are proposed. Eryiğit and Oflazer [3] proposed a statistical parser for unit-to-unit dependencies. Eryiğit et al. [4] developed a classifier based parser

for Turkish inside the MaltParser framework [7]. This parser is used for CoNLL-X shared task [8] for Turkish. In nearly all implementations, use of inflectional groups (IGs) as the main unit of dependency links is preferred.

3.1 Morphological Analysis

Due to its limitations, Zemberek library is found to be a poor choice for generating sufficiently detailed analyses for dependency parsing. TRMorph, a freely available two-level morphological analyzer [2] is chosen for the task. TRMorph is based on a finite state transducer framework, and uses a heavily modified version of Zemberek’s lexicon. Although various morphological analyzers exist for Turkish, currently TRMorph seems to be the only freely available one.

TRMorph is capable of extracting derivational morphemes and the resulting POS tags after each derivation. Still, the morphological analyses does not have one-to-one correspondence with outputs of other morphological analyzers used in previous works in dependency parsing of Turkish. Although this does not pose an important problem for inflectional morphemes, differences in derivational boundaries might result in incomparable results. In order to stay on common ground with previous works, a postprocessing step is added to the results of the analyzer. Some morphemes, which TRMorph does not mark as derivational (and thus does not generate new POS tags), are marked as derivational boundaries and replaced by new POS tags. This group of postprocessed morphemes consist of non-finite verb markers; such as converb markers (form subordinating clauses with adverbial function), participle markers (make non-finite verbs of relative clauses) and verbal noun markers (form noun clauses from non-finite verbs).

3.2 Morphological Disambiguation

In a highly agglutinative language like Turkish, morphemes play a very important role for the syntactic relations between words. Morphological disambiguation is more complex than assigning the correct POS tag to each word, as other morphemes would also affect the syntactic relations [4]. The number of potential combinations of morphemes is unlimited, and the numbers are still high even when calculated per inflectional group. Morphological analyzers used in previous studies cover nearly 120 different morphemes, which can be appended to words in a circular manner, producing unlimited number of inflectional groups and affixes. The possible number of combinations per inflectional group is rather limited, but still on the order of thousands [5].

Due to the relatively small training set size, purely statistical methods for disambiguation would be inappropriate for this study. Yuret and Türe [13] implement a rule learner for disambiguation of Turkish morphemes, in which a different decision list is learned per morpheme. This approach successfully overcomes the data sparseness problem. The inputs to the decision lists are the all possible string suffixes of words within a 5-word window. The decision lists are constructed by a greedy algorithm.

The decision lists act as oracles that tell whether a morpheme should or should not exist in a word (checking strings within 5 word window). Probabilities of

the provided alternative morphological analyses can then be calculated (based on their aprior probabilities in training dataset and i.i.d. assumption), and the most probably analysis can be identified.

Yuret and Türe also point that the decision lists can be used as oracles by themselves, which turns the disambiguator into an analyzer. They achieve fairly good results for this experiment in their dataset.

Although Yuret and Türe publicly provide a trained model for disambiguation, this model was inappropriate for our requirements. The morphological analyzer used by the published model was different than TRMorph's. The second reason was the content of their training set, which does not consist of any radiology reports. Considering the decision lists are structured around various string suffixes of nearby words, the available trained model would perform badly on our content which is very different.

3.3 Dependency Parsing

We used MaltParser for training and parsing using a support vector machine based model. Parameters used for parsing Turkish in CoNLL-X and CoNLL-07 tasks are publicly available, which were also used in work of Eryiğit et al. [4]. This availability provided us with the option of training the parser with our own training set. Because the CoNLL tasks were shaped around Turkish Treebank, whose size cannot be compared with our training set size, we did not attempt optimizing the parameters. Motivated by Eryiğit et al., we used IG-to-IG links for dependency parsing.

3.4 Training Set

In order to construct dependency links, a training set of limited content is used. Only abdominal reports are selected for this task. Narrowing down the topic is an important requirement because of the need for a sufficiently comprehensive training set.

Twelve reports written by different doctors are selected. Training reports are first tagged for both morphological disambiguation and dependency links. An incremental bootstrapping procedure is used for labor intensive tagging work; first only a small number (4) of the training reports are manually tagged, then the next untagged report is tagged automatically by the parser and the results are edited manually. Resulting set is again fed into the morphological disambiguator and dependency parser for generating better training models for the next reports.

The training set consists of 10 abdominal reports, 182 sentences and 1415 words. There are 136 unique sentences and 462 unique morphological analyses.

3.5 Retrieval with Dependency Links

The outlined method generates a directed dependency graph for a given sentence. These dependency links can be seen as analogous to the *individual meanings* of the first method.

In application, all sentences in test reports are morphologically disambiguated and then analyzed for dependencies. The resulting dependency graphs are recorded in a database.

For retrieval, the query (which is written in a similar style with report sentences) is treated as a sentence of its own and gets disambiguated and parsed. The resulting dependency links are searched within the recorded links. Only the roots of the morphological analyses and direction of dependency links are compared, types of dependencies are ignored. The dependencies are compared a projective fashion: If in the search query there is a dependency link (*modifier B, head A*), and in the test set there is a sentence with two dependency links (*modifier B, head C*), (*modifier C, head A*), this is counted as an intersection. Projections are only made in documents, in queries only direct dependencies are used.

As in the first method, mapping intersections to relevant documents can be applied in different ways. These are:

- counting only a single intersection within the whole report as relevant,
- requiring the intersection set’s size to be greater than a threshold (preferable percentage of number of dependency links in query), but allowing different links to be in different sentences within the same report,
- requiring the intersection set’s size to be greater than a threshold and forcing all links to be in the same sentence.

4 Experimental Results

We tested the methods mentioned in two test sets. The first test set consists of radiological reports on different topics. There are 53 reports and 100 queries in this set, forming 5300 data points. The results contain all three alternatives of the first method’s application, compared to a baseline retrieval algorithm.

The baseline algorithm performs stemming by extracting root lexemes of input words (either query or document), and checks for intersection of these root lexemes with the document. Similar to the proposed methods’ applications, different thresholds on the size of intersection set is tested for the baseline algorithm.

Table 5 contains comparison of baseline algorithm’s and rule based algorithm’s results. Although the baseline algorithm achieved very high recall in all cases, its precision is close to zero. The reason lies in the structure of the queries. As a

Table 5. Evaluation results for rule based method

Method	Match target	Threshold	Recall	Precision	F_1 Score
Baseline	Whole report	Single match	96.05%	5.15%	9.78%
Baseline	Whole report	50%	96.05%	5.42%	10.26%
Baseline	Single sentence	50%	96.05%	5.46%	10.33%
Rule Based	Whole report	Single match	67.10%	28.65%	40.15%
Rule Based	Whole report	50%	65.78%	31.84%	42.91%
Rule Based	Single sentence	50%	65.78%	31.84%	42.91%

query string is a partial sentence written in a similar style to report sentences, a query string contains all the words that would be existant in a target sentence. The reason for very low precision is that most of the query strings contain generic words like *cyst*, *liver*, *lesion*. Most of these words exist in all sentences, even if they are not related together. On the other hand, proposed algorithm focuses on the relations of individual words, and makes a compromise between very low precision and very high recall. As the query strings mostly contain one or two relations, threshold setting does not affect the results of rule based algorithm as much as anticipated. This is also the reason why more threshold values were not tested.

The second test set consists of only abdominal radiological reports. There are 10 reports and 28 queries. In this test set, the performances of the rule based method and data driven method are compared.

Table 6. Evaluation results for second test set

Method	Match target	Threshold	Recall	Precision	F_1 Score
Rule based	Whole report	Single match	71%	66%	68%
Dependency based	Whole report	Single match	89%	62%	73%
Dependency based	Whole report	50%	78%	81%	79%

The results of the second test set may be found in Table 6. With a threshold of a single match, the data driven method achieved fairly higher recall than the rule based method, with a slight decrease in precision. When a threshold value of 0.5 is utilized, the data driven method achieved a better recall than the rule based method (although smaller than the previous comparison), and its precision increased significantly. Data driven method resulted in better F_1 scores in all cases. This was expected, as data driven method is more flexible than the rule based method.

5 Conclusion and Future Work

In this work we presented two methods for information retrieval from Turkish radiology reports. Turkish radiology reports pose different problems than western counterparts due to the nature of Turkish language. We proposed a rule based method and an analogical dependency link based method. We compared their performances with a baseline search engine that checks if query words' root lexemes exist in the reports, with a threshold on intersection set size.

In our implementation, one of the greatest challenges was morphologically analyzing root words that have not been encountered before. Doctors' differences on spelling of Latin based words in Turkish aggravated this problem. In the first method, we devised a workaround so that even if morphological analyzer fails for some words, the processing can continue with string suffixes. In the second method, we planned to use the morphological disambiguator as an analyzer in

these cases, but failed due to the small size of the training set of disambiguator. If the training set size for morphological disambiguator was large enough, these problems could easily be circumvented.

Our initial observation was that, a person with no medical knowledge can still partially understand Turkish radiology reports. We avoided using any medical ontologies or lexicons, in order to test the limits of this observation in isolation of other factors. Although current work is limited to information retrieval context, it is planned to extend the base idea into information extraction context.

Acknowledgements. This study is supported by Turkish Scientific and Technological Research Council (TÜBİTAK) project 3080179.

References

1. Akin, A.A., Akin, M.D.: Zemberek, an open source NLP framework for Turkic Languages (2007)
2. Çöltekin, Ç.: A freely available morphological analyzer for Turkish. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), European Language Resources Association (ELRA), Valletta (2010)
3. Eryiğit, G., Oflazer, K.: Statistical dependency parsing of Turkish. In: Proceedings of the 11th EACL, Trento, April 3-7, pp. 89–96 (2006)
4. Eryiğit, G., Nivre, J., Oflazer, K.: Dependency parsing of Turkish. *Computational Linguistics* 34(3), 357–389 (2008)
5. Hakkani-Tür, D., Oflazer, K., Tür, G.: Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities* 36, 381–410 (2002), <http://dx.doi.org/10.1023/A:1020271707826>, doi:10.1023/A:1020271707826
6. Nivre, J.: Dependency grammar and dependency parsing. Tech. rep., Växjö University (2005)
7. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering Journal* 13(2), 99–135 (2007)
8. Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., Marinov, S.: Labeled pseudo-projective dependency parsing with support vector machines. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X), New York, NY, June 8-9, pp. 221–225 (2006)
9. Oflazer, K.: Two-level description of Turkish morphology. In: Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics, EACL 1993, pp. 472–472. Association for Computational Linguistics, Stroudsburg (1993), <http://dx.doi.org/10.3115/976744.976810>
10. Oflazer, K., Hakkani-Tür, D.Z., Tür, G.: Corpus annotation for parser evaluation. In: Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC), pp. 35–41 (1999)
11. Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G.: Building A Turkish Treebank (2003)
12. Taira, R.K., Soderland, S.G., Jakobovits, R.M.: Automatic Structuring of Radiology Free-Text Reports. *Radiographics* 21(1), 237–245 (2001), <http://radiographics.rsna.org/content/21/1/237.abstract>
13. Yuret, D., Türe, F.: Learning Morphological Disambiguation Rules for Turkish. In: Proceedings of HLT-NAACL (2006)

Discovering and Analyzing Multi-granular Web Search Results

Gloria Bordogna¹ and Giuseppe Psaila²

¹ CNR-IDPA - National Research Council Institute for the Dynamics of Environmental Processes

via Pasubio 5, I-24044 Dalmine (BG), Italy

gloria.bordogna@idpa.cnr.it

² Università di Bergamo - Faculty of Engineering,

via Marconi 5, I-24044 Dalmine (BG), Italy

psaila@unibg.it

Abstract. In this paper, we propose an approach based on the use of soft aggregation operators and multi-granular graphs for discovering and analyzing the results of Web searches, organized into granules of distinct resolution. This practice enables user-driven explorations of the topics retrieved in a search process on the Internet, being based on the graphical representation of both the information granules, and their discovered relationships. We further present the application of both the soft operators and multi-granular graphs within the meta-search system *Matrioshka*, and discuss their semantics and usefulness.

Keywords: Clustered Web search results, soft aggregation operators, Web exploration, graph-based representation.

1 Introduction

The motivations of this work are related to the so called "ranked list problem", i.e., the inadequacy of the usual ranked list to present results retrieved by search engines. In fact, with the rapid growth of the Web, users' queries have become more ambiguous than ever [9]: entries that in the past were associated with a unique meaning, now have diversified semantics: for example, ambiguities of personal names is a big problem. This situation calls for alternative modalities of visualization and interaction with search results whose objective is improving the potential exploitation and comprehension of retrieved contents.

Clustering has been proposed as the most promising alternative to the ranked list [6,14]. It organizes the results of a single search into groups that exhibit high intra-similarity and low inter-similarity. The advantages of results clustering are related to the more synthetic view of the retrieved contents that they offer, where each cluster represents homogeneous sub-topics of a query.

The community of Information Retrieval has devoted strong efforts to propose effective clustering algorithms, and commercial search engines are available that provide clustered results such as *clusty* (www.clusty.com) and *carrotsearch*

(carrotsearch.com). Despite these efforts, the current clustering approaches are still inadequate, since they do not take into account the perspective of users in forming the clusters, and still do not provide concise, comprehensive and expressive labels to the user [5].

In [1] we first proposed a different approach to improve the exploitation of clustered results: we did not face the problem by providing new clustering algorithms, but we proposed a novel methodology for a *user-driven discovery and exploration of the cluster contents*, based on both the use of manipulation operators of the clusters, the automatic generation of disambiguated queries suggestions from clusters [3], and cluster re-ranking facilities [2]. All these features have been defined within the project named *Matrioshka*, and implemented in the homonymous prototypal system (<http://matrioshka.unibg.it>) that has undergone a user's evaluation reported in [3]. To our knowledge, no similar proposal can be found in the literature.

Specifically, in this paper, we go a step further in this direction by proposing both *soft operators* for combining clusters, that allow users to discover and reveal the hidden shared topics retrieved by multiple Web searches, and *multi-granular graphs*, for representing and analyzing the discovered topics.

In the literature there are some proposals of using graphs to represent the result of a single Web search, such as the *wonderwheel* feature recently introduced by *Google*, or even tree structures obtained by applying hierarchical categorization or clustering to the results of a single search such as in the *Open Directory Project* and in *clusty*.

Our approach is different for several aspects.

- First of all, we generate graphs to represent and analyze the contents yielded by multiple searches in a search process and not simply by a single search.
- Second, graphs represent the contents and their relationships at distinct levels of granularity (general overview, intermediate view, detailed view).
- Finally, since the retrieved Web pages are represented by their snippets, that focus on the subpart of the Web page containing the query terms, the graphs represent the retrieved contents and their relationships from the perspective of the user needs. So, since a Web page can be retrieved by distinct queries, multiple representations of the same Web page are derived, each of them focused on a distinct point of view.

The *soft operators* used to discover the hidden relationships between pairs of information granules are defined based on fuzzy set theory. These operators provide users with some practical means to explore the overall set of results of several Web searches, to filter out their shared documents and shared topics, correlated documents and correlated topics. The highlighting of the relationships between documents retrieved by distinct queries, can give new hints on their relevance. However, it is not possible to exploit the full content of the documents retrieved by the query, because that would require an HTTP access to the Web pages reported in the result lists; consequently, our solution extracts representative terms from within the texts (titles and snippets) associated to each URL reported in the result lists provided by the search engines and expands the

representative terms with related terms in a thesaurus (in the implemented system the expansion was performed by using *Wordnet*).

In the paper, we first overview the related literature; second, we recap the information granules that constitute the basis for our framework; then, we introduce the soft operators used to combine the information granules in order to identify documents sharing common topics, that otherwise would remain hidden. Finally, we describe the multi-granular graphs and an example of their application within the meta-search system *Matrioshka* [2], for the exploration of the results of a search process.

2 Related Works

Our proposal is related to three research fields: multi-document summarization, graph-based representations of search results, and graph mining.

Multi-document summarization is aimed at extracting information from multiple documents about the same topic [8]. Most of the techniques are independent of the query that retrieved the documents, and follow two alternative approaches: either they simply extract relevant passages from the set of documents by applying statistical analysis independently of the query that retrieved the documents, or they employ *Natural Language Processing*. A multi-document summary is a phrase or a set of keywords that represents the topics of the set of documents.

Our approach can be regarded as a multi-document summarization task, since we also generate clusters by partitioning the results of a Web search in homogeneous sets and represent the clusters' contents by synthetic labels. Nevertheless, as the features of the documents used for clustering are extracted from the documents' dynamic snippets, that contain the query terms, the generated clusters' labels are query dependent. This permits the approximate synthesis of cluster contents focused on the context of user's needs. Further, not only the topics dealt with in clusters are identified, but also the shared sub-topics between each pair of clusters. In the same manner, also the topics in both each retrieved Web page and each retrieved list, and the shared sub-topics between pairs of both Web pages and retrieved lists are identified. To this aim, we combine the ranked lists, the clusters and the items in the ranked lists through "*intersection*" operations. This permits to discover hidden topics, not necessarily relevant in any single cluster, nor top-rank in any single ranked list, that are relevant to the global search process the user is engaged in, since they represent results and contents shared by multiple Web searches.

Graph-based representations of search results have been proposed for distinct purposes, such as for presenting the results of clustering and categorization [16,12], for representing syntactic relationships between words/phrases in documents [10] and for representing the outcome of Web searches [11]. The essential idea is to model a network of multimedia documents as a graph.

Our proposal takes inspiration from these approaches: ranked lists, clusters, single Web pages, as well as their inter-relationships, are represented by multi-granular graphs. These graphs constitute a graphical overview, like a summary, of the main retrieved topics. The multi-granular graphs permit the analysis of the

relationships between topics of distinct levels, both in a top-down fashion, i.e., from the coarsest representation to the most detailed one, and in a bottom-up fashion.

Here comes the connection with *graph mining*, that consists in exploring the nodes of a graph to discover the content they represent [15]. In our framework, the relationships between topics and documents can be discovered: a user can get documents dealing with a topic represented by a node, or can get information about a *hidden* topic represented by an edge between pairs of nodes, simply by expanding nodes and edges.

3 Multi-granular Organizations of Web Search Results

In our framework, the information granule of finest resolution is the *ranked item*, representing a document in a ranked list retrieved by a search engine as a result of a query evaluation. A ranked item i is defined by a 5-tuple

$$i : \langle Uri_i, Title_i, Snippet_i, Bag_i, Irank_i \rangle$$

Uri_i is the *Uniform Resource Identifier* of the ranked Web document. $Title_i$ and $Snippet_i$ are, respectively, the document title and snippet.

$Bag_i = \{s_1/w(s_1), \dots, s_n/w(s_n)\}$ is a fuzzy set of strings s_j (single terms) each one weighted with a score $w(s_j) \in [0, 1]$ expressing the significance of the string in representing the contents of the item i . In the rest of the paper, $Bag_i[s]$ is a short notation for $w(s)$, where $s \in Bag_i$. We assume that $Bag_i[s] = 0$ when $s \notin Bag_i$ (see Section 4 for the procedure to generate it). Finally, $Irank_i$ is a score in the range $[0, 1]$ that expresses the estimated relevance of the retrieved document w.r.t. the query and is computed as detailed in Section 4. This is the reason why an item is named *ranked item*.

The intermediate information granule is the *cluster*, that is composed of items and has a rank as well. A cluster c is defined by the 3-tuple:

$$c : \langle Content_c, Label_c, Crank_c \rangle$$

$Content_c$ is the set of items associated with the cluster. $Label_c$ is a set of terms that semantically synthesize the main contents of the cluster; generated by function *SynthLabel* defined in [1], i.e., $Label_c = SynthLabel(Content_c)$. $Crank_c$ is a measure of the relevance of the cluster, in the range $[0, 1]$.

A cluster can be generated by applying an operator combining two other clusters, or by a clustering operation applied to a query result ranked list.

A *group* is the coarsest information granule, composed of clusters. A group g is defined as follows.

$$g : \langle Clusters_g, Label_g \rangle$$

$Clusters_g$ is the set of clusters belonging to the group. A group can be obtained from the ranked list of documents retrieved by a search engine, or can be generated by an operator working on groups [1]. $Label_g$ is a set of terms that semantically synthesize the main contents of the group; in the case of a group generated by a query to a search engine, it is the text of the query submitted to the search engine, otherwise it is the title of the ranked item most representative of the group, as defined in [1].

4 Soft Operators to Combine Granules of Search Results

An immediate way to generate a group of ranked items is to perform a Web search by submitting a query q to a search engine SE . Further, the N top ranked items in the list of retrieved results are clustered, by applying a clustering algorithm. A *group* of ranked clusters containing the items is thus generated.

Here, we do not focus on the clustering technique. It is worth noticing that we do not need to access the text of the documents for extracting the features necessary to cluster them. We parse the result list provided by the search engine, containing the first N results, and extract all the information that constitutes the representation of a ranked item. In the implemented system *Matrioshka* [2], the *Lingo* clustering algorithm [13] is used, and the *label* of the cluster is the *title* of the ranked item which is the *most relevant* in the cluster [1].

We are aware that the effectiveness of the method also depends on the clustering algorithm. Nevertheless, the methodology we propose to combine clusters can aid in better understanding the contents of clusters, thus complements the information provided by the clusters labels.

Notice that the same Web page, retrieved by different search engines (or by different queries), may be represented by distinct items in distinct result lists. In fact, in this case the document is uniquely identified by the same Uri_i , while it may have distinct $Title_i$, $Snippet_i$, Bag_i and $Iranks_i$. For this reason, we compute $Iranks_i$ as a function of the position of the item in the query result list, independently of the actual relevance score computed by the search engine.

On the other side, distinct Web pages have distinct $Uris$ but may share the same or similar *titles* and *snippets*, because they are indeed duplicated documents at distinct Web sites retrieved by the same query.

The strings in Bag_i are obtained by performing lexicographic analysis of the Uri_i , $Title_i$ and $Snippet_i$ of the items by removing stop-words, conflating terms having the same stem, expanding single terms with associated terms by using *Wordnet* [7]; then, all the selected single terms in $Title_i$ and $Snippet_i$ are included in the bag of strings. Each string s in Bag_i is weighted by its relative frequency $w(s)$: an occurrence in the title is considered as two occurrences in the snippet and Uri , and the total number of occurrences of a string is then normalized w.r.t. the sum of all weights of all strings $s \in Bag_i$ so that $w(s) \in [0, 1]$.

Operations between Sets of Items

The *intersection* and *union* operations between content of clusters are defined in two different ways. To filter documents retrieved by two different search engines, that have different snippets and bags but the same Uri , we first introduced in [1] the ranked intersection, $RIntersection$, and the ranked union, $RUnion$, as the usual intersection and union of fuzzy sets of ranked items, where a ranked item is uniquely identified by its Uri (that is compared based on an exact matching), and its membership degree is the *Iranks*. The $RIntersection$ and $RUnion$ combine two fuzzy sets of ranked items and yield a new fuzzy set of ranked items. Each item in the result represents two distinct ranked items in the operands having the same Uri ; in case of $RIntersection$ (resp. $RUnion$), its *Iranks* is the minimum (resp. maximum) of the *Iranks* of the input items. The *Title*, the *Snippet* and

the *Bag* of the resulting items are those of the input ranked item having the minimum (in the case of *Ranked Intersection*) or the maximum (in the case of *Ranked Union*) value of *Iranks*, without making any change. The rationale of this choice is the fact that, in the aggregation based on the intersection (resp. union), we want to represent the document by its worst (resp. best) representative, in accordance with the modeling of the AND and the OR within fuzzy set theory.

Nevertheless, it can happen that the same Web page is duplicated at distinct sites, or its contents are near duplicates, so two Web pages may differ just for their *Uris*, while they may have very similar *Title*, *Snippet* and *Bag*. The crisp ranked intersection filters out duplicated Web pages from the results. In particular situations, this could be a limitation, since one would like to identify ranked items dealing with similar and duplicated topics. Let us consider, for example, two pages of *Expedia* that refer to the same hotel, but have different *Uris*, because they refer to two different booking dates. The *RIntersection* operator, does not consider these documents as the same document, even if their semantics is the same.

This is the rationale for the introduction of the *soft operators*, that is the second way we define intersection and union between (content of) clusters [4]. The soft intersection, *SIntersection*, and the soft union, *SUnion*, are named *soft* because they identify the ranked items uniquely by their bags of weighted strings, i.e., fuzzy subsets of strings. A fuzzy relation between any two items can be defined as for two fuzzy sets in order to perform a partial matching of pairs of ranked items. Thus *SIntersection*, and *SUnion*, are defined as the intersection and union of fuzzy sets of fuzzy sets.

In order to find out duplicated documents, the *Soft Intersection* between clusters can be applied. It yields a set of items *C* generated considering the shared contents between the bags of the input ranked items. Then, it identifies topics that approximately represent the content shared by two input clusters.

Definition 1: Fuzzy Inclusion Relation. Given two bags *Bag*₁ and *Bag*₂, the *Fuzzy Inclusion Relation* $\subseteq_F (Bag_1, Bag_2)$ is defined as:

$$\subseteq_F (Bag_1, Bag_2) = \frac{\sum_{s \in Bag_1} \min(Bag_1[s], Bag_2[s])}{\sum_{s \in Bag_1} Bag_1[s]} \quad \square$$

This measure is the inclusion degree of *Bag*₁ within *Bag*₂. It gets zero when the bags of strings *Bag*₁ and *Bag*₂ do not have any common string; it gets the maximum value 1 when all the strings in *Bag*₁ are also present in *Bag*₂ with greater weight; it gets intermediate values when the two bags of strings have some common string.

Definition 2: Soft Intersection. The operation *SIntersection*, denoted by \cap^S , performs the *content intersection* of two sets of items (cluster contents) *C*₁ and *C*₂ of ranked items.

To define it, we represent a ranked item *i* uniquely by its bag of strings *Bag*_{*i*} and not by its *Uri*_{*i*}; then, a partial matching of any pairs of Bags of the items in the two input sets of items is evaluated, to generate the resulting set of items.

$C = SIntersection(C_1, C_2) = \cap^S(C_1, C_2) = (C_1 \cap^R C_2) \cup^R StrongEntail(C_1, C_2)$ where *StrongEntail*(*C*₁, *C*₂) is based on \subseteq_F as follows (with $\epsilon_\cap \in [0, 1]$):

For any pair of items $i_1 \in C_1$, $i_2 \in C_2$, with bags Bag_1 and Bag_2 , resp.

1. if $Uri_1 \neq Uri_2 \wedge i_1, i_2 \notin \cap^R(C_1, C_2)$
then
 2. if $[\subseteq_F (Bag_1, Bag_2) \geq \subseteq_F (Bag_2, Bag_1)] \wedge [\subseteq_F (Bag_1, Bag_2) > \epsilon_\cap]$
 3. then $i_1 \in StrongEntail(C_1, C_2)$
 4. if $[\subseteq_F (Bag_2, Bag_1) > \subseteq_F (Bag_1, Bag_2)] \wedge [\subseteq_F (Bag_2, Bag_1) > \epsilon_\cap]$
 5. then $i_2 \in StrongEntail(C_1, C_2)$

The condition at Line 1. imposes that, in order to include an item $i_1 \in C_1$ in the set returned by $StrongEntail(C_1, C_2)$, it must be not already present in the ranked intersection. Then, item i_1 is included in the result (Line 3.) if there exists another item $i_2 \in C_2$ in the other cluster that includes i_1 at least to the degree ϵ_\cap (Line 2); the higher the value of ϵ_\cap , the stronger the chosen item entails the other. Symmetrically, item i_2 is included (at Line 5.) if item i_1 includes i_2 at least to the degree ϵ_\cap (Line 2). Thus, the resulting item is chosen as the more specific item between i_1 and i_2 . In the case in which $\subseteq_F (Bag_1, Bag_2) = \subseteq_F (Bag_2, Bag_1) > \epsilon_\cap$, i_1 and i_2 are very similar as far as their contents and specificity and so any of the two can be chosen (we chose i_1). \square

The soft intersection thus relaxes the constraint of the ranked intersection. It generates a set of items C , that contains all ranked items i resulting from the ranked intersection of the input sets (clusters) C_1 and C_2 plus other ranked items belonging to either C_1 or C_2 , that share a minimum percentage of common contents, on the basis of their bags of strings. This way, in the result we retain the ranked items which have the most specific contents, as it can be guessed from the bag of strings.

Let us explain the rationale of this definition with a simple example. Given two documents, one dealing with *Italian tourist places*, and the second with *Tourist places in the Mediterranean countries*; they probably share most of the places listed in the first document, since Italy is a Mediterranean country, but the vice versa is unlikely to occur, since the second document contains also places of other countries such as Greece, Spain and so on. So, the soft intersection retains only the shared contents, i.e., the first document on Italian places.

Definition 3: Soft Union. The operation $SUnion$, denoted by \cup^S , performs the content union of two sets of ranked items (contents of clusters) C_1 and C_2 .

To define it, we uniquely represent a ranked item i by its bag of strings Bag_i ; by evaluating a partial matching of any pairs of $Bags$ of the items in the two input sets, the resulting set of items is generated.

$$C = SUnion(C_1, C_2) = \cup^S(C_1, C_2) = (C_1 \cap^R C_2) \cup^R LooselyEntail(C_1, C_2)$$

where $LooselyEntail(C_1, C_2)$ is based on \subseteq_F as follows (with $\epsilon_\cup \in [0, 1]$):

For any pair of items $i_1 \in C_1$, $i_2 \in C_2$, with bags Bag_1 and Bag_2 , resp.

1. if $Uri_1 \neq Uri_2 \wedge i_1, i_2 \notin \cap^R(C_1, C_2)$
then
 2. if $[\subseteq_F (Bag_1, Bag_2) \leq \subseteq_F (Bag_2, Bag_1)] \wedge [\subseteq_F (Bag_1, Bag_2) > \epsilon_\cup]$
 3. then $i_1 \in LooselyEntail(C_1, C_2)$

- 4. if $[\subseteq_F (Bag_2, Bag_1) < \subseteq_F (Bag_1, Bag_2)] \wedge [\subseteq_F (Bag_2, Bag_1) > \epsilon_U]$
- 5. then $i_2 \in LooselyEntail(C_1, C_2)$

The soft union restricts the ranked union. It generates a set of items C , that contains all ranked items i resulting from the ranked intersection of the input sets C_1 and C_2 , plus the ranked items belonging to either C_1 or C_2 , that share a minimum percentage of common contents, on the basis of their *Bags*. This way, in the result we retain the ranked items which have the most general content, as it can be guessed from the *Bags* of strings. The lower the value of ϵ_U , the more general the chosen item w.r.t. the other one. If $\subseteq_F (Bag_1, Bag_2) = \subseteq_F (Bag_2, Bag_1) > \epsilon_U$, i_1 and i_2 share the same percentage of contents above ϵ_U , so we can choose any of the two items (we retain i_1). \square

Let us give an example. Assume that we want to have a panoramic overview of the Mediterranean Tourist information; having two documents, one dealing with *Italian tourist places*, and the second with *Tourist places in the Mediterranean countries*, the second one is more general and thus it is selected.

Operators between Groups

Based on the two families of operators for sets of items so far introduced, i.e., ranked intersection and ranked union on one side, soft intersection and soft union on the other side, we can define operators for manipulating groups. Recall that groups of clusters are the coarsest granules of information in our framework. Here, we just define the basic ones used to identify shared documents and topics.

Definition 4: Group Intersection Operators. The *group ranked intersection* operator and the *group soft intersection* operators are defined so as to exploit the (crisp) *ranked* intersection and the *soft* intersection between all the pairs of clusters originated from the two input groups. Given two groups of clusters g_1 and g_2 , both the *group ranked intersection* operator \cap^{GR} and the *group soft intersection* operator \cap^{GS} , hereafter indicated simply by \cap^G , are defined as:

$$\cap^G : G \times G \rightarrow G \quad \cap^G(g_1, g_2) \rightarrow g$$

where g is the resulting group. For each pair of clusters c_1, c_2 , such that the intersection of their contents (either ranked \cap^R or soft \cap^S , hereafter indicated simply by \cap) is not empty (i.e., $c_1.Content \cap c_2.Content \neq \emptyset$), there is a cluster $c \in g$ defined as follows:

- $c.Content = c_1.Content \cap c_2.Content$
- $c.Label = SynthLabel(c.Content)$
- $c.Crank$ is the average of $Iranks_i$ for items $i \in c.Content$. \square

Definition 5: Group Join Operators. The *group ranked join* operator and the *group soft join* operators are defined so as to exploit the (crisp) *ranked* union and the *soft* union between all the pairs of clusters within the two input groups. Given two groups of clusters g_1 and g_2 , both the *group ranked join* operator \bowtie^{GR} and the *group soft join* operator \bowtie^{GS} , hereafter indicated simply by \bowtie^G , are defined as follows:

$$\bowtie^G : G \times G \rightarrow G \quad \bowtie^G(g_1, g_2) \rightarrow g$$

where g is the resulting group. For each pair of clusters c_1, c_2 , such that the intersection of their contents (either ranked \cap^R or soft \cap^S , hereafter indicated

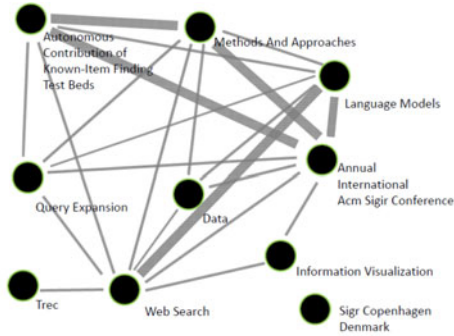


Fig. 1. Clusters graph of the query “*Proceedings SIGIR*” submitted to *Google Scholar*

simply by \cap) is not empty (i.e., $c_1.Content \cap c_2.Content \neq \emptyset$), there is a cluster $c \in g$ defined as follows (where \cup denotes either the ranked union \cup^R or the soft union \cup^S):

$$c.Content = c_1.Content \cup c_2.Content$$

$$c.Label = SynthLabel(c'.Content)$$

$$c.Crank \text{ is the average of } Irank_i \text{ for items } i \in c.Content. \quad \square$$

5 Discovering and Analyzing Search Results

Matrioshka is a meta-search system (<http://matrioshka.unibg.it>) designed and implemented to perform user-driven explorations of the results retrieved in a search process. Among its functionalities, it permits to submit queries by choosing among four search engines (*Google*, *Google Scholar*, *Yahoo! API!*, *Bing*); the result lists are clustered and clusters are stored into groups.

It permits the manipulation of groups by the application of the group operators. The discovery of the hidden topics can be performed by interacting with *Matrioshka* interface, by analyzing the clusters labels, the documents within the clusters, and the contents of the groups. It is possible to generate new groups by coalescing, reclustered and combining the groups through the operators.

A user’s validation of this approach was performed and the results are reported in [3] in which the user explicitly applies the manipulation operators. This evaluation consisted in designing an experiment in which users submit the same query firstly directly to a search engine among *Google*, *Google Scholar*, *Yahoo! API!*, *Bing*, and secondly indirectly through the *Matrioshka* system. The evaluation metrics were defined to quantify the user’s gain in terms of both precision and effort in retrieving relevant documents by means of *Matrioshka* w.r.t. using directly the search engine. The results obtained pointed out the need to introduce some automatic mechanism that helps users apply the manipulation operator and to visualize the discovered relationships between pairs of groups, clusters, and documents. This is the motivation that brought us to define the *multi-granular graph visualization of search results*.

Thus, the utility *Graph*, implemented in *Matrioshka*, provides an alternative way for exploring the shared topics between distinct groups. *Graph* automatically performs the *group ranked intersection* and the *group soft intersection* of each pair of groups selected by the user within the ones previously obtained in carrying on the search process. Finally, the utility *Graph* represents the obtained results and the original selected groups in terms of labeled multi-granular graphs. There are three layers, one for each type of information granules.

- **Group Graph.** In this level, each node corresponds to a *group*. A group is the overall set of documents, organized into clusters that has been retrieved by a single query, or produced by the application of a group operator. This is the highest overview of retrieved results.
- **Cluster Graph.** Each node corresponds to a *cluster* of documents; this is the intermediate overview. Figure 1 shows the cluster graph of the group generated by searching “*Proceedings SIGIR*” with *Google Scholar*. A menu “*Option*” allows users to choose the type of edges to visualize, i.e., the type of relationship that one wants to analyze. The graph in Figure 1 is generated with the *RIntersection* option, which means that the shown edges are computed by applying the ranked intersection, and thus highlight the existence of shared Web pages between the connected clusters. Indeed, the clustering algorithm generates an overlapped partition of Web pages.
- **Document Graph.** Each node corresponds to a *document*; this is the most detailed representation of the Web results.

It is possible to show two kinds of edges in a graph:

- **Crisp edge.** Depicted as a thick edge, it reveals the existence of same Web pages in the two connected nodes (either groups, clusters or documents); these common Web pages are identified by the application of *ranked intersection* \cap^R to the connected nodes.
- **Soft edge.** Depicted as a thin edge, it reveals the existence of similar Web pages in the two connected nodes, i.e., Web pages sharing the topic expressed by the edge label. The similar documents are identified by the application of the *soft intersection* \cap^S to the connected nodes.

Figure 1 shows the cluster graph for the group obtained from query “*Proceeding SIGIR*”; there are both crisp edges (the thick ones) and soft edges (the thin ones). Crisp edges in Figure 1 denote that the clustering applied to the result of the query “*Proceeding SIGIR*” generated an overlapping partition in which some Web pages belong to several clusters.

Figure 2 is the document graph of two clusters among the ones shown in Figure 1; square objects are documents, circular objects are the two selected clusters (dashed lines connect documents to the cluster they belong to); the thick edge in the top denotes that the two connected documents have the same *Uri*; all the other thin edges are soft edges. It can be seen that the number of the soft edges is much greater than the number of crisp edges.

Let us give an example of exploratory analysis by the use of the soft operators. We submitted two queries, “*Proceeding SIGIR*” and “*Proceeding ECIR*”

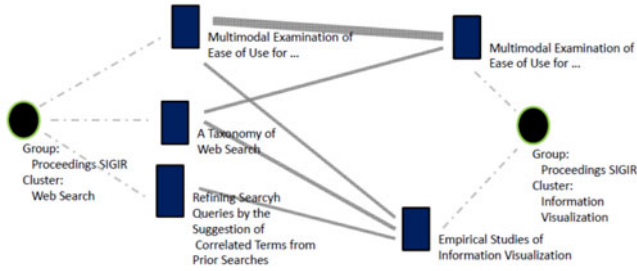


Fig. 2. Fragment of the Document Graph of the search “*Proceedings SIGIR*” submitted to *Google Scholar* with soft edges highlighting documents with shared topics.

to *Google Scholar* through *Matrioshka*. Then, we performed a *ranked intersection* of the obtained groups: an empty group was generated, revealing that the two lists do not have any common Web page. Then, we tried a soft intersection with threshold $\epsilon_{\cap} = 0.5$ between the same two groups, and a group entitled “*Proceeding information ACM Retrieval Conference*” was generated with several clusters. The label of the group identifies the shared common topic between the two original groups. In fact, the original queries asked for the proceedings of two ACM conferences on information retrieval (*ACM SIGIR* and *ACM ECIR*) and the resulting group indeed expresses their common topic. Finally, we applied the soft join operator between the two original groups to generate a group of correlated documents. We set $\epsilon_{\cup} = 0.5$: analyzing the resulting group, we discover in the same cluster a few documents that are near duplicate, with different *Uri* but more or less the same content.

From the analysis of the user’s behavior in interacting with *Matrioshka*, we observed that the soft intersection operator is the one that is most frequently used in conjunction with the coalescing and re-clustering operators, that permit to collapse all clusters of a group into a single cluster, and to re-cluster the content of a group, respectively. From the analysis of the user’s interaction with the multi-granular graphs, we observed that the most frequently visualized graphs are the ones with soft edges. This is because one useful feature of these graphs is to permit the discovering of similar Web pages, which can be interesting in several contexts. For example, in order to discover plagiarism, the user has to perform several operations: send several queries to search engines and combine the results, in order to find out the documents that are more correlated, based on their content. Then, after coalescing and re-clustering, the user can exploit the utility *Graph* to analyze actual correlation of documents and, by clicking on two strongly correlated documents, can read them and possibly discover plagiarism.

6 Conclusions and Future Work

In this paper, a user-driven methodology to discover and explore clustered results obtained in a Web search process by possibly querying several search engines has been proposed. This paradigm is based on the application of soft operators to

combine pairs of information granules of distinct resolution (groups, clusters and single documents) that can highlight their common and similar documents.

The results of these operators are visualized in the form of multi-granular graphs where the nodes identify information granules and the edges their shared topics, respectively. The user can select the desired type of relationship, i.e., the type of edge, to visualize on the current displayed graph: now the choice is limited to *Rintersection* and *Sintersection*. We are extending the possible choices of this menu, by including other operators, associated with distinct semantics of the relationships between nodes. We also plan to include new soft operators to highlight other types of relationships between pairs of information granules.

References

1. Bordogna, G., Campi, A., Psaila, G., Ronchi, S.: A language for manipulating clustered web documents results. In: Proceedings of CIKM 2008, Int. Conf. on Information and Knowledge Management, pp. 23–32 (2008)
2. Bordogna, G., Campi, A., Psaila, G., Ronchi, S.: A cluster manipulation paradigm for mobile web search interaction. In: Proceedings of IIR-2010, 1st Italian Workshop on Information Retrieval, Padova, Italy (January 2010)
3. Bordogna, G., Campi, A., Psaila, G., Ronchi, S.: Disambiguated query suggestions and personalized content-similarity and novelty ranking of clustered results to optimize web searches. Information Processing and Management (in press, 2011)
4. Bordogna, G., Psaila, G.: Soft operators for exploring information granules of web search results. In: Proceedings of WCSC-2011, 1st World Congress on Soft Computing, San Francisco, CA, USA (May 2011)
5. Carpineto, C., Osinski, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Computer Survey 41(3), 1–38 (2009)
6. de Graaf, E., Kok, J., Kusters, W.: Clustering improves the exploration of graph mining results. In: Artificial Intelligence and Innovations 2007: from Theory to Applications. IFIP, vol. 247, pp. 13–20. Springer, Heidelberg (2007)
7. Fellbaum, C.: WordNet: an Electronic Lexical Database. The MIT Press, Cambridge (1998)
8. Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: Proceedings of NAACL-ANLP-2000, the 2000 NAACL-ANLP Workshop on Automatic Summarization, pp. 40–48. ACL, Seattle (2000)
9. Jansen, B.J., Spink, A.: How are we searching the world wide web? a comparison of nine search engine transaction logs. Information Processing and Management 43, 248–263 (2006)
10. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of MMIES-2008, Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 17–24. Association for Computational Linguistics, Manchester (2008)
11. Liu, Y., Zhang, M., Ma, S., Ru, L.: User browsing graph: Structure, evolution and application. In: Proceedings of WSDM 2009, 2nd Int. Conf. on Web Search and Web Data Mining, Barcelona, Spain (February 2009)
12. Markov, A., Last, M., Kandel, A.: Fast categorization of web documents represented by graphs. In: Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B. (eds.) WebKDD 2006. LNCS (LNAI), vol. 4811, pp. 56–71. Springer, Heidelberg (2007)

13. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems* 20, 48–54 (2005)
14. Roussinov, D.G., Chen, H.: Information navigation on the web by clustering and summarizing query results. *Information Proc. and Manag.* 37, 789–816 (2001)
15. Schenker, A.: Graph-theoretic techniques for web content mining. PhD thesis, Tampa, FL, USA (2003)
16. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of web documents using a graph model. In: *Proceedings of ICDAR-2003, Int. Conf. on Document Analysis and Recognition, Los Alamitos, CA, USA, vol. 1*, pp. 240–244 (August 2003)

Semantically Oriented Sentiment Mining in Location-Based Social Network Spaces

Domenico Carlone and Daniel Ortiz-Arroyo

Computational Intelligence and Security Laboratory,
Department of Electronic Systems,
Aalborg University,
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
dcarlo09@student.aau.dk, do@cs.aaue.dk

Abstract. In this paper we describe a system that performs sentiment classification of reviews from social network sites using natural language techniques. The pattern-based method used in our system, applies classification rules for positive or negative sentiments depending on its overall score, calculated with the aid of SentiWordNet. We investigate several classifier models created from a combination of different methods applied at word and review levels. Our experimental results show that using part-of-speech helps to achieve better accuracy.

Keywords: Opinion Mining, Sentiment Classification, SentiWordNet, Social Networks.

1 Introduction

Sentiment analysis or opinion mining is an emerging discipline within the fields of information retrieval and natural language processing (NLP). Sentiment analysis consists in detecting the subjectivity and sentiments contained in general opinions. Opinions are expressions that describe the emotions and feelings of people regarding a subject, entity or event [1]. Conversely, facts are objective descriptions.

Sentiment analysis has many applications. For instance, it can be applied to understand people's attitudes for marketing analysis purposes. Moreover, the automatic detection of opinions can be used to substitute surveys and questionnaires. Finally, the Internet can be used as a source of information on people's opinions about products, services, events, or political topics. For this purpose, social network sites provide a convenient way to share opinions.

Many studies have been carried out on sentiment-based classification within the field of sentiment analysis. However, few of these studies have been performed in the domain of reviews posted in social network sites. In this paper we present an algorithm for mining opinions from some social network sites such as Foursquare, Yelp, Qype, Where, CitySearch. These sites are mainly concerned with describing "interesting" places within cities. In these social networks users

post opinions about clubs, events or restaurants and some of their features such as food quality, customer satisfaction or atmosphere.

Our system is capable of collecting and classifying user's opinions by identifying their semantic orientation. Reviews retrieved from social networks sites are classified based on the presence of certain terms that are likely to express sentiments. Opinions are classified as belonging to one of two opposing polarities: positive or negative [3]. In order to apply our classification method, the text from reviews is preprocessed using Natural Language Processing (NLP) techniques.

Since opinions frequently express the strength of a person's feelings with respect to some subject, our method associates a degree of positivity or negativity to each review/comment. This is done to obtain a ranked list of reviews for the best places. The effectiveness of the proposed system is evaluated in terms of precision, recall, F-measure and overall accuracy.

This paper is organized as follows. Section 2 presents a summary of related work. Section 3 describes in detail our system and in section 4 we present some experimental results. Finally section 5 provides our conclusions and describes future work.

2 Related Work

Some recent research work in sentiment analysis focuses on designing methods to determine the sentiment contained in documents. Other research focuses on more specific tasks, such as finding the sentiments of words [4] or searching for subjective expressions [5].

Machine learning and semantic orientation analysis are some of the methods applied in sentiment analysis. The former employs, for instance, well known probabilistic algorithms such as Naive Bayes (NB) [6]. The latest is a rule-based (or pattern-based) approach that applies Natural Language Processing (NLP) [2] techniques and external linguistic resources. One of the main differences between these two approaches lies in the need to use a training phase, in the case of supervised machine learning. The two approaches have been combined in a hybrid solution as is described in [8] and [9].

Methods based on the application of NLP-based techniques, extract phrases containing opinions using predefined part-of-speech (POS) patterns. In [7], Turney et al. use POS tagging to extract two-words phrases from reviews containing at least one adjective or one adverb. The semantic orientation is estimated assigning a score. Then an average is calculated with the scores obtained with the sentences and phrases contained in the reviews. Turney's work and others such as [4] found that there is a high correlation between the presence of adjectives and a sentence's subjectivity.

Other studies demonstrate that other parts of speech such as nouns and verbs are also good indicatives of sentiment [10]. In a similar work Pang et al. [6], examine three different machine learning methods for sentiment classification: Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy. They found that best performance was obtained when SVM was used in combination with unigrams, reaching a maximum accuracy of 83%.

The two techniques most commonly used in sentiment classification based on a semantic orientated approach are corpus-based and dictionary-based techniques. Within the former approach, Turney in [7] calculated the semantic orientation of a phrase using point-wise mutual information. This method essentially calculates the probability of collocations between the terms contained in a phrase and two reference words such as excellent and poor, that are representative of positive and negative polarities. Conversely, dictionary-based techniques utilize dictionaries and sentiment lexicons that provide information about semantic relations between words and terms' sentiment properties to determine overall sentiment of opinions.

The problem of identifying sentiment in text can be addressed by determining the subjectivity or semantic orientation (i.e. polarity) it contains. Lexicons addressing the former tasks are called *subjectivity lexicons* as they provide lists of subjective words. An example of this approach is introduced in [5]. Other lexicons include the prior polarity of words, such as Harvard General Inquirer (GI), Micro-WNOp, and SentiWordNet [11]. The first two include prior polarities together with indicators (i.e. adjectives) of term attitudes (e.g. "strong negative" or "weak positive"). SentiWordNet on the other hand, determines the degrees of words' polarities within the range [0,1]. SentiWordNet includes an evaluation not only of the positivity and negativity of a word but also its objectivity.

Other research work has applied SentiWordNet to the problem of automatically classifying sentiment. For instance, Pera, Qumsiyeh and Ng [12] introduced a domain independent sentiment classifier which categorizes reviews on the base of their semantic, syntactic, and sentiment content. To calculate the overall sentiment score of a review, the proposed classifier determines first the polarity score of each word contained in it; thereafter, it calculates the review's sentiment orientation by subtracting the sum of its negative words scores from the sum of its positive words scores.

Thet et al. in [13], proposed a linguistic approach for sentiment analysis of message posts on discussion boards, in which it is performed clause-level sentiment analysis. Firstly, they calculate the prior words' sentiment scores, employing SentiWordNet in combination with a lexicon from the domain of movie reviews especially built for the purpose. Then, they determine the contextual sentiment score for each clause by analyzing grammatical dependencies of words (through dependency trees) and handling pattern-rules.

In [14] Denecke tested rule-based and machine learning models in a multi-domain, classification scenario. Their results confirmed that the lexicon-based approach that made use of SentiWordNet had limited accuracy compared to the machine learning method.

Few studies have combined semantic orientation and machine learning approaches to improve Sentiment Classification performance. Ohana and Tierney [15] compared two approaches to assess the performance of using of SentiWordNet to the task of sentiment classification at document level on film reviews. In the first method, the lexicon was applied to count the positive and negative terms found in a document. Sentiment orientation was determined based

on which class received the highest score, similarly as was done in the methods described in [6] and [16]. Later, term scores were used to determine sentiment orientation. The second method in [15] employed SentiWordNet as a source of positive and negative features. These features were used to train a SVM supervised learning algorithm that showed an improvement in accuracy.

3 System Description

Our sentiment classification system for location-based social network sites performs a series of steps, starting with the collection of reviews from social network sites and ending up performing sentiment analysis and classification. Fig. 1 illustrates the steps performed by our system.

The dataset used in our experiments was extracted directly from social network sites, given that most of the datasets available in the domain of sentiment analysis belong to movie reviews and that no dataset was available in our domain. In this work we used “Yelp” and “Foursquare” sites as the data sources. Our dataset consists in geo-coded place reviews collected from these sites.

Reviews and other information contained in Yelp and Foursquare’s repositories were extracted by sending requests to their Web Services APIs. The retrieved information was about reviews on certain interesting places and locations. Then the data was stored in a database to ease its access.

Reviews were then processed through several stages using NLP steps as depicted in Fig. 1. First, tokens were extracted one at a time and then normalized using rules specifically designed for the English language. For instance, short forms’ expansion was employed to eliminate contractions. Terms were also transformed to lowercase for easing the searching for entries in the SentiWordNet database. For the same previous reason, words were brought to their base form through lemmatization.

Tokens were then tagged so that they could be used in the SentiWordNet lexicon. POS tagging was used to identify words, corresponding to parts-of-speech, that are good predictors of the sentiment expressed in sentences. If a lexicon entry corresponding to the analyzed token was found in SentiWordNet, the token score algorithm was applied. Then, the resulting token score was sent to our *Prior-Polarity Classifier* to be used in the calculation of the review’s sentiment Score. Next, the token score algorithm used by our classifier model was applied. Finally, our classifier determines if the processed review was positive, negative or objective. Following sections describe the most important stages of our system with more detail.

3.1 Classification Algorithm

Our classification algorithm takes as input all the normalized tokens coming out from the pre-processing phase, together with the related part-of-speech tags assigned. The collection of terms involved in calculating a review score is reduced to tokenized words that belong to one of the four POS classes of SentiWordNet (adjectives, adverbs, nouns, verbs).

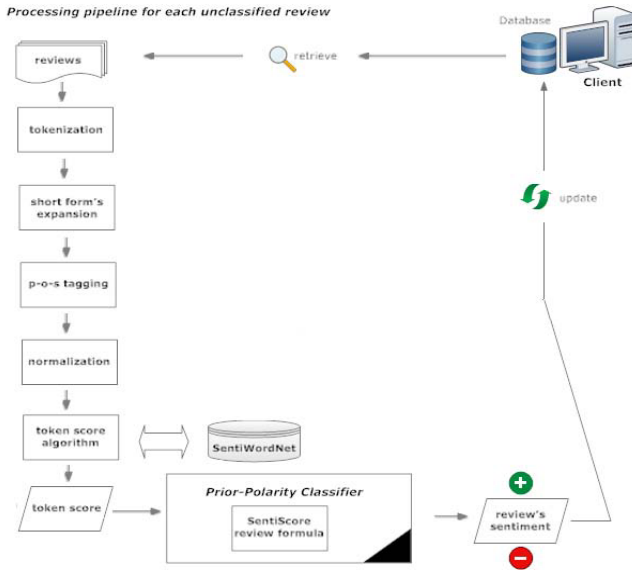


Fig. 1. Sentiment Analysis Pipeline

Words in Natural Language can be polysemous and because of their multiple meanings, tokens can have multiple entries in SentiWordNet. Consequently, in order to assign the polarity score to a word, it is first necessary to perform *Word Sense Disambiguation (WSD)*. However, our system in the current state does not apply any Word Sense Disambiguation method. Alternatively, to determine what is the effect of the different meanings of a word in the overall sentiment, we used a simple statistical approach. For each word, all possible senses are collected together with the three corresponding polarity scores: positive, negative and objective. Then, we applied and evaluated the following three strategies for the calculation of the final triple of token scores, namely:

- *Random Sense*
- *All Senses Arithmetic Mean*
- *POS matching Senses Arithmetic Mean*

The first method consists in the random selection of a sense among all possible senses of a word. This is the simplest approach that intuitively should show worst performance. The second method is the arithmetic mean of each of the three polarity scores computed, on all the possible senses, i.e. is an average of the sentiment entries of a word for all possible POS taggings. The third method is also an average of the sentiment entries of a word, but in this case the entries used to calculate the average are only those that match the POS tag assigned in the pre-processing phase. Therefore, in this method not all senses are considered but only the senses of the words found in SentiWordNet that match the computed

POS tag; if more than one sense belongs to the subset obtained after the POS tagging filtering, then the arithmetic mean is applied.

Each of the previous three scoring methods is applied to the three possible polarities: positive, negative, objective. At the end of this step, for each token we will have the following nine different scores:

- *Random Sense Score: Pos, Neg, Obj*
- *All Senses Arithmetic Mean Score: Pos, Neg, Obj*
- *POS matching Senses Arithmetic Mean Score: Pos, Neg, Obj*

The last six arithmetic mean scores are calculated with the formula:

$$score_{pol}(T) = \frac{1}{n} \sum_{s=1}^n score_{pol}(s) \quad (1)$$

where $pol \in \{pos, neg, obj\}$, and n is the number of the s senses (*synsets*) corresponding to the SentiWordNet entries for the *token* T . As was explained before, in the case of POS matching senses arithmetic mean score, n is reduced to the subset of all the senses in SentiWordNet that match the computed POS tag.

We applied the three methods obtaining a final triple score for positive, negative and objective scores. The approach we applied is similar to the one reported in [14] where positive and negative SentiWordNet scores for a term are compared. If the positivity (or negativity) is larger, the word is considered positive (or negative, respectively) and its strength is represented by its positivity (or negativity) score. If both values are equal, the word is ignored, since the interest is toward opinionated words. The objective value is only taken into account in the case we want to apply a cutoff value in order to exclude, from the computation of the overall sentiment review score, words that are too "objective".

To calculate the overall sentiment score of a review R we subtracted the sum of the scores of its negative words from the sum of its positive words scores as is shown in equation 2:

$$SentiScore(R) = \frac{\sum_{pos=0}^j Score(Token_{pos}) - \sum_{neg=0}^k Score(Token_{neg})}{j + k} \quad (2)$$

where j and k are the number of positive and negative words in R respectively, $Token$ is a word in R , $Score(Token)$ is the highest SentiWordNet score of the word considering the positive and negative scores.

Since large reviews can contain more or less positive or negative words, the different numbers may impact the sentiment score. For this reason *SentiScore* is normalized by dividing it by the number of sentiment words in R . Normalization keeps values within the interval $[-1,+1]$.

Finally if the *SentiScore*(R) obtained by equation 2 is higher (lower) than zero, then a review R is labeled as positive (negative); when *SentiScore*(R) is zero, it means that the score of positive words equals the score of negative words, in this case the review is considered objective.

3.2 Classifier Models

Our classifier employs the *SentiScore* equation [2](#) to classify reviews. However, we decided to apply several classifier models to investigate their effect in accuracy. The first model considers the inclusion of nouns in the estimation of the SentiScore. Words in SentiWordNet are partitioned into adjectives, adverbs, nouns and verbs. Sometimes nouns are judged to be objective words and in some research work they are completely excluded.

The second model consists in applying a cutoff to the objective score of a token to exclude, from the computation of the *SentiScore*, words that have a high degree of objectivity. It has to be noted that in SentiWordNet a word can be simultaneously positive, negative and objective. In fact, most SentiWordNet's words have an objective score greater than zero, even if they are positives or negatives. We decided that a word is considered polarized if its $ObjScore(T) < 1 - cutoff$. The reason to use this condition is because in SentiWordNet the summation of the positive, negative, objective scores for a term is 1, and the objective score results from the complement of positive plus negative scores. For instance applying a cutoff of 0.3 will exclude those words whose objective score is higher than 0.7. With this cutoff value we expressly allow to include words whose polarity is objective in the computation. Since our reviews are very short, applying a high cutoff limit together with the condition of POS tag matching, may reduce the number of words considered polarized to either a very small number or even zero. For words that pass the cutoff condition, the algorithm compares its positive and negative scores with the *SentiScore* formula.

The algorithm used to compute the semantic orientation of a word is shown below. POS can be restricted to just {verbs, adverbs, adjectives} in case we choose not to consider noun's POS senses.

```

for each Token = POS
consider the Score Triple calculated using a chosen score Method

  if ObjScore(T) > 1-(cutoff):
    do not include word in the SentiScore computation

  else
    if PosScore(T) > NegScore(T):
      add Token,Scores(Token) to positive set

    if NegScore(T) > PosScore(T):
      add Token,Scores(Token) to negative set

    if PosScore(T) = NegScore(T):
      do not include word in the SentiScore computation

end for each
Perform Sentscore computation using tokens' positive and negative scores

```

Note that if the positivity and negativity values of a word are equal, the difference between the polarity scores will be zero; therefore the word will not be included in the computation of the *SentiScore* formula since we are interested in opinionated words.

The algorithm that is most similar to our approach is presented in [8](#), where the overall sentiment score is calculated applying a classification rule. Conversely,

in [14] the number of positive, negative and objective words is calculated and their values compared to classify a review. In both, the strategy for the calculation of the token scores consists in the arithmetic mean executed on the triAple scores for all the term's senses found. A cutoff value is applied in [3].

4 Experimentation

In order to discover the best classification model, several criteria were applied at both, token and review levels. Similarly, we used several cutoff points, as was done in [3], where the best accuracy was reached with a 0.8 *cutoff*. In [3] only words that have a positive or negative polarity greater than the established cutoff are considered. However, when a cutoff point of 0.8 is used, the size of the SentiWordNet lexicon is reduced from 52,902 to 924. This approach is too strict to be applied in the short reviews we have in our dataset. Therefore, we decided to experiment using two lower cut-off values of 0.3 and 0.5. The rule we applied is that a token T , belonging to a review R , is considered in the computation of its $SentiScore(R)$ if $ObjScore(T) < 1 - cutoff$.

As a result of our experiments, we obtained 18 different sentiment scores for each review, corresponding to the 18 classifier models produced by combining the 3 token scoring methods with 6 different review scoring methods.

The system was implemented in Python using MySQL database and the open source library Natural Language Toolkit (NLTK)¹ for tokenization and part-of-speech tagging.

4.1 Dataset

Experiments were conducted on a dataset consisting of both 400 and 200 positive reviews additionally to 200 negative reviews. The reviews used are a subset of the whole data collected during the opinion extraction phase of our system.

It must be noted that Yelp's reviews are rated on a 5-point scale, with 1 being the most negative and 5 being the most positive. We decided to convert these favorability ratings into a polarity corresponding to one of the three sentiment categories (positive, negative, neutral), to being able to use them during testing. Since each review has a rating based on the number of stars (1 to 5), we decided to use 1 or 2 as a negative rating and 4 or 5 as positive one. Opinions marked with 3 stars are considered neutral (objective) and therefore excluded from the evaluation. As it is suggested in [6] and [10], ratings in terms of the number of stars, can be used as indicator of the overall sentiment of reviewers.

4.2 Evaluation Metrics and Results

The effectiveness of the system was evaluated in terms of Precision, Recall, F-measure and overall Accuracy.

The contingency Table 1 shows true positives and true negatives as the correct classifications. Precision and recall metrics are split in *Positive Precision* ($Prec_p$)

¹ <http://www.nltk.org/>

Table 1. Relevance/Retrieval contingency table

	Relevant	Nonrelevant
Retrieved	<i>true positives (tp)</i>	<i>false positives (fp)</i>
Not Retrieved	<i>false negatives (fn)</i>	<i>true negatives (tn)</i>

Table 2. Equations for Precision and Recall

	Positive	Negative
Precision	$Prec_p = tp / (tp + fp)$	$Prec_n = tn / (tn + fn)$
Recall	$Rec_p = tp / (tp + fn)$	$Rec_n = fn / (tp + fn)$

Table 3. Evaluation of results: Precision

Metric	Review Score's Methods	Token Score's Methods		
		Random	all Senses AM	P-O-S, AM
$Prec_n$	cutoff=0	48,5%	55%	56,5%
	cutoff=0, no-nouns=true	49%	53%	53,5%
	cutoff=0.3	52,5%	56%	57,5%
	cutoff=0.3, no-nouns=true	48,5%	52%	55%
	cutoff=0.5	52%	54%	50,5%
	cutoff=0.5, no-nouns=true	49%	50%	48,5%
$Prec_p$	cutoff=0	65,5%	68%	65%
	cutoff=0, no-nouns=true	61%	68%	65,5%
	cutoff=0.3	49,5%	55%	53%
	cutoff=0.3, no-nouns=true	44%	47%	52%
	cutoff=0.5	32%	32%	35%
	cutoff=0.5, no-nouns=true	29,5%	26,5%	35%

and *Positive Recall* (Rec_p), *Negative Precision* ($Prec_n$), and *Negative Recall* (Rec_n) as shown in Table 2.

The *Accuracy* of the system is given by:

$$\text{Accuracy} = (tp + tn) / (tp + fp + fn + tn) \tag{3}$$

Precision and Recall are combined in the *F-measure*:

$$\text{F-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Tables 3, 4, 5, 6 summarize classifier’s performance in terms of precision, recall, F-measure, and overall accuracy respectively. These tables show the results we obtained using different methods at token and review level and that the best results were obtained using no cut-off and a token score method based on POS matching senses.

The differences in performance between reviews’ classification on positive and negative opinions, as measured by precision, recall and F-Measure (e.g. Table 3, Table 4, Table 5), may be attributed to a cause mentioned in [15]. [15] describes that reviewers generally include negative remarks on positive opinions to provide

Table 4. Evaluation results: Recall

Metric	Review Score's Methods	Token Score's Methods		
		Random	all Senses AM	P-O-S, AM
Rec_n	cutoff=0	44%	39,82%	40,092%
	cutoff=0, no-nouns=true	45,54%	40,87%	41,518%
	cutoff=0.3	48,97%	44,44%	44,5%
	cutoff=0.3, no-nouns=true	53,93%	50,526%	46,4%
	cutoff=0.5	60%	41,02%	58,58%
	cutoff=0.5, no-nouns=true	63,35%	65,36%	59,5376%
Rec_p	cutoff=0	55,98%	60,18%	59,907%
	cutoff=0, no-nouns=true	54,464%	58,8745%	58,482%
	cutoff=0.3	51%	55,555%	55,497%
	cutoff=0.3, no-nouns=true	46,07%	49,47%	53,608%
	cutoff=0.5	40%	41,025%	41,42%
	cutoff=0.5, no-nouns=true	36,646%	34,64%	40,462%

Table 5. Evaluation results: F-Measure

Metric	Review Score's Methods	Token Score's Methods		
		Random	all Senses AM	P-O-S, AM
$F - score_n$	cut-off=0	46,14%	46,195%	46,9%
	cut-off=0, no-nouns=true	47,2%	46,15%	46,75%
	cut-off=0.3	50,67%	49,555%	50,17%
	cut-off=0.3, no-nouns=true	51,071%	51,252%	50,335%
	cut-off=0.5	55,714%	44,9%	54,24%
	cut-off=0.5, no-nouns=true	55,26%	56,657%	53,455%
$F - score_p$	cut-off=0	60,37%	63,8514%	62,35%
	cut-off=0, no-nouns=true	57,547%	63,106%	61,8%
	cut-off=0.3	50,24%	55,276%	54,22%
	cut-off=0.3, no-nouns=true	45,011%	48,203%	52,79%
	cut-off=0.5	35,555%	35,955%	37,94%
	cut-off=0.5, no-nouns=true	32,7%	30,03%	37,53%
$F - score$	cutoff=0	53,255%	55,0232%	54,625%
	cutoff=0, no-nouns=true	52,373%	54,628%	54,275%
	cutoff=0.3	50,455%	52,4155%	52,195%
	cutoff=0.3, no-nouns=true	48,041%	49,7275%	51,5625%
	cutoff=0.5	45,6345%	40,4275%	46,09%
	cutoff=0.5, no-nouns=true	43,98%	43,3435%	45,4925%

Table 6. Evaluation results: Accuracy

Metric	Review Score's Methods	Token Score's Methods		
		Random	all Senses AM	P-O-S, AM
Accuracy	cutoff=0	57%	61,5%	60,75%
	cutoff=0, no-nouns=true	55%	60,5%	59,5%
	cutoff=0.3	51%	55,5%	55,25%
	cutoff=0.3, no-nouns=true	46,25%	49,5%	53,5%
	cutoff=0.5	42%	43%	42,75%
	cutoff=0.5, no-nouns=true	39,25%	38,25%	41,75%

a more balanced assessment. Additionally reviewers may choose to build up the expectation of a general good view to ended up giving a negative impression.

One of the reasons for the low accuracies in Table(6) may be due to the limited number of opinionated words contained in the short reviews collected. Our results also show that classifier’s performance decreases when the cutoff value is increased. The reason may be also due to the short reviews given that applying a high cutoff, together with the condition of POS tag matching, reduces the number of words considered polarized to either zero or a very small number. This reduces the information at classifier’s disposal to correctly determine a review’s sentiment orientation.

5 Conclusions and Future Work

In this paper we have described a rule-base classifier model that exploits SentiWordNet. Our model was applied to classify reviews of interesting places extracted from social network sites. Our method achieved an accuracy in classification comparable to those obtained by previous similar systems e.g. [8], [14], [3]. However, since these systems were evaluated using different datasets in different domains, no direct comparison can be performed at this time.

A number of factors affect the performance of our classifier. For instance, the use of ironic words or colloquial language makes difficult to determine the polarity of a review. Additionally, most of the errors we obtained came from the wrong assignment of prior sentiment scores to words. For instance, words that have certain polarity in SentiWordNet may have a different polarity within the context of a review. Other inaccuracies come from the assignment of part-of-speech tags; for example, in a phrase such as “What a cool place”, the term cool is wrongly tagged as proper noun, and consequently identified by SentiWordNet as being objective, instead of being positive (adjective). More imprecisions may come from SentiWordNet itself since it has been found that some words have the wrong scores assigned. As future work we consider using word sense disambiguation and combining our rule-based method with a machine-learning approach.

References

1. Liu, B.: Sentiment analysis and subjectivity. In: Indurkha, N., Damerau, F.J. (eds.) *Handbook of Natural Language Processing*, 2nd edn. CRC Press, Taylor and Francis Group (2010) ISBN 978-1420085921
2. Nasukawa, T., Yi, J.: Sentiment analysis: capturing favorability using natural language processing. In: *K-CAP 2003: Proceedings of The 2nd International Conference on Knowledge Capture*, pp. 70–77. ACM, New York (2003)
3. Mejova, Y.: Tapping into sociological lexicons for sentiment polarity classification. In: *4th Russian Summer School in Information Retrieval*, pp. 14–27 (September 2010)

4. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics, pp. 174–181 (1997)
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT 2005: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354 (2005)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP 2002: Proceedings of the ACL-2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
7. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424 (2002)
8. Denecke, K.: Using sentiwordnet for multilingual sentiment analysis. In: ICDE Workshops 2008, pp. 507–512 (2008)
9. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. *Journal of Informetrics* 3(2), 143–157 (2009)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* 2(1-2), 1–135 (2008)
11. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta (May 2010)
12. Pera, M.S., Qumsiyeh, R., Ng, Y.-K.: An unsupervised sentiment classifier on summarized or full reviews. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 142–156. Springer, Heidelberg (2010)
13. Thet, T.T., Na, J.-C., Khoo, C.S.G., Shakthikumar, S.: Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In: Proceeding of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA 2009, pp. 81–84. ACM, New York (2009)
14. Denecke, K.: Are sentiwordnet scores suited for multidomain sentiment classification? In: Fourth IEEE International Conference on Digital Information Management, ICDIM 2009, November 1-4, pp. 33–38. University of Michigan, IEEE, Ann Arbor, Michigan (2009)
15. Ohana, B., Tierney, B.: Sentiment classification of reviews using SentiWordNet. In: 9th. IT & T Conference, p. 13 (2009)
16. Kennedy, A., Inkpen, D.: Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence* 22(2), 110–125 (2006)

Skyline Snippets

Markus Endres and Werner Kießling

University of Augsburg
Institute for Computer Science
86135 Augsburg, Germany

{endres,kiessling}@informatik.uni-augsburg.de

<http://www.informatik.uni-augsburg.de/en/chairs/dbis/>

Abstract. There is a strong demand for a deep personalization of search systems for many Internet applications. In this respect the proper handling of user preferences plays an important role. Here we focus on the efficient evaluation of the Pareto preference operator for structured data in very large databases. The result set of such a Pareto query, also known as the “skyline”, tends to become very large for higher dimensionalities. Often it is too time-consuming or just not necessary to compute the entire skyline, instead only some fraction of it, called a “snippet”, is sufficient. In this paper we contribute a novel algorithm for a fast computation of such skyline snippets. Our solutions do not rely on the availability of specialized pre-computed indexes, hence are generally applicable. We demonstrate the performance of our approach by several benchmarks studies. The presented results suggest that even for complex Pareto queries, yielding very large skylines, snippets can be computed sufficiently fast, and therefore can be integrated into online Web services.

1 Introduction

Today there is an abundance of Web services that enable users to access and query structured data from relational databases over the Internet. With the advent of convenient mobile Internet access through the latest generation of smartphones this trend becomes even stronger. Thus there is a big demand to provide proper search engine technology for this type of Internet usage. This requirement has to address two aspects: On the one hand, an excellent relevance of the retrieved data has to be achieved, avoiding pitfalls like empty-results searches or at the other extreme flooding with far too many search results. On the other hand, online users expect fast response times in the range of at most a few seconds. Both requirements become even more vital for mobile Internet access, where display sizes are limited and mobile phone costs are still high these days. To improve the relevance of search results the issue of personalization has been studied quite intensively recently. Thereby the concept of user preferences has turned out to play a crucial role. Preferences are an integral part of many important aspects of our daily life, e.g. when buying a car, renting an apartment or looking for a new job. In fact, preference handling has evolved as a broad multi-disciplinary research area [1], in particular within the AI and database research

communities. For structured data stored in relational SQL databases extensions like the Preference SQL language from Kießling and Köstler [2] have been proposed. For example, consider the food database provided by *The United States Department of Agriculture* (USDA, <http://www.nal.usda.gov/fnic/>) containing nutritional facts for more than 7000 types of food. A user may be interested in finding meals that satisfy his or her preferences concerning a meal. A sample preference query is shown in Figure 1 using Preference SQL:

```
SELECT * FROM Soup S, Meat M, Beverage B
PREFERRING S.Name IN (Chicken, Noodle) AND
                M.Name IN (Beef)                AND
                B.Vc HIGHEST 5
```

Fig. 1. Sample Preference SQL query

In this running example a user expresses his or her preferences after the keyword `PREFERRING`. It is a Pareto preference (`AND` in the `PREFERRING`-clause) consisting of three preferences on soups, meat, and beverages. The keyword `IN` denotes a preference for members of a given set, a so-called POS-preference. Therefore the user prefers Chicken and Noodle soups over all others. Furthermore, the user wants Beef and a drink with a maximum of vitamin C (`B.Vc HIGHEST`).

The result of this preference query returns the best matches found in the food database regarding the stated user preferences. In database terminology such Pareto preference queries are usually called *skyline queries*. High efficiency of skyline query evaluation is very essential, even more for mobile Internet access. Therefore preference (skyline) query optimization techniques have been investigated recently, both considering algebraic optimization techniques as well as efficient evaluation algorithms for large databases [3,4,5,6].

The focus of this paper will now be as follows: It is well known that skyline query results tend to get large for higher dimensionality and large database relations. In our running example we have only 3 dimensions, but dimensions of up to 10 are not uncommon; also the food database is only moderately large, but database containing many millions of tuples do occur in practice as well. Computing the full skyline under such circumstances is not very helpful for the user in many applications, and moreover, the response times would be unacceptably slow. Therefore methods that achieve to progressively and efficiently retrieve only some fraction of the skyline are of practical interest. All of them have in common that they have to establish and maintain some highly specialized indexes. However, for arbitrary skyline queries this requirement is not satisfied in general. Instead, in this paper we want to propose a novel approach for efficiently computing a fraction of the whole skyline, called *skyline snippets*, without having to rely on suitable pre-existing index structures.

The problem most related to skyline snippets has been addressed in the literature under the topic of subspace skylines. The aim of a subspace skyline is to find

the best objects concerning a subset of all available attributes, whereas a traditional skyline query uses all attributes to determine the skyline. One method to compute subspace skylines is to pre-materialize the skylines for all subspaces and organize them in data cubes [7,8]. This requires large storage space and is hard to maintain. Another technique uses some index structure [9], which entails the same problems as index-based skyline algorithms. There are further algorithms to compute only a fraction of the full skyline. Progressive skyline algorithms, e.g., [10,11,12] return interesting points as they are identified using an index structure. All of them need pre-processing of the data. Top-k skyline evaluation calculates the first stratum or skyline with some kind of post-processing [13]. Further techniques to reduce the number of skyline points in high dimensions have also been proposed, e.g., the skyline frequency [14].

The rest of this paper is organized as follows: In Section 2 we review the theoretical background of our preference model for relational database systems, addressing the notions of preference constructors and of sub- and super-preferences. Thereafter Section 3 will provide our central theoretical results for skyline snippets computation. We have implemented a prototype of our novel skyline snippets algorithm and have performed intensive performance benchmarks, which will be reported in Section 4. Finally Section 5 summarizes our claimed contributions and outlines further research directions.

2 Preference Background

Preference (skyline) queries and their integration into databases have been in focus for some time, leading to diverse approaches, e.g. [4,15,16]. We shortly review the preference model from [16].

Definition 1. Preference

A preference $P = (A, <_P)$, where A is a set of attributes, is a strict partial order on the domain of A . Thus $<_P$ is irreflexive and transitive. The term $\mathbf{x} <_P \mathbf{y}$ is interpreted as “I like y more than x ”.

The BMO-set (also called *winnow* by [4]), of a preference $P = (A, <_P)$ on an input database relation R are all tuples that are not dominated w.r.t. the preference [15,16].

Definition 2. BMO-set

The Best-Matches-Only result set (BMO) contains only the best matches w.r.t. the strict partial order of a preference P . It is computed by the preference selection:

$$\sigma[P](R) := \{t \in R \mid \neg \exists t' \in R : t <_P t'\}$$

It finds all best matching tuples t for the preference P with $A \subseteq \text{attr}(R)$, where $\text{attr}(R)$ denotes all attributes of a database relation R .

Preferences on single attributes are called *base preferences*. There are two distinct types of base preferences, namely *numerical* and *categorical*, depending on the

data type of the domain A . For the scope of this paper we concentrate on base preference constructors that specify *weak orders* [16]. A preference P is a weak order preference (WOP) if negative transitivity holds, i.e., for two domain values x and y

$$\neg(x <_P y) \wedge \neg(y <_P z) \implies \neg(x <_P z)$$

For a WOP $P = (A, <_P)$ the dominance test can be efficiently done by a numerical *utility function* $f_P : \text{dom}(A) \rightarrow \mathbb{R}_0^+$ which depends on the type of preference [5]. Dominated domain values have higher function values.

$$x <_P y \iff f_P(x) > f_P(y)$$

For the purpose of this paper we consider the following base preference constructors (more can be found in [15,16]):

▷ $P := \text{POS}(A, \text{POS-set})$:

The POS-set specifies a set of values preferred over all other values in $\text{dom}(A)$.

The utility function is

$$f_P(x) := \begin{cases} 0 & : x \in \text{POS-set} \\ 1 & : x \notin \text{POS-set} \end{cases}$$

The sample query in Figure 1 shows two POS-preferences. The first one on the kind of soup $P_1 := \text{POS}(S.Name, \{Chicken, Noodle\})$, and the second on the kind of meat $P_2 := \text{POS}(M.Name, \{Beef\})$.

Continuous numerical domains need a different type of preference. For this purpose [16] defined several numerical preference constructors, e.g., HIGHEST and LOWEST. The extremal preferences HIGHEST and LOWEST allow users to express their desire for values as high or as low as possible. We use the advanced version of numerical preference constructors allowing the partitioning of the range of domain values [16]. For this purpose the so-called d -parameter ($d > 0$) is introduced. For numerical base preferences the utility function is interpreted as the numerical distance from a perfect value.

▷ $P := \text{HIGHEST}_d(A)$:

A value x is worse than a value y if the value of x is lower than that of y . The best possible value is the maximum value of $\text{dom}(A)$, max. The utility function is:

$$f_P(x) := \left\lceil \frac{\max - x}{d} \right\rceil$$

▷ $P := \text{LOWEST}_d(A)$:

A value x is worse than a value y if the value of x is higher than that of y . The best possible value is the minimum value of $\text{dom}(A)$, min. The utility function is:

$$f_P(x) := \left\lceil \frac{x - \min}{d} \right\rceil$$

For example, the query in Figure 1 shows a HIGHEST preference for the amount of Vc in the *Beverage* products: $P_3 := \text{HIGHEST}_{d=5}(B.Vc)$. We set $d = 5$. This means, differences up to 5mg do not matter.

There is the need to combine several base preferences into more *complex preferences*. One way is to list a number of preferences that are all equally important to the user. This is the well-known concept of *Pareto preferences*.

Definition 3. *Pareto Preference*

For WOPs $P_1 = (A_1, <_{P_1}), \dots, P_m = (A_m, <_{P_m})$, a *Pareto preference*

$$P := \otimes(P_1, \dots, P_m) = (A_1 \times \dots \times A_m, <_P)$$

is defined as:

$$(x_1, \dots, x_m) <_P (y_1, \dots, y_m) \iff \exists i \in \{1, \dots, m\} : f_{P_i}(x_i) > f_{P_i}(y_i) \wedge \\ \forall j \in \{1, \dots, m\}, j \neq i : f_{P_j}(x_j) \geq f_{P_j}(y_j)$$

The query in Figure 1 shows a Pareto preference, where all three preferences are equally important: $P := \otimes(P_1, P_2, P_3)$

If we restrict the attention to LOWEST (MIN) and HIGHEST (MAX) as input preferences for a Pareto preference P , then Pareto preference queries coincide with the traditional *skyline queries* [17]. The BMO-set of a Pareto preference query P is generally referred to as “the skyline” of P .

Based on a Pareto preference we define *sub-* and *super-preferences*. A *sub-preference* is a part of a Pareto preference P , and a *super-preference* is a generalization of a preference P . Note that considering *sub-preferences* is similar to the concept of *subspace skylines* [7,8,18].

Definition 4. *Sub- and Super-Preference*

Given a Pareto preference $P := \otimes(P_1, \dots, P_m)$, we define a *sub-preference* of P as $P^I := \otimes(P_{i_1}, \dots, P_{i_n})$ where $I := \{P_{i_1}, \dots, P_{i_n}\}$, $i_j \in \{1, \dots, m\}$, is a set describing the preferences P_{i_j} taking part in P^I .

Each P^{I_2} is called a *super-preference* of P^{I_1} , if $I_1 \subset I_2$.

In Figure 1 for $P := \otimes(P_1, P_2, P_3)$ we have for example three sub-preferences: $P^{\{P_1, P_2\}} := \otimes(P_1, P_2)$, $P^{\{P_1, P_3\}} := \otimes(P_1, P_3)$, and $P^{\{P_2, P_3\}} := \otimes(P_2, P_3)$. Moreover, P itself is a super-preference of $P^{\{P_1, P_2\}}$ and $P^{\{P_1, P_3\}}$.

3 Skyline Snippets

Now we introduce and analyze the novel concept of *skyline snippets* in order to compute a fraction of the skyline. Skyline snippets do not rely on any pre-computation and thus can be applied to any kind of query.

It is well-known that skyline membership in general is *not monotonic*, cp. [15,18]. This means, given a sub-preference P^I of a preference P then the following does not hold in general:

$$\sigma[P^I](R) \not\subseteq \sigma[P](R)$$

However, there might be special situations where skyline membership of data points can be proved without having to compare them to all data points. This is the challenge to be solved in this paper, without the help of any specialized pre-built index structures.

3.1 Pareto k-Partition

Until now we have considered sub-preferences of a preference $P = \otimes(P_1, \dots, P_m)$ and the relationship between different sub-preferences. On the other hand, we can consider a *disjoint partition* of P , i.e., we do not compare sub-preferences P^{I_1} and P^{I_2} where $I_1 \subset I_2$, but consider sub-preferences where $I_1 \cap I_2 = \emptyset$ and $I_1 \cup I_2 = \{P_1, \dots, P_m\}$.

Definition 5. *k-Partition*

Given a Pareto preference $P = \otimes(P_1, \dots, P_m)$. Then we define a *k-partition* as a decomposition of P into k disjoint Pareto sub-preferences P^{I_i} , $i = 1, \dots, k$, such that

$$\otimes(P_1, \dots, P_m) = \otimes(P^{I_1}, \dots, P^{I_k})$$

with $\bigcap_{i=1}^k I_i = \emptyset$ and $\bigcup_{i=1}^k I_i = \{P_1, \dots, P_m\}$.

Example 1. Consider a Pareto preference $P = \otimes(P_1, \dots, P_4)$ and $k = 2$. Then a few 2-partitions of P are: $P = \otimes(P^{\{P_1, P_2\}}, P^{\{P_3, P_4\}})$, $P = \otimes(P^{\{P_1, P_3\}}, P^{\{P_2, P_4\}})$, and $P = \otimes(P^{\{P_1\}}, P^{\{P_2, P_3, P_4\}})$.

Since Pareto is associative and commutative [15], each Pareto preference can be *k-partitioned* arbitrarily. The next theorem shows the relationship between the preference itself and its *k-partitions*.

Theorem 1. *Skyline Snippets*

Consider a Pareto preference $P = \otimes(P_1, \dots, P_m)$, its *k-partition* $\otimes(P^{I_1}, \dots, P^{I_k})$ and the skyline $S = \sigma[P](R)$ on some relation $R = (A_1, \dots, A_m)$.

- a) Let $S_k = \bigcup_{i=1}^k \sigma[P^{I_i}](R)$, then
 - i) $\sigma[P](S_k) \neq \emptyset$
 - ii) $\sigma[P](S_k) \subseteq S$

$\sigma[P](S_k)$ is called a *k-snippet* of the skyline S .

- b) Let $L_k = \bigcap_{i=1}^k \sigma[P^{I_i}](R)$. If $L_k \neq \emptyset$, then $L_k \subseteq S$.

Proof

- a)
 - i) $\sigma[P](S_k) \neq \emptyset$ is obvious since preference selection never will be empty.
 - ii) We prove $\sigma[P](S_k) \subseteq S$. Let $t = (a_1, \dots, a_m) \in \sigma[P](S_k)$. This means, $\neg \exists t' = (a'_1, \dots, a'_m) \in S_k$ such that $(a_1, \dots, a_m) <_P (a'_1, \dots, a'_m)$. Furthermore, $t \in S_k$, i.e., $\neg \exists t'' = (a''_1, \dots, a''_m) \in R$ such that

$$(\exists P^{I_i}, 1 \leq i \leq k : (a_1, \dots, a_m) <_{P^{I_i}} (a''_1, \dots, a''_m))$$

It follows: $\neg \exists t' = (a'_1, \dots, a'_m) \in R$ such that

$$(a_1, \dots, a_m) <_P (a'_1, \dots, a'_m) \Leftrightarrow t \in S = \sigma[P](R)$$

- b) Let $t = (a_1, \dots, a_m) \in L_k$. Then $\neg \exists t' = (a'_1, \dots, a'_m) \in R$ such that

$$(a_1, \dots, a_m) <_{P^{I_i}} (a'_m, \dots, a'_m), \forall 1 \leq i \leq k \implies t \in S = \sigma[P](R)$$

Example 2. Consider a Pareto preference $P := \otimes(P_1, \dots, P_4)$ on a database relation $R(A_1, A_2, A_3, A_4)$, all $P_i := \text{LOWEST}_{d=1}(A_i)$. Compare Table 1 for a sample data set of R .

Table 1. Sample data set

R	ID	A₁	A₂	A₃	A₄
	t_1	0	1	0	0
	t_2	0	0	1	0
	t_3	1	0	0	1

The skyline is $S := \{t_1, t_2, t_3\}$. Let $P^{\{P_1, P_3\}}$ and $P^{\{P_2, P_4\}}$ be a 2-partition of P . Then we have $\sigma[P^{\{P_1, P_3\}}](R) = \{t_1\}$ and $\sigma[P^{\{P_2, P_4\}}](R) = \{t_2\}$. Thus, our 2-snippet has the following tuples: $\sigma[P](\{t_1\} \cup \{t_2\}) = \{t_1, t_2\} \subseteq S$. Furthermore, have a look at a 2-partition of P , namely $P^{\{P_1\}}$ and $P^{\{P_2, P_3, P_4\}}$. Then we have $\sigma[P^{\{P_1\}}](R) = \{t_1, t_2\}$ and $\sigma[P^{\{P_2, P_3, P_4\}}](R) = \{t_1, t_2, t_3\}$. In this case, L_k is a subset of S : $L_k := \{t_1, t_2\} \cap \{t_1, t_2, t_3\} = \{t_1, t_2\}$.

The example above demonstrates that our skyline snippets prefer “extremal” skyline objects. Each partition computes the best objects concerning the corresponding sub-preference.

3.2 The SSA Algorithm

Computing the skyline by Theorem 1a) is a more complex task than the evaluation by Theorem 1b). However, in Theorem 1b) it is not guaranteed that the intersection of the skylines of the sub-preferences is not empty. If the intersection is not empty, than we have a *lucky snippet*. Otherwise we have to compute additionally some snippets by Theorem 1a). Therefore, it is only interesting to consider Theorem 1a) in detail, which leads us directly to the skyline snippets algorithm SSA sketched in Algorithm 1.

Algorithm 1. Skyline Snippets Algorithm (SSA)

Input: Data set R , Pareto preference $P := \otimes(P_1, \dots, P_m)$, k

Output: A partial Skyline of P

- 1: **function** $SSA(P := \otimes(P_1, \dots, P_m), R, k)$
 - 2: $\mathcal{P} \leftarrow$ select a k -partition of P , $U \leftarrow \emptyset$ //Initialize union set
 - 3: **for all** $p \in \mathcal{P}$ **do**
 - 4: $U \leftarrow U \cup \sigma[p](R)$ //Evaluate p on R by a skyline algorithm
 - 5: **end for**
 - 6: **return** $\sigma[P](U)$ //Evaluate P on U by a skyline algorithm
 - 7: **end function**
-

The algorithm has three arguments: the Pareto preference P , the data set R and the value k to decompose the Pareto preference in k -partitions. First, select a

k -partition (line 2). Afterwards, initialize the union set of the sub-preferences' BMO-sets. From line 3 on evaluate each sub-preference in \mathcal{P} by an arbitrary skyline algorithm, e.g. 5,17. Note that the evaluation of the sub-preferences can be done in parallel on multi-core architectures. Finally, evaluate the original Pareto preference P on the union of the BMO-sets (line 6).

4 Performance Benchmarks

In this section we present some benchmarks on our skyline snippets algorithm (SSA, Algorithm 1) in comparison to Hexagon 5 (also known as Lattice Skyline 19) and a progressive variant of Hexagon. A progressive skyline algorithm returns interesting points as they are identified using a special data structure.

4.1 Test Framework

For the benchmarks we used our *Preference SQL* system 3, a Java SE 6 framework for preference queries on conventional database systems. This framework implements our skyline snippets algorithm and Hexagon. All experiments are performed on a 2.53GHz Core 2 Duo machine running Mac OS X with 4 GB RAM for the JVM. We used an Oracle 11g database to store all generated data. All used Pareto preferences consists of LOWEST_d preferences.

For the evaluation we use synthetic data sets. This allows us to carefully explore the effect of various data characteristics. For this, we generate data sets with anti-correlated (ANTI), correlated (COR), and independent (IND) distributions using an implementation of the popular data set generator of 17. We modify three parameters: the data cardinality n , the number of distinct domain values c , and the d -value.

4.2 Benchmark Results

We present benchmarks to demonstrate the performance of the skyline snippets algorithm. As a yardstick we compare it to the Hexagon algorithm and its progressive variant, both not requiring special pre-built indexes. We discuss the results for anti-correlated and correlated data sets. For the results on independent data we refer to 20.

Benchmark 1: Hexagon vs. SSA. Our first test series compares our snippets algorithm to Hexagon. The task for Hexagon is to compute the *full* skyline whereas SSA only have to compute a few skyline points. We fixed the data cardinality to $n = 500K$ and the domain size to $c = 100K$. Furthermore, we varied the number of dimensions from 2 to 10 and fixed the d -value to $d = 10K$. The results are shown in the Figures 2a, 2b for different data distributions. For higher dimensions SSA demonstrates its advantage: a very fast computation of some skyline points.

Benchmark 2: Hexagon (prog.) vs. SSA. Table 2 and 3 present benchmarks for SSA in comparison to the progressive variant of Hexagon. We stopped this

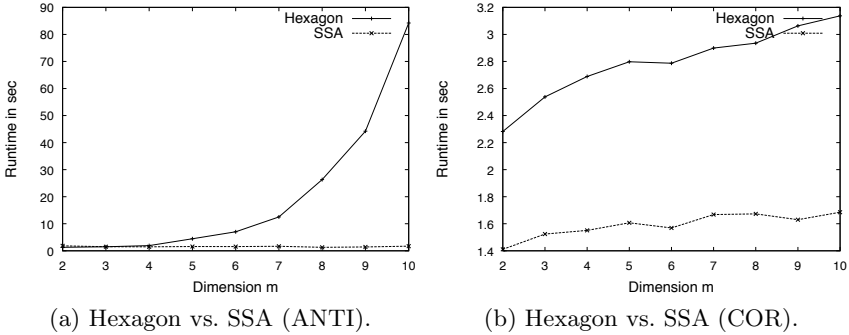


Fig. 2. Example for sub-preferences and a monotonic Skyline membership

progressive algorithm after it has computed as many skyline points (#Skylines) as SSA, since SSA produces an unknown number of best objects. In this benchmark we used a Pareto preference $P := \otimes(P_1, \dots, P_8)$. We generated arbitrary k -partitions $k = 1, 2, 4, 8$ to evaluate the influence of the partitions to the results and runtimes of SSA. The data cardinality is fixed to $n = 500K$ with a domain cardinality $c = 100K$. In the case of anti-correlated data SSA computes 5902 tuples in 7.92sec if $k = 1$, i.e., the original Pareto preference must be evaluated (only one partition). Since all Pareto preferences were computed by the standard Hexagon algorithm, the runtime to the progressive Hexagon differs only slightly. On the other hand, if $k = 4$, SSA computes 1075 in less than one second, whereas Hexagon (prog.) needs about 6 sec.

Table 2. Hexagon (prog.) vs. SSA (ANTI)

	#Skylines	sec	#Skylines	sec	#Skylines	sec	#Skylines	sec
Hexagon _p	5902	7.64	3801	6.22	1075	5.95	419	5.29
SSA	5902	7.92	3801	3.81	1075	0.812	419	0.198
	$k = 1$		$k = 2$		$k = 4$		$k = 8$	

Table 3. Hexagon (prog.) vs. SSA (COR)

	#Skylines	sec	#Skylines	sec	#Skylines	sec	#Skylines	sec
Hexagon _p	3440	3.12	2436	2.56	637	2.30	219	2.01
SSA	3440	3.10	2436	1.98	637	0.78	219	0.11
	$k = 1$		$k = 2$		$k = 4$		$k = 8$	

Benchmark 3: Skyline points, Hexagon vs. SSA. We compare the number of tuples in the complete skyline to the number of tuples computed by SSA. For the computation of the complete skyline we applied the original Hexagon algorithm. We set the data cardinality to $n = 500K$, the number of distinct

domain values to $c = 100K$, and the d -parameter to $d = 10K$. We considered a Pareto preference $P := \otimes(P_1, \dots, P_m)$, $m = 2, \dots, 8$. For simplicity, we generated only $\frac{m}{2}$ -partitions. Note that we arranged the partitions in the order of the preferences in P . Table 4 and 5 show the number of skyline points found by Hexagon (the complete skyline) and our snippets algorithm SSA. Note that for each dimension m we generated a new data set. We also show the number of skyline points found by each partition of P . For dimension $d = 2$ our snippets algorithm always finds all skyline points because there is only one partition.

Table 4. Skyline points computed by Hexagon and SSA (ANTI)

m	# Skyline points	$\sigma[P](S_k)$	$P^{\{P_1, P_2\}}$	$P^{\{P_3, P_4\}}$	$P^{\{P_5, P_6\}}$	$P^{\{P_7, P_8\}}$
2	771	771	771	-	-	-
4	12312	1348	1211	1394	-	-
6	18771	2851	1378	1631	1299	-
8	24432	5495	1812	1919	1058	1403

Table 5. Skyline points computed by Hexagon and SSA (COR)

m	# Skyline points	$\sigma[P](S_k)$	$P^{\{P_1, P_2\}}$	$P^{\{P_3, P_4\}}$	$P^{\{P_5, P_6\}}$	$P^{\{P_7, P_8\}}$
2	659	659	659	-	-	-
4	3126	982	706	703	-	-
6	8931	117	516	621	581	-
8	11026	1131	643	681	657	597

4.3 Observations

To summarize our benchmarks experiences gained so far we can state that the skyline snippets algorithm SSA shows excellent performance. Since measured response times typically were in the range of at most a few seconds we conclude that SSA is sufficiently fast for many practical application scenarios, including mobile Web service access. Secondly, we could observe that the fraction of the whole skyline retrieved by SSA is significantly smaller, but still contains sufficient skyline points for a user in real-world applications, which in particular pays off for mobile Internet usage. We performed our tests by using Hexagon and its progressive variant as a yardstick. Other progressive skyline algorithms might be faster, but at the burden of having to maintain special pre-built indexes which, however, are not always available if needed. Thus, if one is looking for a generic algorithm to quickly compute a fraction of the entire skyline without this burden, then the skyline snippet algorithm SSA to the best of our knowledge is the best choice available.

5 Conclusion

The main motivation of skyline snippets is the fact that in high-dimensional space full-space skyline queries may return too many results for users to

analyze and to make appropriate decisions. In many applications it is sufficient to know only a piece of the full skyline. Furthermore, skyline query evaluation on high-dimensional space is a time and memory consuming task in general. Thus methods to efficiently compute only some fraction, called skyline snippets, are of high practical interest, in particular for the rapidly evolving Web services accessed by mobile Internet.

To enable flexible and universal usage, we have developed a novel skyline snippet algorithm that does not have to rely on specialized pre-existing index structures. Performance benchmarks of our prototype implementation have shown very encouraging results. We could observe that the query response times for skyline snippets computation for our chosen benchmark setting were typically in the range of a few seconds or less, which is what mobile Internet users would like to expect.

As on-going research work we will perform more extended performance benchmarks, investigating e.g. the influence of the different types of preference constructors appearing in a skyline query or the performance impacts that different k-partitions of a skyline snippets might have. Developing heuristics for choosing k-partitions is another challenging question. For this we will introduce additional information on the k-partitions, e.g. as mentioned in [21], where the Telescope algorithm exploits a prioritization of the sub-preferences to zoom into interesting skyline points.

In summary, we consider this skyline snippets approach as one essential building stone towards powerful and flexible Web service access to large structured databases through the mobile Internet.

References

1. Brafman, R.I., Domshlak, C.: Preference Handling: An Introductory Tutorial. *AI Magazine* 30(1) (2008)
2. Kießling, W., Köstler, G.: Preference SQL - Design, Implementation, Experiences. In: *VLDB 2002: Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 990–1001. VLDB Endowment, Hong Kong (2002)
3. Hafenrichter, B., Kießling, W.: Optimization of Relational Preference Queries. In: *ADC 2005: Proceedings of the 16th Australasian Database Conference, Darlinghurst, Australia*, pp. 175–184. Australian Computer Society, Inc. (2005)
4. Chomicki, J.: Preference Formulas in Relational Queries. In: *TODS 2003: ACM Transactions on Database Systems*, vol. 28, pp. 427–466. ACM Press, New York (2003)
5. Preisinger, T., Kießling, W.: The Hexagon Algorithm for Evaluating Pareto Preference Queries. In: *MPref 2007: Proceedings of the 3rd Multidisciplinary Workshop on Advances in Preference Handling (in Conjunction with VLDB 2007)* (2007)
6. Endres, M., Kießling, W.: Semi-Skyline Optimization of Constrained Skyline Queries. In: *ADC 2011: Proceedings of the 22nd Australasian Database Conference*. Australian Computer Society (2011)
7. Pei, J., Jin, W., Ester, M., Tao, Y.: Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces. In: *VLDB 2005: Proceedings of the 31st International Conference on Very Large Data Bases*, pp. 253–264. ACM, New York (2005)

8. Yuan, Y., Lin, X., Liu, Q., Wang, W., Yu, J. X., Zhang, Q.: Efficient Computation of the Skyline Cube. In: VLDB 2005: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 241–252. VLDB Endowment (2005)
9. Tao, Y., Xiao, X., Pei, J.: SUBSKY: Efficient Computation of Skylines in Subspaces. In: ICDE 2006: Proceedings of the 22nd International Conference on Data Engineering, p. 65. IEEE Computer Society, Los Alamitos (2006)
10. Tan, K.-L., Eng, P.-K., Ooi, B.C.: Efficient Progressive Skyline Computation. In: VLDB 2001: Proceedings of the 27th International Conference on Very Large Data Bases, pp. 301–310. Morgan Kaufmann Publishers Inc., San Francisco (2001)
11. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive Skyline Computation in Database Systems. *ACM Trans. Database Syst.* 30(1), 41–82 (2005)
12. Lo, E., Yip, K.Y., Lin, K.-I., Cheung, D.W.: Progressive skylining over Web-accessible databases. *IEEE Transactions on Knowledge and Data Engineering* 57(2), 122–147 (2006)
13. Brando, C., Goncalves, M., González, V.: Evaluating Top-k Skyline Queries over Relational Databases. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 254–263. Springer, Heidelberg (2007)
14. Chan, C.Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: On High Dimensional Skylines. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 478–495. Springer, Heidelberg (2006)
15. Kießling, W.: Foundations of Preferences in Database Systems. In: VLDB 2002: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 311–322. VLDB Endowment, Hong Kong (2002)
16. Kießling, W.: Preference Queries with SV-Semantics. In: Haritsa, J.R., Vijayarman, T.M. (eds.) COMAD 2005: Advances in Data Management 2005, Proceedings of the 11th International Conference on Management of Data, pp. 15–26. Computer Society of India, Goa (2005)
17. Börzsönyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. In: ICDE 2001: Proceedings of the 17th International Conference on Data Engineering, pp. 421–430. IEEE Computer Society, Washington, DC, USA (2001)
18. Pei, J., Yuan, Y., Lin, X., Jin, W., Ester, M., Liu, Q.: Towards Multidimensional Subspace Skyline Analysis. *ACM Trans. Database Syst.* 31(4), 1335–5915 (2006)
19. Morse, M., Patel, J.M., Jagadish, H.V.: Efficient Skyline Computation over Low-Cardinality Domains. In: VLDB 2007: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 267–278. VLDB Endowment (2007)
20. Endres, M., Kießling, W.: Semi-Skylines and Skyline Snippets. Technical Report 2010-1, Institute of Computer Science. University of Augsburg (2010)
21. Lee, J., You, G.w., Hwang, S.w.: Telescope: Zooming to Interesting Skylines. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DAS-FAA 2007. LNCS, vol. 4443, pp. 539–550. Springer, Heidelberg (2007)

Evaluating Top-k Algorithms with Various Sources of Data and User Preferences

Alan Eckhardt¹, Erik Horničák¹, and Peter Vojtáš¹

Department of Software Engineering, Charles University in Prague,
Prague, Czech Republic

{eckhardt,vojtas}@ksi.mff.cuni.cz, ejojejo@gmail.com

Abstract. Our main motivation is the data access model and aggregation algorithm for middleware by R. Fagin, A. Lotem and M. Naor. They assume data attributes in a variety of repositories ordered by a grade of attribute values of objects. Moreover they assume the user has an aggregation function, which eventually qualifies an object to top-k answers.

In this paper we adopt a model of various users (there is no single ordering of objects in repositories and no single aggregation) with user preference learning algorithm on the middleware side. We present a new model of repository for simultaneous access by many users. The model is an extension of original model of Fagin, Lotem, Naor. Our solution is based on a model of fast learning of user preferences from his/her reactions. Experiments are focused on the performance of top-k algorithms (both TA and NRA) using data integration on an experimental prototype of our solution. Cache size, network latency and batch size were the features studied in experiments.

1 Introduction

Our main motivation is the data access model and aggregation algorithm for middleware by R. Fagin, A. Lotem and M. Naor [1]. The model can be viewed as data integration from various sources, similar to [2]. We assume that the data are identified by a unique id.

A typical motivation example from [1] reads as: User wants to get information about NY restaurants. The user has an aggregation function that gives a score to each restaurant based on how good it is, how inexpensive it is, and how close it is. In this example, the Zagat-Review web site gives ratings of restaurants, the NYT-Review web site gives prices, and the MapQuest web site gives distances. Other examples deal with multimedia search or with a meta search engine. In all of these cases either data (like colour) or user preferences are inherently fuzzy.

They assume data attributes in a variety of repositories ordered by a grade of an object (accessed either sequentially or directly by id - called random access). Moreover they assume the user has an aggregation function which eventually qualifies an object to top-k answers. For such models, they give optimal algorithms for finding top-k objects according to the cost of sequential and direct access. The optimality was proved in [1].

In this approach, we distinguish between data distributed by rows and data distributed by columns. We understand FLN model as a model of data distributed by columns. This model can be seen as a model of semantic web service integration. The semantics is needed because of universal identification of objects across data providers, allowing easy integration of data attributes. Data from different providers can be aggregated in this way, “from the bottom”, in contrast to the distribution by rows, where the data has to be divided “from the top”.

In this paper we adopt a model of various users (there is no single ordering of objects in repositories and no single aggregation). In our example it can happen that the user does not want the cheapest restaurant neither the closest - maybe he/she looks for an expensive restaurant in an attractive part of the city with good parking possibilities. For this we assume we have a user preference learning algorithm on middleware side.

Basic property, on which relies the optimality of FLN algorithm, is the monotonicity of objects accessed with regard to user preference ordering. For this, we assume on the server side an index which makes it possible to access objects in a monotone way (assuming user particular attribute preferences are simple).

Main contributions of this paper are experiments with a combination of user preference learning, a model of server with variable cache size and batch size and evaluation of top-k algorithms performance depending on cache size, batch size and network latency. In the distributed environment, the possibility of direct access to objects can be realized by linked data technology, e.g. by identifying object by an URI.

2 Preference Querying

Basic idea of the Fagin, Lotem, Naor model is preserved, we assume that objects are repeatedly appearing in different repositories and can be accessed in a monotonic way with regard to user preference on attributes (offered by the respective server) and aggregated on the client side.

2.1 User Preferences

Monotone access to data is in our case replaced by an index on attribute values (B). In this case the fuzzy set with particular user preference serves as the navigation in the index. To make this scenario realistic, we assume that these functions are simple and can be sent when invoking the server.

We basically recognize four types of preferences over numerical attributes. Either it is ascending, as on the top left image in Figure 1 for attribute MPix (of a digital camera), or descending, as on the top right image in Figure 1 for attribute Price.

Then, a little bit more complicated, it can take a form of a hill - on the bottom left image in Figure 1. Last is valley preference, represented on the bottom right image in Figure 1. This type of preference is maybe the least frequent.

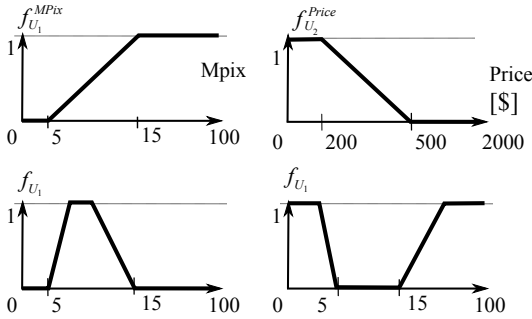


Fig. 1. Four basic types of preference over numerical domains

An example may be the attribute Display, when the user wants either a small notebook for travelling or a big one as a desktop replacement.

All these types contain some important points - we will refer to them as to “points of change”. The normalisation fuzzy set changes the slope at these points. For example, on the top left image in Figure 1, the values 5 and 15 are the points of change.

These four basic types of preferences capture in our view the most common types of user’s ordering of numerical domains and our preference learning algorithm provides such representation. Of course, they are not fixed - for example the ideal value for Hill preference is variable depending on the user. The values in which the preference changes, depends on the user in the same way as the actual shape of preference.

We must note here again that the order is what is important in preferences. So the actual shape between the points of change is not the key point. For simplicity, the functions in figures are piece-wise linear, but they could have arbitrary shape of the curve. The shape will capture how quickly the preference changes between the points of change. Actually, we can imagine smooth curves without any point of change, but such complicated curves will be hard to learn, to obtain from the user and/or to represent analytically. That is why we restrict our model to piece-wise linear functions.

2.2 Top-k Queries

Top-k queries are queries that do not necessarily return the whole result set; the user is often not even capable to see the whole data set. Only top k results are sufficient. Therefore, the order of the result set is important. A most known example are the results of a search engine - the whole set of answers is enormously big beyond the capacity of the reader to process all the results, so the best answers are at the top of the list. The main task of top-k queries is to order all the results so that the most interesting results are at the top. In our example of goods search, this means that the most preferred objects are at the top, so the user can make use of the recommendation.

Algorithms for the processing of the top-k queries need to have m lists of all objects, each sorted according to the preference of an attribute. How this preference is obtained is not considered in the top-k queries processing, which investigates the optimality and the speed of the query answering. Our model of user preferences provides the preferences for each attribute, which are not easily obtained using other models that are less explicitly divided into two steps - e.g. multilayer perceptrons, SVM, decision trees...

Top-k queries were investigated e.g. in [14,5,6,7]. We have examined this area in [3,8].

Having the preferences for each attribute separately can also benefit in efficient query answering using indexes. Even a simple B^+ tree allows finding attribute value in logarithmic complexity.

2.3 Source of Preference Specification

In [9] we proposed PrefWork for learning user preference models from user ratings. These preference models can be used as specification of user preferences for top-k queries, without the obligation for the user to specify his/her preferences for every attribute and attribute value individually.

In the following text, we assume that f_i and @ come from PrefWork.

2.4 Top-k Algorithms

First, we fix the notation used in the paper. All objects p belong to the set of objects P , $|P| = N$. Each $p \in P$ have m attributes $p.a_1, \dots, p.a_m$. We have user fuzzy set $f_U^i : D_{A_i} \rightarrow [0, 1]$ for each attribute - a piece-wise linear function from PrefWork, $@_U : [0, 1]^m \rightarrow [0, 1]$ is also given. User preference function is $@_U(f_U^1(p.a_1), \dots, f_U^m(p.a_m))$ or we will abbreviate the notation and write only $@_U(p)$. The task of top-k algorithm is to find top k objects from P with maximal $@_U(p)$.

The basic setting is that we have m servers S_1, \dots, S_m , each providing the data about one attribute, ordered by the user preferences. In reality, each server may provide more than one attribute, but for the sake of clarity, we use only the simple case.

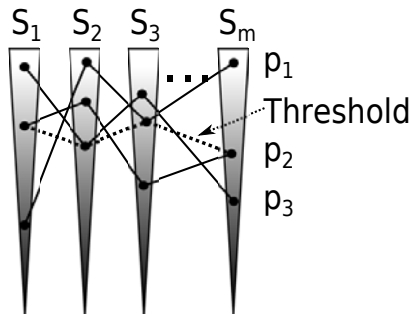


Fig. 2. Servers with objects and threshold

An example of lists and objects at different positions in each server is in Figure 2. Note that object Threshold is a virtual object and will be defined in Section Threshold algorithm.

There are two possible types of accesses to the data - either sequential access that takes the next object id and its attribute value from the top of the ordered list on the server, or random access that returns the attribute value of a given object id. The actual position of the algorithm on the server S_i is denoted as $pos_U(S_i)$. $S_i(x)$ means the object at the position x in the server S_i .

In this section, we briefly describe three basic algorithms for top-k query processing.

Naive Algorithm. The most simple algorithm - it processes all servers S_1, \dots, S_m , gets all the attribute values for all objects, computes $@_U(p)$ for every object p and orders the whole set P . Finally, it returns the top-k objects. This is highly inefficient and we will not deal with it in further text.

Threshold Algorithm. Threshold algorithm (TA) works in steps. In each step, it uses sequential access for a server S_i and gets the object id and the attribute value from the top object p or a batch of id's. Then, it uses random access to the rest of the servers to get the attribute values of the object p .

Algorithm 1. Threshold algorithm (User U)

```

1:  $TOPK = \{\}, depth = 0$ ;
2:  $FuzzySet fs[] = PrefWork.getPreference(U)$ ;
3: while TRUE do {infinite loop}
4:    $depth++$ ;
5:   for all  $i = 1, \dots, m$  do
6:      $[o, o.a_i] = Server_i.sequentialAccess(fs[i], depth)$ ;
7:      $t.a_i = o.a_i$ ; {Update threshold}
8:     for all  $j = 0..m, j \neq i$  do {Get all attributes using random access}
9:        $o.a_j = Server_j.randomAccess(fs[i], o)$ ;
10:    end for
11:    if  $@_U(o) > TOPK[k]$  then
12:       $insert(TOPK, o)$ ; {insert  $o$  on the right position in TOPK}
13:    end if
14:    if  $@_U(TOPK[k]) > @_U(t)$  then {k-th object is better than threshold}
15:      return  $TOPK$ ; {Return TOPK}
16:    end if
17:  end for
18: end while

```

In this way, we know all the attributes of all the objects. The servers are processed successively, i.e. in the first step, TA takes the batch of objects from the server S_1 , in the second step from the server S_2 , etc. In the $m + 1$ step, TA takes the object from S_1 again.

The notion of threshold is defined as follows: threshold t is virtual object, with attribute values corresponding to the actual position of TA in the indexes and navigation with regard to f_U^i . I.e. $t = S_1(pos_U(S_1), \dots, S_m(pos_U(S_m)))$. TA needs a list $TOPK$ of top-k objects found so far. Threshold algorithm is described in Algorithm 1.

Note that on the line 7, $@_U(t)$ is decreased, because the attribute value $t.a_i$ is replaced with a less preferred attribute value. Functions *sequentialAccess* and *randomAccess* are implemented trivially using B^+ tree index. More details of TA can be found in [1].

Algorithm 2. NRA algorithm (User U)

```

1: DISC = CAND = {}, depth = 0, counter = 0;;
2: FuzzySet fs[] = PrefWork.getPreference(U);
3: while TRUE do {infinite loop}
4:   depth++;
5:   for all i = 1, ..., m do
6:     counter++;
7:     [o, o.a_i] = Server_i.sequentialAccess(fs[i], depth);
8:     if o ∈ DISC then
9:       continue;
10:    end if
11:    update(CAND, o, i, o.a_i); {Update value a_i of o in CAND}
12:    if counter mod b ≠ 0 then {resort only every b steps}
13:      continue;
14:    end if
15:    resort(CAND); {Resort CAND using b(o) and w(o)}
16:    if o ≠ CAND[k] then {o is not k-th object}
17:      continue;
18:    end if
19:    for all j = k + 1, ..., N do
20:      if b(CAND[j]) ≤ w(CAND[k]) then
21:        add(DISC, CAND[j]);
22:        discard(CAND, j);
23:      end if
24:    end for
25:    if b(CAND[k + 1]) < w(CAND[k]) then {k-th is better than k+1-th object}
26:      return CAND[1 - k]; {Return top k - the first k objects in CAND}
27:    end if
28:  end for
29: end while

```

3P-NRA Algorithm. 3P-NRA was proposed in [10] as extension of NRA algorithm from [1].

Basic idea of NRA is not to use random access, since it is much more costly than the sequential access and sometimes it is not available. Since the data returned by server S_i is ordered by the preference of one attribute, an object o

is in different positions in all servers. When doing only sequential access, it may occur that we do not know an attribute value of an object. That is the reason we have to have a lowest possible rating $w_U(o)$ (as worst) and the highest possible rating $b_U(o)$ (as best). The known attributes of o give us an idea about how good is the object, but the overall rating $@_U(o)$ cannot be determined until we know all the attribute values of o . We have $w_U(o) \leq @_U(o)$ and $@_U(o) \leq b_U(o)$.

The boundary ratings are computed by supposing the worst, resp. the best possible case. I.e., we suppose that for $w(o)$, $f_U^i(o) = 0$ for all unknown attribute values a_i (and $f_U^i(o) = 1$ for $b(o)$).

NRA needs to have a buffer CAND (as candidates) of objects already seen. They are candidates to get to top k objects. The size of the buffer is bounded only by N , the size of the database, because we cannot be certain if an object is in the top k or not, unless under special circumstances. CAND will be sorted according to the lower bound $w(o)$, in case of a tie according to the $b(o)$. The k -th object of CAND is of special interest, we denote it as $k - th$ and $k+1$ -th object as $k + 1 - th$.

Another structure needed is the list of “discarded” objects DISC.

NRA also proceeds in steps, as described in Algorithm 2. Note that the top k objects may be estimated quickly, but the certainty that they are really the top k is often reached much lately.

On the lines 19-24, we discard all objects without the potential to get into the CAND set. This potential is represented by $b(o)$. The whole algorithm stops in step 25, when the $k+1$ -th object is certain to be worse than the k -th object.

On the line 15, the whole set CAND is sorted according to $b(o)$. This operation is very slow for large datasets. We solved this issue by sorting only every b steps. How big b should be is examined in Experiments in Section 4.

3 System Architecture

In this paper, we propose architecture for top- k queries over different sources of data, originally developed as master thesis [11]. The system is divided into client and server side, which are separated by the internet. Note that the query engine may be used by more than one user and has to handle different preferences.

As the server part only implements B^+ index from [3] and provides two methods *sequentialAccess* and *randomAccess*, we omit the server description. The accesses have three parameters - the user preference fuzzy set, encoded in XML by web service client, the offset of the search and the number of objects to retrieve. A sample call looking like this $S_i.randomAccess(fuzzySet, 100, 10)$ would return 10 objects, starting at the 100-th object. The objects are sorted according to *fuzzySet*. We omitted the batch size in description of algorithms for the sake of clarity.

3.1 Client Architecture

The client side is focused on preference handling and performing the top- k algorithms. It also caches downloaded data from the server in batches for faster processing. The architecture is presented in Figure 3.

The cache is implemented as a queue. Top-k algorithm pulls objects from the queue and another process gets data from the server and puts them into the queue. When the size of the queue drops below a predefined threshold, new data are obtained from the server.

Direct accesses cannot be cached because of their nature. It is impossible to predict which will be the next object id required for direct access. Therefore direct access communicates directly with the server, resulting into a slower operation.

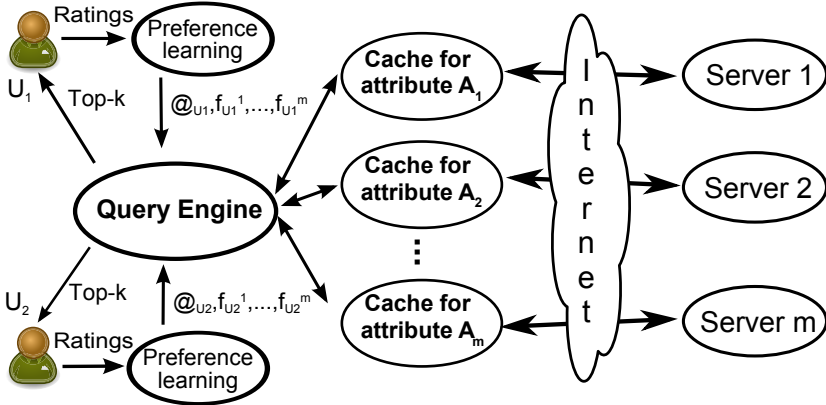


Fig. 3. The overall architecture

The interesting part is where the preferences come from. In previous works [12,13] we proposed framework PrefWork for preference learning and various heuristics. The main benefit is that the preferences are learned in the form of weighted average (corresponds to $@_U$) and piece-wise linear functions for the preference of numerical attributes. This preference models can be directly used by the top-k algorithms and indexing with B^+ tree.

4 Experiments

An experimental prototype to test algorithms 1 and 2 was implemented. The experiments designed to study the dependence of network latency and the speed of the both types of top-k algorithms are described. The network latency was simulated by virtual server with defined pause between the query and the answer. In this way, it was possible to consistently measure the influence of the delay on the algorithm performance. The true network latency was too small and difficult to predict in our setting. The time required by the algorithm to finish for various settings was measured.

The experiments were performed on artificial database with 100 000 objects and 4 attributes. k was set to 10. The preference aggregation $@$ was weighted average with randomly associated weights. The fuzzy sets for attributes were also

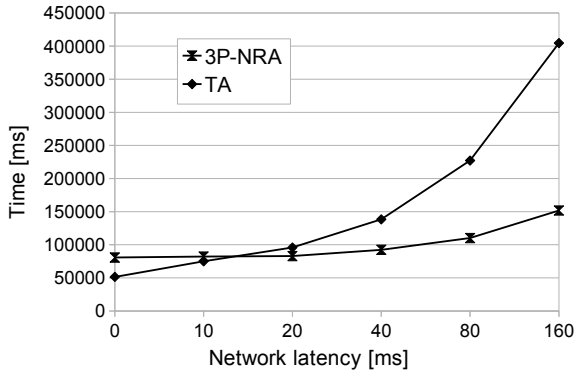


Fig. 4. Time affected by the latency of network

randomly generated. However, the preference model was fixed for all algorithms and all other experiment settings.

The motivation for these experiments is to re-evaluate already known algorithms within previously disregarded conditions, such as the network latency, cache size or batch size. The cache is useful for a certain length of latency and for certain cache sizes.

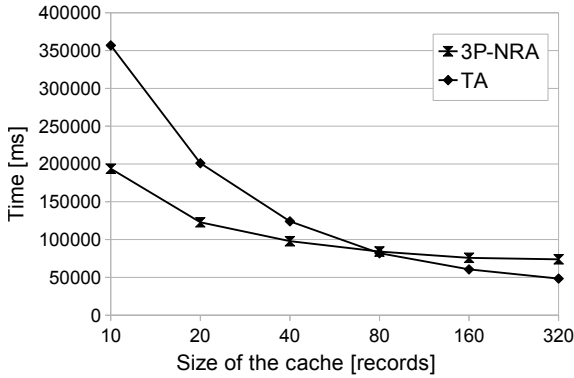


Fig. 5. Time affected by the size of the cache

The results for measuring the time for various network latencies are in Figure 4. We can see that for small latencies, TA is a little bit better, but as the latency increase, NRA scales much better, resulting into a half time for the highest latencies. Note that the latencies are in logarithmic scale. Cache size was 100 objects.

The results for measuring the time for various size of the cache on the client are in Figure 5. The cache influences more TA - with the increase in cache size, the increase in speed is more pronounced. This may be the results of lower number of slow random accesses. Note that the cache sizes are in logarithmic scale. The network latency was set to 10ms.

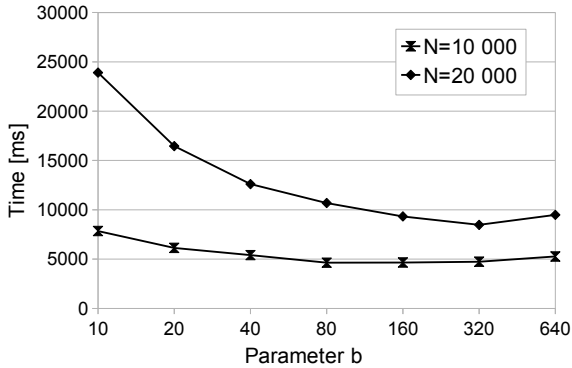


Fig. 6. Time affected by the batch size for 3P-NRA and small data

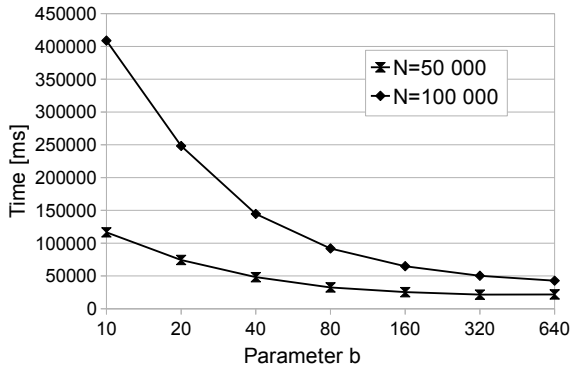


Fig. 7. Time affected by the batch size for 3P-NRA and large data

The results for measuring the time for various sizes the batch size for NRA algorithm are in Figures 6 for small sizes of database and 7 for larger sizes. The figures are separate for better reading.

The results are interesting for small sizes of database - for largest batches b , the time is *worse* than for middle sized batches ($b \approx 100$ objects). This is because the reordering and discarding of objects from the buffer speeds up the succeeding computation. If this is done sparsely, the NRA performs worse than using a more frequent discarding.

For large databases, the results are clear - the larger the batch is, the lower is the time.

5 Conclusion

We have proposed a top-k algorithm using data integration connected with preference learning algorithm, which together provide a complete recommendation

framework. The user preferences are learned from user's ratings or behaviour, and these preferences are used to retrieve top k most preferred objects.

The framework was evaluated with focus on the data stored on different servers, measuring how the latency affects the effectiveness of the algorithms.

5.1 Future Work

In future, it would be nice to be able to evaluate algorithms on real users and real-world data setting. Also some heuristics specific to data integration may prove themselves useful.

We have proposed in [14] a way of learning preferences that depends on the value of another attribute. This type of preference would require a change of the communication between servers.

There is also a possibility for incremental Threshold algorithm that would alter the result set immediately when the user provides a new feedback for PrefWork. PrefWork provides a new preference model to the top-k algorithm and the existing result set can be altered only a little bit. We assume that the preference model does not change too much and only small changes in top-k results would be necessary.

Another possible extension would be a more intelligent cache optimization for real time adjustment of network latency for a particular service.

Acknowledgements. The work on this paper was supported by Czech projects MSM 0021620838, SVV-2010-261312, GACR 202-10-0761, GACR 202-11-0968 and GACR 201-09-0990.

References

1. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: Proceedings of Twentieth ACM Symposium on Principles of Database Systems (PODS 2001), pp. 102–113. ACM, New York (2001)
2. Music Event Explorer - meex, <http://swa.cefriel.it/meex>
3. Eckhardt, A., Pokorný, J., Vojtáš, P.: A system recommending top-k objects for multiple users preferences. In: Martin, T. (ed.) 2007 IEEE Conference on Fuzzy Systems, London, United Kingdom. IEEE Fuzzy systems, pp. 1101–1106 (2007)
4. Chomicki, J.: Preference queries. Journal CoRR cs.DB/0207093 (2002)
5. Ilyas, I.F., Aref, W.G., Elmagarmid, A.K.: Supporting top-k join queries in relational databases. In: VLDB 2003: Proceedings of the 29th International Conference on Very Large Data Bases, pp. 754–765. VLDB Endowment (2003)
6. Güntzer, U., Balke, W.T., Kießling, W.: Optimizing multi-feature queries in image databases. In: Proceedings of the Twenty Sixth Very Large Databases (VLDB) Conference, Las Vegas, pp. 419–428 (2001)
7. Shmueli-Scheuer, M., Li, C., Mass, Y., Roitman, H., Schenkel, R., Weikum, G.: Best-effort top-k query processing under budgetary constraints. In: Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 928–939. IEEE Computer Society, Washington, DC, USA (2009)

8. Eckhardt, A., Pokorný, J., Vojtáš, P.: Integrating user and group preferences for top-k search. In: Tjoa, A.M., Wagner, R.R. (eds.) *Database and Expert Systems Applications, Regensburg, Germany*, pp. 317–322. IEEE, Los Alamitos (2007)
9. Eckhardt, A., Vojtáš, P.: Considering data-mining techniques in user preference learning. In: *2008 International Workshop on Web Information Retrieval Support Systems*, pp. 33–36 (2008)
10. Gursky, P.: Towards better semantics in the multifeature querying. In: Snásel, V., Richta, K., Pokorný, J. (eds.) *DATESO. CEUR Workshop Proceedings*, vol. 176. CEUR-WS.org (2006)
11. Horničák, E.: Preference querying, indexes, optimization. In: *Slovak. Master thesis, Charles University in Prague, Charles University, Czech Republic* (2011)
12. Eckhardt, A.: Inductive models of user preferences for semantic web. In: Pokorný, J., Snásel, V., Richta, K. (eds.) *DATESO 2007. CEUR Workshop Proceedings*, vol. 235, pp. 108–119. Matfyz Press, Praha (2007)
13. Eckhardt, A., Vojtáš, P.: Combining various methods of automated user decision and preferences modelling. In: Torra, V., Narukawa, Y., Inuiguchi, M. (eds.) *MDAI 2009. LNCS*, vol. 5861, pp. 172–181. Springer, Heidelberg (2009)
14. Eckhardt, A., Vojtáš, P.: Learning user preferences for 2CP-regression for a recommender system. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) *SOFSEM 2010. LNCS*, vol. 5901, pp. 346–357. Springer, Heidelberg (2010)

Efficient and Effective Query Answering for Trajectory Cuboids

Elio Masciari

ICAR-CNR

Institute for the High Performance Computing of Italian National Research Council
masciari@icar.cnr.it

Abstract. Trajectory data streams are huge amounts of data pertaining to time and position of moving objects generated by different sources continuously using a wide variety of technologies (e.g., RFID tags, GPS, GSM networks). Mining such amounts of data is challenging, since the possibility to extract useful information from this peculiar kind of data is crucial in many application scenarios such as vehicle traffic management, hand-off in cellular networks, supply chain management. Moreover, spatial data streams poses interesting challenges both for their proper definition and acquisition, thus making the mining process harder than for classical point data. In this paper, we address the problem of trajectory data streams On Line Analytical Processing, that revealed really challenging as we deal with data (trajectories) for which the order of elements is relevant. We propose an end to end framework in order to make the querying step quite effective. We performed several tests on real world datasets that confirmed the efficiency and effectiveness of the proposed techniques.

1 Introduction

In this paper we deal with trajectory data, that are data logs pertaining time and the position of moving objects (or groups of objects) that could be generated in a wide variety of applications, such as GPS systems or weather monitoring systems. Trajectory data carry information about actual position and timestamp of moving objects that are in general high sized and redundant. Indeed, in systems monitoring object movements users are not interested in the point sequences traced by objects, instead they are interested in “range analysis”, e.g. for traffic management purposes an interesting query could be “how many cars entered in Rome downtown in August”. In this respect, we perform a logical compression step by splitting the search space in regions having a suitable granularity and represent them as areas tagged by a timestamped symbol along with summary information about distance, velocity and so on. The sequence of regions (symbols) define the trajectory traveled by a given object. The above representation of trajectory data (i.e. region based instead of a sequence of multidimensional points) heavily reduce the size of input data and is a natural basis for warehousing trajectory data as will be shown in this work. A unique feature of trajectory

warehouses is that we need to provide *path information* since they define how items flow into the system being monitored. So we need to provide a mechanisms for representing paths suitably in order to make the querying process quite efficient. On the other side we still need to make available information about the item hierarchies (in our example the user may want to know summary information about generic moving vehicles or *specific* ones like cars). Thus, in order to build a trajectory warehouse we need to provide an efficient method to answer to path queries and classical aggregate queries. As a key intriguing feature of our application scenario we need to take into account the streaming nature of such data: they flow continuously thus making classical approaches for warehousing unpractical. The first step we perform is the logical compression of input data performed by partitioning the original trajectories in regions. In this paper we exploit *Principal Component Analysis*(PCA) [4] to effectively identify preferred directions for trajectories thus reducing the degree of uncertainty since focusing on a small number of directions we can tune the granularity of regions to be defined on data. Partitioning the search space in a “guided” way will help us to solve the representation problem when preprocessing data during the Extraction, Transformation and Loading phase (*ETL*) since we can exploit a differential regioning, i.e. we can use a finer granularity for regions along the principal directions while we can use a coarser one for “suburban” regions. As a strong motivation for our choice we recall that PCA values (and consequently induced regions) can be incrementally computed, the latter is a mandatory requirement for stream based systems. Data stream warehousing brings many challenges not encountered in database warehousing, because of the real-time response requirement and the presence of bursty arrivals and concept shifts (i.e., changes in the statistical properties of data). In order to cope with such challenges, the continuous stream has to be partitioned into windows [9], thus reducing the size of the data that need to be stored and queried. To tackle this problem we build a set of cuboids having different granularities that will be continuously fed by incoming streams. In particular, we compute finer grain cuboids for data pertaining the most fresh windows while data pertaining to older windows are aggregated using existing cuboids thus saving storage space and computational time. Note that spending more details for most recent data is a common assumption in temporal data management systems. Finally, in order to further save computational time when querying data we exploit a clever numerical representation of paths by using a prime numbers based scheme exploiting the Chinese Remainder Theorem and the Fundamental Theorem of Arithmetic. In particular, we devise a path encoding scheme for processing tracking queries and path oriented queries efficiently.

2 ETL for Trajectories

In this paper we tackle the problem of data warehousing from large corpus of trajectory data streams. While for transactional data a tuple is a collection of features, a trajectory is an ordered set (i.e., a sequence) of timestamped points.

Trajectory data are usually recorded in a variety of different formats, and they can be drawn from a continuous domain. We assume a standard format for input trajectories, as defined next. Let P and T denote the set of all possible (spatial) positions and all timestamps, respectively. A trajectory Tr of length n is defined as a finite sequence s_1, \dots, s_n , where $n \geq 1$ and each s_i is a pair (p_i, t_i) where $p_i \in P$ and $t_i \in T$. We assume that P and T are discrete domains. For continuous locations a viable approach is to partition the space into regions to map the initial locations into discrete regions labeled with a timestamped symbol. The problem of finding a suitable partitioning for both the search space and the actual trajectory is a core problem when dealing with spatial data. Every technique proposed so far, somehow deals with regioning and several approaches have been proposed such as partitioning of the search space in several regions of interest (*RoI*) [1] and trajectory partitioning [6,7] by using polylines. In this section, we describe the application of Principal Component Analysis (*PCA*) [4] in order to obtain a better partitioning. Indeed, *PCA* [4] finds *preferred* directions for data being analyzed and thus we can exploit this information for better accuracy in data warehousing since we point out that our goal is to allow efficient storage and querying of data, so it is likely that less frequently crossed regions are not relevant to the analysts. Once detected the preferred directions we perform a partition of the search space along these directions as will be explained in next sections. Many tools have been implemented for computing *PCA* such as [4], in our framework due to the streaming nature of data we exploited the incremental *PCA* (*IPCA*) algorithm proposed in [12]. The algorithm is a new *IPCA* method based on the idea of a singular value decomposition (*SVD*) updating algorithm, namely an *SVD* updating-based *IPCA* (*SVDU-IPCA*) algorithm. For this *SVDU-IPCA* algorithm, it has been mathematically proved that the approximation error is bounded. The latter is a relevant feature since the quality of regioning heavily relies on the quality of *IPCA* results. Due to space limitations we avoid a detailed description of the mathematical steps implemented in our prototype.

3 Encoding Paths for Efficient Counting and Querying

A great problem with trajectory warehousing is to control the exponential explosion of candidate trajectories since the order is relevant. In the pre-elaboration step we exploited regioning to describe the trajectories in a succinct manner, this is a good solution especially compared w.r.t. other approaches (such as [7]) that represent trajectories as segments, since this could cause that the candidate will not belong to the actual dataset and this is obviously a great limitation for warehousing purposes. Moreover, the regioning step heavily reduce the dataset size so the number of regions we have to deal with is of hundreds of regions instead of thousands of points. Since our approach is stream oriented we need to be fast while counting trajectories and (sub)paths. To this end Prime numbers exhibit really nice features that for our goal can be summarized in the following two theorems and has been exploited for similar purposes for RFID tags encodings [5].

Theorem 1 (The Unique Factorization Theorem). *Any natural number greater than 1 is uniquely expressed by the product of prime numbers.*

As an example consider the trajectory $T_1 = ABC$ crossing three regions A, B, C . We can assign to regions A, B and C respectively the prime numbers 3, 5, 7 and the position of A will be the first ($pos(A) = 1$), the position of B will be the second ($pos(B) = 2$), and the position of C will be the third ($pos(C) = 3$). Thus the resulting value for T_1 (in the following we refer to it as P) is the product of the three prime numbers, $P_1 = 3 * 5 * 7 = 105$ and there does not exist the product of any other three prime numbers that gives as results 105.

As it is easy to see this solution allows to easily manage trajectories since containment and frequency count can be done efficiently by simple mathematical operations. Anyway, this solution does not allow to distinguish among $ABC, ACB, BAC, BCA, CAB, CBA$, since the products for these trajectories is always 105. To this end we can exploit another fundamental theorem of arithmetics.

Theorem 2 (The Chinese Remainder Theorem). *Suppose that n_1, n_2, \dots, n_k are pairwise relatively prime numbers. Then, there exists W (we refer to it as witness) between 0 and $N = n_1 \cdot n_2 \cdot \dots \cdot n_k$ solving the system of simultaneous congruences: $W \% n_1 = a_1, W \% n_2 = a_2, \dots, W \% n_k = a_k$*

In the above example for trajectory T_1 the witness W_1 is 52 since $52 \% 3 = 1 = pos(A)$ and $52 \% 5 = 2 = pos(B)$ and $52 \% 7 = 3 = pos(C)$.

From the above property it follows that in order to fully encode a trajectory (i.e. keeping the region content and the order) it suffices to store two numbers its prime number product and its witness. As a nice consequence to obtain any information we can test with the maximum efficiency containment relationships (a simple division) and order checking (a sequence of divisions). In order to assure that no problem will arise in the encoding phase and witness computation we assume that the first prime numbers we choose for encoding is greater than the trajectory size. So for example if the trajectory length is 3 we encode it using prime numbers 5, 7, 11. A devil's advocate may argue that multiple occurrences of the same region violate the injectivity of the encoding function. To this end the following example will clarify our strategy.

Consider the following trajectory $T_2 = ABCAD$, now we have the problem for encoding A since it appear twice in the first and fourth position. so we need to encode that the encoding value of A is such that we can say that both $pos(A) = 1$ and $pos(A) = 4$ (we do not need two separate value since the region is the same and we are interested in the order). Assume that A is encoded as $(41)_5$ this meaning that A occurs in the first and fourth position (recall that the trajectory length is 5). The decimal number associated to it is $A = 21$, and we chose as the encoding for $A = 23$ that is the first prime number greater than 21. Now we encode the trajectory using $A = 23, B = 7, C = 11, D = 13$ thus obtaining $P_2 = 23023$ and $W_2 = 2137$ (since the reminder we need for A is 21). As it easy to see we are still able to properly encode even trajectories containing cycles. Finally, one may argue that the size of prime numbers could be large, however in our case it is bounded since the number of regions is small. Moreover, there

are approaches using prime numbers encoding for XML documents dealing with million of nodes [11]. We now give the definition of encoding, first we formalize the region encoding.

Definition 1 (Region Encoding). *Given a set $R = \{R_1, R_2, \dots, R_n\}$ of regions, a function enc from R to \mathcal{P} (the positive prime numbers domain) is a region encoding function for R .*

Based on the above definition we can define the trajectory encoding.

Definition 2 (Trajectory Encoding). *Let $T_i = R_1R_2 \dots R_n$ be a regioned trajectory. A trajectory encoding ($E(T_i)$) is a function that associates T_i with a pair of integer numbers $\langle P_i, W_i \rangle$ where $P_i = \prod_{1..n} enc(R_i)$ and W_i is the witness for P_i .*

Once we encode each trajectory as a pair $E(T)$ [1] we can store trajectories in a search binary tree making the search, update and verification operations quite efficient since at each node we store the $E(T)$ pair. It could happen that there exists more than one trajectory encoded with the same value P . In this case, we store once the P value and the list of witnesses saving space for pointers and for the duplicate P 's value. Consider the following set of trajectories along with their encoding values (we used region encoding values: $A = 5, B = 7, C = 11, D = 13, E = 15$): $(ABC, \langle 385, 366 \rangle)$, $(ACB, \langle 385, 101 \rangle)$, $(BCDE, \langle 15015, 3214 \rangle)$, $(DEC, \langle 2145, 872 \rangle)$. ABC and ACB will have the same P value (385) but their witnesses are $W_1 = 366$ and $W_2 = 101$, so we are always able to distinguish them.

We insert the encoded trajectories in a binary tree associated to the above set of trajectories. The insertion, remove and update operations will have the usual meaning.



Fig. 1. Binary tree for encoded trajectories

In order to test if a candidate trajectory is present in the tree we need simply to find if its encoding value is present along with its witness. As a desirable side effect, for different length trajectories we can test if a (sub/super)trajectory is frequent simply testing if its encoding value is a factor of some node in the tree. This property naturally follows from the Apriori property for ordered sets, i.e. if

¹ We use a really efficient implementation for long integers in “C++ Big Integer Library” written and maintained by Matt McCutchen.

<p>Method: <i>checkContainment</i></p> <p>Input: Two trajectories encodings $E(T_1)$ and $E(T_2)$;</p> <p>Output: Yes if $T_1 \in T_2$, no otherwise;</p> <p>Method:</p> <ol style="list-style-type: none"> 1: if $P_1 > P_2$ return NO 2: if $P_2 \% P_1 = 0$ then 3: for each R_i^1 4: if $pos((next(R_i^1))^2) < pos(R_i^2)$ return NO 5: return YES <hr/> <p>Method: <i>updateTree</i></p> <p>Input: a trajectory tree B_T and a new window NEW;</p> <p>Output: the updated version of B_T;</p> <p>Vars: a tree node N;</p> <p>Method:</p> <ol style="list-style-type: none"> 1: for each $T_i \in NEW$ 2: $N = depthSearch(E(T_i))$ 3: if $N \neq null$ 4: $updateFrequency(N)$ 5: else $insertNode(E(T_i))$ 6: if $memoryneeded$ $deleteOlderNode(B_T)$ 7: return B_T
--

Fig. 2. Algorithms for checking containment and update tree

ABC is frequent AB, BC, A, B, C are frequent as well. This property holds since, due to witnesses we can always test the precedence relations.

A simple sub-routine for checking ordered containment. In order to perform efficient trajectory verification we should be able to check containment relationships between trajectories. More in detail, given two trajectories, say T_1 and T_2 , and their encodings $E(T_1)$ and $E(T_2)$ we need to answer this question ‘Is T_1 a sub-sequence of T_2 ?’ To this end we can exploit the mathematical features of encoded trajectories to design a simple and effective algorithm. In the following we denote the index of a region $R_i \in T_i$ as $pos(R_i)$ and the region following R_i in T_i as $next(R_i)$. If a region R_i appear in two different trajectories T_l and T_m we denote it as R_i^l and R_i^m .

The algorithm first check if P value of T_1 is a divider of P value of T_2 , if the latter holds this means that all the regions belonging to T_1 also belongs to T_2 . To test if the containment holds we need to check that every pair of consecutive regions in T_1 are also consecutive in T_2 . Consider again our toy example and suppose to check wether $T_1 = BC$ is contained in $T_2 = ABC$. we have that $P_2 = 385$ and $P_1 = 77$. We test that $P_2 \% P_1 = 0$, now we have that $pos(B^2) = 2 < 3 = pos(C^2)$ thus the answer to the containment test is

YES. The algorithm for checking containment is reported in Figure 2. Since we are interested in monitoring trajectories frequencies, we need to store an additional information at each node, i.e. the frequency of the trajectory stored at that node. Obviously if we have more than one witness for a given P value we will store the frequencies for each witness. Again, doing this we prevent new nodes insertion thus saving memory space. The initial tree is built for the first window W_1 , say it B_T . As new trajectories arise in the stream we continuously update it. To perform this updating step we run the algorithm in Figure 2. For every incoming trajectory $T_i \in NEW$ we search the node N it belongs to by performing a *depthSearch* on B_T of the encoding values for T ($E(T_i)$). If such a node N exists (this means that T_i is frequent) we update its frequency. Function *updateFrequency* updates the frequency of the witness W_i for T_i (we recall that there may exist more than one witness for the same P value). If the trajectory does not exist in the tree we insert the corresponding node (annotating its timestamp). Finally, as the stream flow if we need to release memory we delete the older nodes (i.e. the ones with older timestamps).

4 Warehouse Architecture

In this section we will describe our schema proposal for trajectory warehousing that will exploit our pre-processing strategies shown in Section 3 to perform efficient analysis. Due to the intrinsic complexity of trajectory data the definition of suitable measures is crucial. The pre-elaboration strategies devised in previous sections allows us to transform the raw incoming trajectory streams in a suitable form for loading them into the Fact and Dimension tables of the trajectory Data Warehouse. It is a snowflake schema in order to better managing the spatial features of a trajectory. In particular we need to be able to deal with the *distinct count*, i.e. a trajectory may span over many cells so we need to take it into account when performing OLAP operations. The schema includes a spatial and a temporal dimension for describing the placements and movements of objects (*where is* queries) and time information (*when* queries). Similar measures were exploited in [8], but we point out that in our approach no approximation is introduced. More specifically, the trajectory partitioning step is a lossless operation in trajectory warehouses since we are interested in *aggregate* information so the exact point-based representation of trajectory is ineffective and the prime number scheme exploited in our work allow to keep efficiently and effectively details that would not be available otherwise. The Trajectory Data Warehouse stores aggregations about trajectories belonging to a spatio-temporal cells. Hence, apart from the keys to dimension tables, the fact table also contains a set of measures representing aggregate information. The measures considered in the schema include (we report here only the most intuitive measure) the number of distinct trajectories (*Intersections*), the average traveled distance (*Distance*), the average time interval duration (*Duration*), obviously we can derive additional information about the average velocity and other specific measures, for a particular group of objects (having a certain trajectory shape) moving in a specific spatial area during a specific time period. The

DW picture is not reported here due to space limitations. To feed the warehouse we exploit the ETL processing explained above that revealed quite efficient and effective so as to fill in the measures of the warehouse with the appropriate numeric values for each base cell.

Building Cuboids. Once the logical compression has been performed we can build the Trajectory Data Warehouse starting from a low level data structure: Trajectory Cuboids (TRAC). Cuboids are intended to be disk-resident, summarizing the contents of an encoded trajectory database in a compact yet complete manner while allowing efficient execution of both OLAP and path specific queries. The TRAC Cuboid consists of three tables: 1) *Items*, which stores information about the moving objects, 2) *Collection*, which stores information on trajectories crossing the regions being analyzed, and 3) *Time*, which stores time information. Each dimension in the three tables has an associated concept hierarchy. Indeed, a concept hierarchy is a partial order of mappings from lower levels of abstraction to higher ones. The lowest corresponds to single regions in the data stream itself tracking single objects. The highest dimension as usual in Data Warehouses is *ALL* which represents any value of the dimension. In order to provide fast responses to queries specified at various levels of abstraction, it is important to pre-compute some TRACs at different levels of the concept hierarchies for the most interesting dimensions of three tables. Computing all the possible generalizations is a really expensive operations so partial materialization is a preferred choice. Determining which set of cuboids in a data warehouse to be materialized in order to answer OLAP queries efficiently (given the constraints on storage space and precomputation time) is a challenge extensively studied in the data cube field [3], and the principles are generally applicable to the selective materialization of trajectory Cuboids. A well accepted solution is to compute a set of Cuboids (in our case TRACs) at the minimal interesting level at which users will be interested in inquiring the database, and a small set of higher level structures that are frequently requested and that can be used to quickly compute non materialized TRACs. A TRAC residing at the minimal interesting level will be computed directly from the encoded trajectory window and will be the lowest cuboid that can be queried unless some specific (very unlikely) request at trajectory level. As the stream flows we recompute the basic TRACs and merge the older ones in order to save space while keeping an high informative level. In particular, we can merge cuboids by summarizing information at an higher level. Cuboids are merged at the higher informative level in order to avoid loss of information or wrong values computation. As an example consider two cuboids to be merged whose spatial hierarchy bottom level are respectively *block* and *city*, the resulting cuboid spatial hierarchy will have as bottom level *city*, since we can easily obtain aggregate information about cities in the first cuboid while the opposite does not hold. Merging and collapsing dimensions has been widely used in Data Warehouse field and also for RFID trajectories path management [2], we point out that the distinguishing feature of our approach is the exploitation of our novel encoding strategy that makes our solution quite effective and efficient. Moreover, we are interested in a model that describes the behavior of

<p>Method: <i>BuildTRAC</i></p> <p>Input: a trajectory tree B_T encoding a trajectory window; a set of measures m to be computed;</p> <p>Output: a trajectory cuboid $TRAC$;</p> <p>Vars: a tree node N; a set of dimensions dim at a specified granularity level;</p> <p>Method: 1: for each $N \in B_T$ 2: for each dim_i 3: if $N \neq null$ 4: for each m_i 5: $TRAC = updateMeasure(N, m_i)$ 6: return $TRAC$</p>
<p>Method: <i>updateMeasure</i></p> <p>Input: a node of the trajectory tree B_T; the measure m_i to be computed;</p> <p>Output: a trajectory cuboid $TRAC$;</p> <p>Vars: a cuboid cell C_i at a specified granularity level; an encoding value e for C_i</p> <p>Method: 0: ... 1: if $m_i = Intersections$ 2: for each C_i 3: $e = encodeRegions(C_i)$ 4: if $N.value \% e == 0$ 5: $C_i.Intersections ++$ 6: ...</p>

Fig. 3. *TRAC* Load and Compute Measures Algorithms

objects given a collection of their traveled paths. In order to effectively tackle this problem, we will materialize only cells in the TRACs that contain at least a minimum number of paths (minimum support), this can be done efficiently by simply accessing the frequency stored in B_T . The last is the *Iceberg* assumption and *Iceberg TRACs* can be computed efficiently by using an apriori-like pruning of infrequent cells. We can materialize the cube from low abstraction levels to high abstraction ones. If at some point a low level cell is not frequent, we do not need to check the frequency of any specialization of the cell. The following algorithm is used to feed a TRAC given a set of measures pertaining to it.

We point out that the *Iceberg* property is guaranteed by B_T (the auxiliary tree structure) definition since it is built upon trajectories exhibiting a minimum frequency. Obviously, each measure to be updated could be either algebraic or

holistic, in both case we are guaranteed about the quality of the results since we are focusing on a fixed window of the incoming stream so we can focus on data pertaining to it, thus we have all the information needed for both algebraic and holistic measures. We show in Figure 3 the pseudo code for updating the frequency of a given path involving a set of regions defining a cuboid cell (*Intersections* measure), the code is part of the overall *updateMeasure* method whose complete set of instruction is not reported here due to space limitations.

Herein: *encodeRegions* encodes the regions belonging to the cuboid cell C_i at a specified granularity level. This operation is needed since depending on the chosen granularity level a cell could contain different number of regions, e.g. if the space granularity is set to city instead of county the number of involved regions will be different. As it is easy to see in our framework checking the intersection of a trajectory in a even complex region is done immediately by checking the remainder of a division operation ($\%$ is the modulo operation). As a desirable side effect our encoding strategy guarantees that region counts are always distinct since the input trajectory is encoded as a whole. Moreover, the above algorithm follows the *cell-oriented* approach, i.e. we search for the trajectory portions that lie within the base cells. Finally, due to the easy encoding of the *whole* trajectory we can also perform operations on the trajectory itself (e.g. compute average trajectory length) thus allowing the *trajectory-oriented* approach for querying the cuboids.

5 Experimental Evaluation

In this section we will show the experimental results for our algorithms. We used a well-known dataset containing a huge number of trajectories. This is a GPS trajectory dataset collected in (Microsoft Research Asia) *GeoLife* project [13,14] by 165 users in a period of over two years (from April 2007 to August 2009). This dataset recoded a broad range of users outdoor movements, including not only life routines like go home and go to work but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. Therefore, the dataset represents a severe test for our warehousing framework. Moreover, other public available trajectory dataset are of limited interest for our work since they contain a limited number of constant trajectories. The experiments were ran on a single node of a 12 node cluster Intel Xeon E5520 2,26GHz, 16 GB RAM per node, 1 TB per node. Since we perform data stream warehousing the dataset W being mined is defined as a sliding window over the continuous stream. W continuously moves forward by a certain amount. Each window W either contains the same number of trajectories (*count-based* or physical window), or contains all trajectories arrived in the same period of time (*time-based* or logical window). The window is maintained by adding the new slide (δ^+) and dropping the expired one (δ^-). Therefore, the successive instances of W are referred as W_1, W_2, \dots . The number of trajectories that are added to (and removed from) each window is called its slide size.



(a) Compression performances vs Region Size (b) Compression performances vs Trajectory Length

Fig. 4. Measuring Approach Performances

Storage space saving. As explained in previous section TRACs are exploited for query processing and analysis. The advantage of these data structures is that they collapse many records in the regioned trajectory collection table. Here, we examine the effects of this compression on our data set. We first perform this analysis by considering different region sizes (0.1, 0.2, 0.5, 1 squared kilometers), the initial dataset size is 1.5 Gbytes and we use different windows size for running the experiments, due to space limitations we show the results obtained for a window size of 10000 trajectories since it is the most widely used for practical applications. In the following cuboid sizes are expressed in Gbytes. First, we obtained the results shown in Figure 4 (a) that shows how the regioning step heavily reduce the input size while keeping relevant information (we recall that we are not interested in single point queries). This result is really encouraging since reducing the input size will result in faster OLAP operations.

Trajectory length analysis. We explained in previous sections how TRAC cuboids are computed and in order to exploit the encoding scheme for better performances we can set the length of (sub)trajectories being queried. More in detail, we can state that TRAC will be computed for trajectories of maximum length n . This feature is useful in traffic management systems or people behavioral analysis and so on. Figure 4 (b) shows the input size reduction w.r.t. the maximum trajectory length. It is interesting to note that the size of the compressed input decrease till a certain length then it slightly raise up. This can be understood by observing that when the trajectory length is low we have more encodings of small size and this results in high dataset size reduction. As we increase the allowed trajectory length we will have lesser encoding but the resulting prime number encoding have an higher size even using an efficient floating point representation thus resulting in a small compression gain, however, real life trajectories rarely span over 20 regions. It is worth noticing that also in this case we obtain a remarkable compression especially considering that we fully maintain path information.

6 Conclusion

In this paper we have proposed a novel model for warehousing trajectory data that allows high-level analysis to be performed efficiently and flexibly in multidimensional space. The model is composed of a hierarchy of highly compact

summaries (TRAC Cuboids) of the trajectories data aggregated at different abstraction levels where data analysis can take place. Our approach can be the basis for performing useful mining analysis that remains as a future work.

References

1. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: KDD-Knowledge Discovery in Databases, pp. 330–339 (2007)
2. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing and analyzing massive rfid data sets. In: ICDE-International Conference on Data Engineering, p. 83 (2006)
3. Han, J., Stefanovic, N., Koperski, K.: Selective materialization: An efficient method for spatial data cube construction. In: PAKDD-Pacific-Asia Conference on Knowledge and Data Mining (1998)
4. Jolliffe, I.T.: Principal Component Analysis. Springer Series in Statistics (2002)
5. Lee, C., Chung, C.: Efficient storage scheme and query processing for supply chain management using rfid. In: SIGMOD-ACM Special Interest Group on Management of Data Conference, pp. 291–302 (2008)
6. Lee, J., Han, J., Li, X.: Trajectory outlier detection: A partition-and-detect framework. In: ICDE-International Conference on Data Engineering, pp. 140–149 (2008)
7. Lee, J., Han, J., Whang, K.: Trajectory clustering: a partition-and-group framework. In: SIGMOD-ACM Special Interest Group on Management of Data Conference, pp. 593–604 (2007)
8. Leonardi, L., Marketos, G., Frentzos, E., Giatrakos, N., Orlando, S., Pelekis, N., Raffaetà, A., Roncato, A., Silvestri, C., Theodoridis, Y.: T-warehouse: Visual olap analysis on trajectory data. In: ICDE-International Conference on Data Engineering, pp. 1141–1144 (2010)
9. Li, J., Maier, D., Tufte, K., Papadimos, V., Tucker, P.A.: No pane, no gain: efficient evaluation of sliding-window aggregates over data streams. SIGMOD Record 34(1), 39–44 (2005)
10. Pelekis, N., Theodoridis, Y., Vosinakis, S., Panayiotopoulos, T.: Hermes - A framework for location-based data management. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 1130–1134. Springer, Heidelberg (2006)
11. Lee, M.-L., Wu, X., Hsu, W.: A prime number labeling scheme for dynamic ordered xml trees. In: ICDE - International Conference on Data Engineering
12. Zhao, H., Yuen, P.C., Kwok, J.T.: A novel incremental principal component analysis and its application for face recognition. IEEE Transaction on Systems, Man, and Cybernetics 36, 873–886 (2006)
13. Zheng, Y., Li, Q., Chen, Y., Xie, X.: Understanding mobility based on gps data. In: UbiComp, pp. 312–321 (2008)
14. Zheng, Y., Zhang, L., Xie, X., Ma, W.: Mining interesting locations and travel sequences from gps trajectories. In: World Wide Web, pp. 791–800 (2009)

A Model for Analyzing and Visualizing Tabular Data

Ekaterina Simonenko, Nicolas Spyratos, and Tsuyoshi Sugibuchi

Laboratoire de Recherche en Informatique,
Université de Paris-Sud,
91405 Orsay Cedex, France
{katia,spyratos,buchi}@lri.fr

Abstract. We present a model to visual analysis of tabular data based on functional dependencies, and a Web based tool that supports casual users in doing the following: (a) construct an analytic query visually, in a flexible, interactive manner, (b) visualize the aggregate result in a user selected mode (histogram, pie, etc.), (c) explore the query result by viewing equivalent representations at different aggregation levels or for different parameter values.

Keywords: Data Analysis, Analytic Query, Data Visualization, Visual Interaction.

1 Introduction

In several Web based applications such as e-commerce, e-learning, digital libraries, etc. one needs to display a dense array of information in a small amount of space (such as a screen) in a manner that communicates clearly and immediately. The information displayed usually consists of aggregates of results obtained through analysis of large amounts of data.

For example, consider the case of a digital library, allowing a community of users to share documents in digital form. To access a desired document, a subscriber queries the library catalogue which stores each document's URI together with a description of the document (such as language, topic, author, etc.). The catalogue can be seen as a table, and a query against the catalogue is just a Boolean combination of elementary conditions of the form " $A = a$ ", where A is an attribute and a is a value of that attribute; for example, the following is a query asking for documents in *French* about *Poetry*: ($Language = French$) and ($Topic = Poetry$).

The digital library administration needs to perform usage analysis in order to plan the library's activities. Such analysis usually concerns the ranking of documents along several dimensions (e.g. topic, author, etc.) according to certain indicators (e.g. author nationality or number of hits). However, the results of such analysis can be very large in size, and the only way to make sense out of big volumes of data is to create summaries and to display the summarized results to the analyst in an appropriate visualization mode - ideally, by the analyst

himself. Moreover, the analyst should be able to perform exploratory analysis on the visualized results by changing the summarization level or by viewing different, yet equivalent representations of the results that might reveal new, interesting information.

There are several offerings today by software companies that allow analyzing large volumes of data and visualizing the results [3,4], including some open source software [2]. However all these tools are closely related to the relational model and require some knowledge of the SQL constructs related to data analysis (such as grouping sets, cube, roll up etc.). Moreover, there is no generally accepted approach today to analytic schema design. The existing proposals either define an analytic schema in an ad-hoc way, or are not abstract enough [10].

In this paper, we present a novel approach to modeling the analysis and visualization of tabular data, based on functional dependencies [1]. Given a table T with a set of functional dependencies F , we define an *analysis schema* (or *schema*, for short) to be a set of functional dependency paths with common origin. Based on this simple concept of schema, we present an approach that allows analysts to do the following:

1. construct an analytic query *visually*, in an *interactive* manner;
2. *visualize* the aggregate result in a user selected mode (histogram, pie, etc.);
3. *explore* the query result by viewing equivalent representations at different aggregation levels or for different parameter values selected by the user.

Our approach is implemented as a web-based tool, called *Visual Analyser*, supporting the above functionalities. This tool was developed in the context of the European project KP-Lab [5]. A prominent feature of our tool is that the process of creating a query and receiving its results as well as analyzing and exploring the results in several ways is well integrated and supported by intuitive actions.

The rest of the paper is organized as follows. In Section 2 we give an informal overview of the model and the basic concepts; then in Section 3 we define the formal model in the context of a relational table T satisfying a set of functional dependencies F . In Section 4 we describe our approach to result visualization, as well as the visual interaction between the user and the interface, and present our tool. We conclude with some remarks and perspectives on future work.

2 The Model: Overview and Basic Concepts

Consider a digital library in which each document is identified by its *URI* and described by two attributes: *Topic*, whose values are keywords describing document content (eg. drama, poetry, etc.); and *Hits*, whose values are integers representing the number of accesses to the document. Therefore, we have two functional dependencies $t : URI \rightarrow Topic$ and $h : URI \rightarrow Hits$, as shown by the graph of Figure 1(a) (we need the labels t and h for later reference).

This graph is a first, rudimentary example of what we call analysis schema (or simply “schema”). Its *origin* is the node *URI* which represents the objects of interest, while the two other nodes describe attributes of the objects.

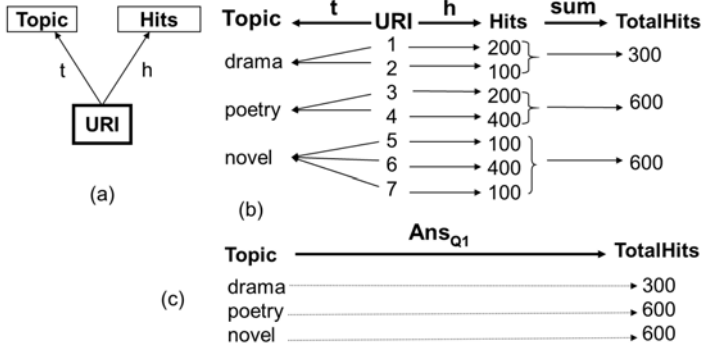


Fig. 1. Example of an application represented by a graph, and a query evaluation

In our approach, we interpret the edges $t : URI \rightarrow Topic$ and $h : URI \rightarrow Hits$ of the schema as function signatures and their extensions as the (current) database. Figure 1(b) shows an example of (current) database. This database represents all documents accumulated so far in the library: it contains 7 document URIs, each associated with its topic and its number of hits through the functions t and h (for simplicity, we represent URIs as integers); for example, document 3 is associated with “Poetry” as topic and with 200 as number of hits (i.e. $t(3) = Poetry$ and $h(3) = 200$).

More generally, we have the following definitions (see [1] for more details):

Definition 1. *Functional schema*

A functional schema is a connected, oriented, acyclic and labeled graph such that:

- there is a single root denoted by O ;
- each node N is associated with a set of values, denoted by $dom(N)$;
- the labels of all arrows are distinct.

Definition 2. *Functional database*

Given a functional schema S , a (functional) database over S is a function d that associates:

- each node N of S with a finite subset $d(N)$ of $dom(N)$, and
- each arrow $f : X \rightarrow Y$ of S with a total function $d(f) : d(X) \rightarrow d(Y)$.

In order to simplify notation, we shall omit the symbol d and we shall use the expression “function $f : X \rightarrow Y$ ” to mean “function $d(f) : d(X) \rightarrow d(Y)$ ”.

Suppose now that we want to analyze document usage, say by finding the total number of hits by topic, that is by evaluating the answer to the following query against the current database:

$$Q_1: \text{”total number of hits by topic.”}$$

In our approach, to answer this query we proceed as follows:

Grouping: we invert the function t , thus grouping the URIs by topic;

Measuring: in each group, we apply the function h to each URI of the group to find the corresponding number of hits;

Aggregation: in each group, we sum up the results of measuring to have the total number of hits for that group.

The final result is shown in Figure 1(c), and it is a function from *Topic* to a new attribute that we call *TotalHits*. In other words the answer to Q_1 is the following function:

$$Ans_{Q_1} : Topic \rightarrow TotalHits. \tag{1}$$

This pattern of grouping a set of objects by inverting a function defined on them, then measuring a property in each group by applying a second function also defined on them, and finally aggregating the measures in each group by applying an operation on the measures constitutes the basic pattern of our approach.

Therefore Q_1 can be specified as a triple:

$$Q_1 = \langle t, h, sum \rangle. \tag{2}$$

Notice however that t and h have the origin of the schema as their common domain of definition, and that this condition is indispensable in order to compute the answer. Moreover, notice that the operation “*sum*” is an operation which is possible to apply over the range of h (i.e. over the integers), and that this condition is also indispensable in order to compute the answer.

In view of the previous discussion, we define an analytic query over a schema to be a triple

$$Q = \langle c, m, op \rangle \tag{3}$$

where c and m are edges of the schema having the origin as their common domain of definition, and op is an operation which is possible to apply over the range of m . We refer to the function c as the *classifier* or the *grouping function* of Q and to the function m as the *measure*.

Now, the classifier and the measure of a query can be composite functions, derived from functions in the schema by applying the operations of what we call functional algebra 11. These operations are quite elementary: composition, pairing, restriction and projection. For example, in the schema of Figure 2, one can ask the following queries:

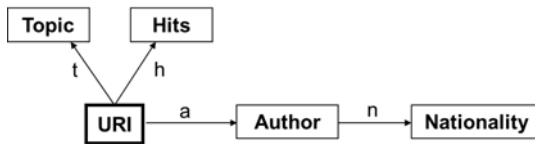


Fig. 2. Adding Author and Nationality

- Q_2 : “total number of hits by topic-author pair”, and
- Q_3 : “total number of hits by author’s nationality”.

During the evaluation of Q_2 , the grouping of URIs will be done according to a function derived from t and a , which associates each URI with a topic-author pair. This function is called the *pairing* of t and a , it is denoted by $t \wedge a$, and it is defined as follows:

$$t \wedge a : URI \rightarrow Topic \times Author, \text{ with } (t \wedge a)(x) = \langle t(x), a(x) \rangle. \quad (4)$$

The pairing $t \wedge a$ will group together all URIs having the same topic and the same author, and the answer will associate each topic-author pair with a total number of hits. Therefore, Q_2 is specified as

$$Q_2 = \langle t \wedge a, h, sum \rangle, \quad (5)$$

its answer being a function, associating each topic-author pair (x, y) to the total number of hits for that pair:

$$Ans_{Q_2} : Topic \times Author \rightarrow TotalHits. \quad (6)$$

As for the Q_3 , during its evaluation, the grouping of URIs will be done according to a function derived from a and n using functional *composition*. This composition, denoted $n \circ a$, associates each URI with the corresponding author’s nationality. Therefore Q_3 is specified as follows:

$$Q_3 = \langle n \circ a, h, sum \rangle. \quad (7)$$

During the evaluation of Q_3 , the function $n \circ a$ will group together all URIs having the same author nationality, and the answer will associate each author nationality with a total number of hits:

$$Ans_{Q_3} : Nationality \rightarrow TotalHits, \quad (8)$$

with for each nationality x , $Ans_{Q_3}(x)$ is the corresponding total number of hits.

Notice that, in all the above examples of queries (Q_1 , Q_2 and Q_3), we can also restrict the first two functions to some desirable subset of their (common) domain of definition, to form new analytic queries.

3 Analysis of Tabular Data: A Real Life Example

In this section we restrict our attention to a table T whose functional dependencies F are in Boyce-Codd Normal Form (BCNF) and we apply our approach to data stored in such a table. Tables in Boyce-Codd Normal Form are also called “normalized” and the databases of most practical applications contain only normalized tables. We recall the following facts from database theory [11]:

1. T is in BCNF if the left-hand side of every nontrivial dependency contains a key
2. Two sets of attributes X and Y are equivalent if they functionally determine each other; in particular, all keys in a table are equivalent.

3. If T is in Boyce-Codd Normal Form then its set of functional dependencies is equivalent to the set $\{K \rightarrow A, \text{ where } A \notin K\}$, where K is a key; in what follows, we assume the set F of functional dependencies of T to be of this form.

Based on the last fact above, we define a functional schema over T as follows:

1. select a key K of T
2. define an edge $K \rightarrow A$, for each attribute A of T not in K
3. for each attribute A of T not in K , define zero, one or more edges of the form $A \rightarrow I$, where I is an attribute not in T , functionally dependent on A (each such I is also called an *indicator*)

We stress here that the choice of a key is very important as each key corresponds to a different, but equivalent representation of the objects stored in the table.

Clearly, a schema defined as above is a functional schema in the sense of the previous section, therefore we can apply to it our OLAP query language, presented in the previous section. (The schema can be eventually implemented as a star schema in a relational database [1]).

The procedure of the functional schema definition, described above, is applicable in various contexts. The unique requirement is to dispose of a set of functional dependencies, satisfied by a specific application. Most of the times such dependencies can be extracted from the underlying data by the well-known methods. Such approach allows to model data, belonging to different domains, in a natural way. This is due essentially to the rather “object-oriented” approach of the functional model.

In the previous section we have seen the application of our approach to the usage analysis in digital libraries. As another example of application, consider the analysis of log data collected from users interactions with an information system, such as the e-learning environment of the KP-LAB Project [5]. KP-LAB focuses on creating an e-learning system, aimed at facilitating innovative practices of sharing, creating and working with knowledge in education and workplaces. Among other activities, KP-LAB project analyses knowledge practices of groups, in order to improve their way of working. For this purpose, teams of students, teachers, and knowledge workers are provided with tools and methods enabling them to reflect on their knowledge practices while being engaged in substantial project work over long periods of time.

To support collaborative analysis and reflection, participants need to have material evidence regarding the pursuit of knowledge practices. The exploitation of historical logging data provides a great deal of information about group activities [6]. Thus, in KP-LAB e-learning environment, each group of users has a private working space in which various “events” created by users take place. All actions, performed by users, are recorded and their attribute values are collected into a table T . The following data is stored in T :

Attribute set of T : $\{EventId, TimeStamp, UserId, ObjectId, Action, SpaceId\}$

Here, *EventId* is the only key, and analysis tasks are specified using the non-key attributes, as well as various attribute-dependent indicators, that are added to the schema for analytic purposes. The resulting functional schema, obtained by the procedure, presented previously, is shown in Figure 3. The analyst can now formulate queries, such as “Give me the top ten active users of the month” or “Give me the average number of records per *ObjectId*” etc.

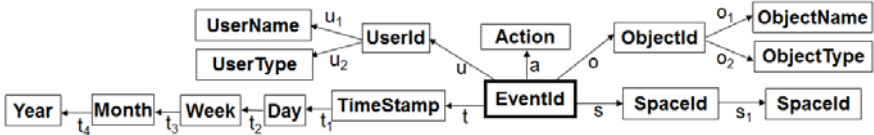


Fig. 3. KP-LAB users actions log functional schema

4 Result Visualization and Exploration

Interactive visualization is a powerful technique to slice a data set from various viewpoints and to find interesting trends from visual representations of query results. To perform interactive OLAP visualization, we need to construct many OLAP queries and dynamically map their results to appropriate visual representations. For non-professional end users these tasks are *not* easy.

In our prototype, we follow a *template-based* approach in the user interface to improve user’s experience with interactive analysis tasks. By interacting with visual components, called *visualization templates*, the user can easily define both queries and visual representations of results, simultaneously.

Moreover, our prototype, VISUAL ANALYSER, also supports what we call *result exploration* that is the possibility of visualizing the query result at aggregation levels different than those specified in the query; or the possibility of creating a *parametric representation* of the result for better visualization.

In contrast to application specific visualization system, general purpose visualization tools or frameworks allowing to dynamically formulate ad-hoc queries and to flexibly visualize query results are usually based on a generic data models, for instance [13,14,12,15] (the relational model) [15] (the multidimensional model). In particular, [15] proposes a grid layout of many charts that is suitable for multidimensional data visualization and a procedure for constructing a set of small SQL queries from a visualization specification. These are studies showing mechanisms of “how” to map data to a visual representation. However, one of our challenges was to explain “why” we can map data to a visual representation. We started from a very primitive concept, functional dependency, and disclosed a correspondence between functional dependencies and structures of visual representations. Our approach successfully explains a reason “why” we need to map functions in a source data to specific parts in a visual representation. For more details, see [9].

As it will be evident in the sequel, we exploit the fact that the result of an OLAP query in our model is a function. Moreover, when the domain of such a

function is a Cartesian product, then we can exploit the possibly many equivalent forms that the result can take using the well known Curry transformation.

4.1 Visualization Template

Broadly speaking, a visualization template is an interactive component for defining a visual representation of a designated *function*. Figure 4 illustrates how a visualization template T_{bar} defines a bar chart representation of a function $f_{bar} : xcoord \rightarrow length$. The function f_{bar} is a variable and we can bind it to a designated function in a data schema or to the answer of a query (which is also a function).

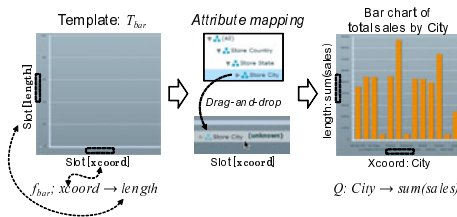


Fig. 4. The concept of visual template

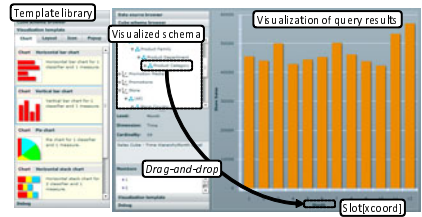


Fig. 5. The prototype user interface

To bind f_{bar} , our prototype requires mappings from attributes in the data schema to the domain $xcoord$ and the range $length$ of f_{bar} . For this purpose, T_{bar} has the visible slots $[xcoord]$ and $[length]$ on its surface to map attributes to $xcoord$ and $length$. We can specify attribute mappings just by dragging attributes from the visualized data schema and dropping them into slots. When T_{bar} accepts attribute mappings $\{A \mapsto xcoord, B \mapsto length\}$, where A and B are attributes of the schema S , T_{bar} automatically binds f_{bar} to a function retrieved by the following rule.

- If a schema function (or derived function) $f : A \rightarrow B$ exists in S , T_{bar} binds f_{bar} to f . (If more than one function of the form $A \rightarrow B$ exist in S , the system shows a dialog for choosing one of them.)
- If no function of the form $A \rightarrow B$ exists in S , T_{bar} binds f_{bar} to a query $Q\langle p_A, p_B, op \rangle : A \rightarrow op(B)$ such that O is the root of S , $p_A : O \rightarrow A$ and $p_B : O \rightarrow B$ are path expressions over S , and op is an operation preset by users.

Current implementation of our prototype (which is available to the users), uses a default bar chart template, as required by KP-LAB project members for more simplicity. However, the next version provides a visualization template library containing a set of basic chart representations (bar chart, pie chart, etc.) and layouts of multiple charts (e.g., grid layout). We can interactively define both, queries and query result representations, by just choosing templates from the library and designing attribute mappings through drag-and-drop manipulation.

4.2 Changing the Aggregation Level of the Result

The user interaction and the automating mechanism described above allow us to quickly change aggregation levels of the result. Consider for example the query “Q: Total sales by Month” over some catering company database. Figure 5 shows the result of this query in the form of a bar chart. The user can explore this result further by asking the system to produce a visualization at levels different than those specified in the query, for example, by Date, by Product, or by Category, or by City, and so on, just by dragging an attribute for classifier and dropping it into the slot associated with the X axis of the bar chart.

4.3 Parametric Visualization

When visualizing a result involving two or more aggregation levels, it is often convenient to make a series of small plots one for every category item. For example, consider a query whose result involves two dimensions, product category and city. We can make a series of plots showing totals by cities for each product category (as shown in Figure 6).

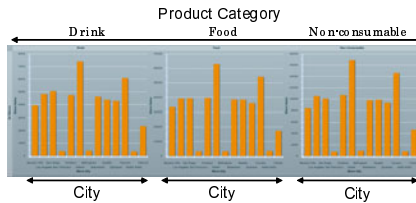


Fig. 6. The example of parametric visualization

These series of plots might convey easily information difficult to grasp from a single plot showing totals by product category and city. The underlying idea here is that, as the answer to Q has the signature $Category \times City \rightarrow Totals$, we can produce an equivalent representation of the answer using the well known “curry operation”: $(Category \times City \rightarrow Totals) \equiv (Category \rightarrow (City \rightarrow Totals))$, suggesting that we can use Category as a “parameter” for producing as many small plots $City \rightarrow Totals$, as there are product categories. Alternatively, one can use City as a parameter for producing as many small plots $Category \rightarrow Totals$ as there are cities. This kind of exploration is basically a repetition of a plot across a grid, where each plot has one variable which changes. In other words, it is a grid of multiple smaller plots driven around in a loop executing it once for every category item. Clearly, parametric visualization is not bound to a specific visualization mode: every basic visualization mode can be parameterized this way, whether it is scatter bar charts, heat maps, line charts, whichever.

Practically, parametric visualization is performed by a combination of a chart template and a grid layout template. Figure 7 shows an outline of grid layout template T_{grid} . T_{grid} defines a grid layout of a visual representation (called *cell*) to represent the function $f_{grid} : row \times column \rightarrow cell$.

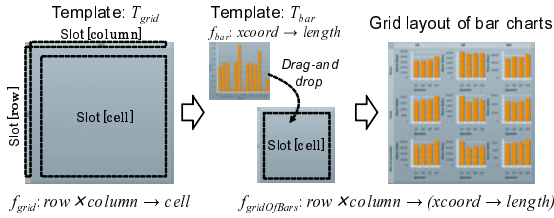


Fig. 7. The concept of parametric visualization

The interesting feature of T_{grid} is that we can map a function instead of an attribute to the range $cell$ of f_{grid} . T_{grid} has a special slot [cell] that accepts any kind of visualization template. By dropping a template into a slot [cell], we can define a grid layout of the dropped visual representation, and map the function belonging to the dropped template to $cell$. For instance, if we drop T_{bar} with function $f_{bar} : xcoord \rightarrow length$ into [cell], T_{grid} produces a grid layout of bar charts to represent function $f_{gridOfBars} : row \times column \rightarrow (xcoord \rightarrow length)$.

Parametric visualization provides yet another, important dimension of result exploration as the eye can grasp more detailed information when visualizing a set of simple plots, in addition to the information obtained from a single, higher dimensional (thus more complex) plot.

4.4 Visual Analyser

Our approach to schema design, query formulation, result visualization and exploration is implemented in an interactive flexible OLAP tool, named VISUAL ANALYSER, for the project KP-LAB [5].

VISUAL ANALYSER allows users to analyze participation and activities within past or ongoing knowledge creation processes, by visually representing them based on information stored in the produced logs. More precisely, it visualizes frequencies of users activities and provides detailed information on the nature of the activities performed on particular knowledge objects. These visualizations stimulate teachers and students to reflect on the distribution and types of their activities with respect to time, type of object or subject etc.

We analyzed the KP-LAB users activities logs, and derived an analysis schema, as described in section 3, and the visualization schema based on the high-level formal model, presented in 9.

This approach allowed us to clearly understand the relationship between data and visual representations, and to naturally derive mechanisms for realizing many features of the VISUAL ANALYSER, including visualization design by drag-and-drop, automatic query formulation and interactive data filter construction.

Through repetitions of formal analysis and real user tests, we have succeeded in polishing the VISUAL ANALYSER system as a simple but powerful environment that provides smooth log analysis experience.

The current version of VISUAL ANALYSER can be accessed and tested at [7].

5 Concluding Remarks

We have seen a functional model for data analysis and an interactive interface (based on that model) that supports users in performing data analysis and visualization of results. The prominent features of our model are that it is simple to grasp and easily amenable to visual interaction. We are currently investigating the use of our tool in the context of new application environments, in particular, usage analysis in digital libraries.

Acknowledgements. This work is partially supported by the following projects:

- European project ASSETS: Advanced Search Services and Enhanced Technological Solutions for the European Digital Library (CIP-ICT PSP-2009-3, Grant Agreement no 250527)
- French CNRS project NOVA: A Novel Data Model and Query Language for Digital Libraries (PICS 5220)

References

1. Spyratos, N.: A functional model for data analysis. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 51–64. Springer, Heidelberg (2006)
2. Mondrian, <http://mondrian.pentaho.org/>
3. IBM Cognos 8 BI Analysis, <http://www.ibm.com/software/data/cognos/products/cognos-8-business-intelligence/analysis.html>
4. Oracle BI Discoverer, <http://www.oracle.com/technology/products/discoverer/>
5. KP-Lab: Knowledge-Practices Laboratory, <http://kp-lab.org/>
6. Paralič, J., Babič, F., Wagner, J., Simonenko, E., Spyratos, N., Sugibuchi, T.: Analyses of knowledge creation processes based on different types of monitored data. In: Rauch, J., Raš, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 321–330. Springer, Heidelberg (2009)
7. Visual Analyser, <http://mielikki.mobile.metropolia.fi:8080/VALSServer/clientV3/VALSClientV3.html>
8. Mansmann, S., Neumuth, T., Scholl, M.H.: OLAP technology for business process intelligence: Challenges and solutions. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 111–122. Springer, Heidelberg (2007)
9. Sugibuchi, T., Spyratos, N., Simonenko, E.: A framework to analyze information visualization based on the functional data model. In: 13th International Conference on Information Visualisation, pp. 18–24. IEEE Computer Society, Los Alamitos (2009)
10. Rizzi, S., Abelló, A., Lechtenböcker, J., Trujillo, J.: Research in data warehouse modeling and design: dead or alive? In: Proceedings of OLAP 2006, ACM 9th International Workshop on Data Warehousing and OLAP, pp. 3–10. ACM, New York (2006)
11. Codd, E.F.: Recent investigations in relational data base systems. In: IFIP Congress, pp. 1017–1021 (1974)

12. Spotfire Web site, <http://spotfire.tibco.com/>
13. Derthick, M., Kolojechick, J., Roth, S.F.: An interactive visualization environment for data exploration. In: Proc. Of Knowledge Discovery In Databases, pp. 2–9 (1997) (press)
14. Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Law, S., Myllymaki, J., Wenger, K.: Devise: Integrated querying and visual exploration of large datasets. In: Proceedings of ACM SIGMOD, pp. 301–312 (1997)
15. Stolte, C., Tang, D., Hanrahan, P.: Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Vis. Comput. Graph.* 8, 52–65 (2002)

An Analysis of an Efficient Data Structure for Evaluating Flexible Constraints on XML Documents

Stefania Marrara¹, Emanuele Panzeri², and Gabriella Pasi²

¹ Università degli Studi di Milano - DTI
via Bramante 65, 26013 Crema (CR), Italy
stefania.marrara@unimib.it

² Università degli studi di Milano Bicocca
viale Sarca 336, 20126 Milano (MI), Italy
{panzeri,pasi}@disco.unimib.it

Abstract. In this work we describe and evaluate an efficient data structure developed to deal with an extension of the XPath language that allows the specification of flexible constraints on both the textual content and the document structure of XML documents. Our approach is based on an inverted file representation of XML documents where both structure and content are taken into account. The proposed approach is described, as well as its implementation, by particularly addressing the problem of offering an efficient evaluation of structure-based flexible constraints.

Keywords: XML flexible querying, XML data structure, XML Retrieval.

1 Introduction

Since its first introduction in 1998, XML constitutes the basis for data interchange on the Internet. Differently from the most popular HyperText Markup Language (HTML), XML allows document designers to set their own tag vocabulary, and to describe the structure of documents. It is widely recognized that, in order to fully exploit the information carried by an XML document, a query language aimed at retrieving XML documents should allow the formulation of constraints on both the document content and structure. In other words, it should allow to specify constraints on both tag nesting as well as their content. The birth of huge collections of XML documents has implied the definition of specific query languages to inquire them; the main representatives are today the W3C standards XPath [25] and XQuery [26]. These XML query languages assume that the user is aware of the target document structure, and they only allow an exact specification of the target documents, due to the Boolean nature of their query-document matching systems. This assumption is often unrealistic, since most XML documents have no pre-set structure (DTD or XML Schema). The XPath and XQuery languages follow a database style query formulation for

fragment selection: the constraints are evaluated in a crisp way without computing any a retrieval.

Recent research in both Information Retrieval and Database communities has led to several approaches aimed at introducing some degrees of flexibility in XML retrieval [11], [13], [16], [17], [23] and to allow flexible fragment retrieval producing in a few cases ranked XML fragments. In particular in [15] and [9] an XPath extension has been proposed that allows the expression of flexible constraints on both content and structure: users can both express keywords for content-based search, and flexible constraints on the document structure, thus allowing to formulate queries that retrieve XML fragments having a similar structure to the approximate one specified in the query. The query evaluation produces a ranked list of XML fragments, according to the Information Retrieval approach to search.

An important issue raised by all the approaches proposed in the literature for defining flexible queries on XML documents is the problem of defining data structures that efficiently allow the evaluation of flexible constraints on both content and structure of XML documents. In fact the amount of data that have to be handled, in comparison with a traditional Information Retrieval System, is bigger in an XML retrieval system due to the fact that each XML fragment could be considered as a single unit of retrieval with the consequence of a considerable increase of the data (fragment position, term frequency, ..) that have to be stored for each fragment in each XML document in the indexed document collection. In this paper we address the problem of defining a data structure which allow an efficient evaluation of flexible constraints, focusing in particular on structure-based flexible constraints. The proposed approach defines an inverted file based representation of XML documents that allows the evaluation of both keyword-based and structure-based flexible constraints. This structure constitutes a modified and enhanced version of the solution proposed in [8], by introducing some key requirements of XML retrieval outlined in the IR literature, that were not considered in the original proposal. The data structure has been tested w.r.t. the evaluation of two novel structure-based flexible constraints proposed in [9], namely *below* and *near*.

The paper is organized as follows. In the next section a review of the most important and recent contributions related to XML retrieval and flexible querying is presented. In Section 3, a brief description of the flexible query language that has inspired the data structure described in this paper is presented. Section 4 will explain the proposed data structure, its implementation for efficiently supporting the flexible constraints, as well as a preliminary efficiency evaluation. Finally, Section 5 summarizes our conclusions and future works.

2 Related Works

In the Information Retrieval research context, the approaches to inquiry XML documents are classified as *Content-Only* search (CO), and both *Content And Structure* search (CAS). *CO* approaches address the issue of querying XML documents by using a keyword based approach ([3], [10] and [12]) without any

possibility to specify constraints on the expected document structure. Most of the existing keyword based search systems return query results based on the notion of *Lowest Common Ancestor (LCA)* [18], and its variants ([13] and [19]).

On the opposite side, CAS approaches focus on approximate matching of limited XPath predicates (usually the *child* axis is transformed into a *descendant* axis during the query evaluation). These approaches are based on the notion of *structural hint* which considers the query structure as a mere template of the required information. All fragments similar to the template are retrieved. Examples of CAS approaches are offered by, for instance, XIRQL [17], NEXI [24], TeXQuery [5], and FleXPath [7], and the recent standard XQuery and XPath Full Text [2] languages.¹ The above approaches propose a matching of a limited set of XPath structural constraints (usually axes), and the associated indexing structures are designed on the basis of a specific set of pre-selected queries. The pre-selected queries individuate the target fragments (i.e. the retrieval units) to be indexed in order to make the system able to retrieve them with an associated score. A special case of CAS approach is offered by the language NEXI, Narrowed Extended XPath I [24], which supports only the *descendant* and *self* steps, extended by the special *about* function which allows a flexible evaluation of the target node content in the IR style.

Other approaches proposed in the literature define some flexibility on the evaluation of the query structure in a more explicit way. For instance, in [17], [21], and [6] the authors define some relaxations such as the introduction of generalized data types, the adoption of edit distances on paths, and some operations to modify the structure like *delete* a node, *insert* intermediate nodes or *rename* a node. FleXPath [7] is the first approach proposing a formalization of relaxations on the evaluation of the structure of XML queries, as well as the first algebraic framework for spanning relaxations. In addition, it proposes new ranking functions with properties that relaxations must satisfy, and it develops efficient evaluation algorithms.

All the above mentioned approaches, however, do not allow users to define the extent and the type of the desired flexible constraints in the query, as the query itself is considered as a mere template of the required information. All fragments similar to the template are retrieved. In these approaches the user has no way to distinguish between portions of the query that must be considered as exact and those that allow a certain flexibility in the retrieval process. In order to allow users to explicitly specify both content-based and structure-based flexible constraints in a query aimed at retrieving XML fragments, in [15] and [9] a flexible language has been proposed as an extension of the XPath query language. In the proposed extension, a flexible constraint is expressed as an approximate requirement on the desired document content and/or structure, and its evaluation produces a score, expressing the degree of compatibility of the document's content/structure with respect to the user requirement. In Section 3 this language is shortly described.

¹ Differently from CAS approaches, XPathFT and XQuery FT do not include structural hints but query structures are evaluated as in the traditional standards.

3 The Flexible Query Language

In [9], [15] and [14] a new approach aimed at introducing flexibility in XPath has been defined by means of the formalization of flexible constraints conceived to extend the syntax of XPath on both the content and the structure of a XML document. Two types of constraints have been introduced: flexible constraints on node contents (the defined constraints are named *cw* and *around*), and flexible constraints on the document structure to specify flexible selection conditions on the tree structure of the XML document (the defined constraints are named *near*, *below*, and *approximately*).

The constraint *cw* (*contain words*) is applied to nodes that have a textual content, and allows to specify keyword-based constraints, as in usual IR query languages. By the proposed syntax, the keyword *cw* is followed by the list of the implicitly and-ed search terms to be retrieved in the specified textual node content: “//book/title[*cw*(., “music”)]”.

The constraint *around* is applied to numeric or date values (within specific numeric or data content nodes): “//book/price[*around*(., 90)]”. The approximate evaluation of the *around* constraint produces a numeric score expressing the compatibility between the constraint and the value of the considered XML element.

The structure-based constraint *below*, inserted as a flexible axis of a path expression, allows to extract XML fragments that are direct descendants of the current node like the “//” XPath *descendant* selector: “book/author*BELOW*name”. For each retrieved fragment a score is computed that is inversely proportional to the distance between the *ideal* path structure (the one in which the target node is direct child of the starting node) and the retrieved path structure.

The constraint *near*, inserted as a flexible axis of a path expression, is defined to the aim of selecting XML fragments that are connected through any path to the current node. Also in this case, for each retrieved fragment a score is computed, which is inversely proportional to the distance between the expected path structure and the retrieved one.

The constraint *approximately* allows to select the elements with a given name that have a number of direct descendants close to the one indicated in the query.

As the focus of this paper is neither to introduce the syntax nor the semantics of the flexible query language, but to analyze data structures that allow an efficient and effective evaluation of both keyword-based constraints and flexible structured-based constraints, in this paper we do not give more details about the flexible query language shortly sketched in this section. In fact, the aim of the paper is more general, and the proposed data structures could be useful to implement any other flexible constraints on both content and structure of XML documents. More details on the flexible query language shortly sketched in this section can be found in [9], [15] and [14].

4 An Enhanced Inverted File Structure

The implementation of the flexible query language is based on an inverted file approach. In this section the inverted file based approach to evaluate content

and structure-based constraints of XML documents is described. We adopted, as a starting point, the data structure presented in [8] which is divided into two main parts: *content* and *paths*. We made this choice since the structure presented in [8] offers a new logical XML node identification scheme that encodes complete rooted data paths. The encoding is context-independent, space efficient, and fast to decode. In fact no decoding is required to determine parent-child and ancestor-descendant relationships between node IDs. However, in order to evaluate flexible constraints (like the ones presented in Section 3), the data structure defined in [8] is neither fully efficient nor complete from an Information Retrieval point of view, since it does not address some key requirements of XML retrieval outlined in the IR literature, such as how to assign the correct tf-idf weight to a term in a XML fragment, or how to cope with multiple heterogeneous collections in an efficient way. In fact, following the tradition of XML databases indexing and retrieval, the approach presented in [8] is designed to deal with a single XML “document” that summarizes in a single tree the entire document collection.

The choice of this data structure as a basis of the indexing structure presented in this section has two important motivations: 1) the encoding model adopted in [8] is able to efficiently evaluate the structural joins necessary to compute any twig query; 2) as shown in [8] (and reviewed here in the following), the encoding proposed is efficient in dealing with the parent-child and descendant-child relationship. Hence the query evaluation process is easier and does not require any particular decoding to deal with path and tree structures.

4.1 The Basic Data Structure

The indexing structure proposed in [8] is based on the notion of *Minimal Path Identifier* (minPID), which is a way to encode XML paths based on the *eXtended DataGuides* (XDG), a specification of the XML structure looser than both DTDs and XML Schemas. XDGs are directly derived from the document collection and enumerates all rooted label paths in a data source by assigning to each of them a unique identifier called *XDG node#*. In addition, an XDG assumes that the maximum number of instances under a single parent node in the source is known (sibling *fanout*). Note that, given a data source, it is always possible to construct an XDG (the only requirement is that all XML documents are well formed), and the *fanout* can be directly computed from the source.

Fig. 1 shows an example of a fragment of an XML document with the corresponding XDG fragment graph. In the XDG graph each possible path is represented only once, and for each node in the path the maximum *fanout* of the node in the collection is shown between parentheses (for example the node *bk* has a maximum number of 583 siblings in the entire XML tree; the example fragment in Fig. 1a shows only four of them). The number on the node itself represents its *XDG node#* assigned on the basis of a pre-left order. Based on the notion of XDG, a *Minimal Path Identifier* (minPID) of a certain data path in a source document can be defined as a pair (p, s) , where p is a rooted label path in the document XDG, and s is a sequence of sibling positions for nodes p_i excluding nodes that do not have siblings (for example the *books* node in Fig. 1).

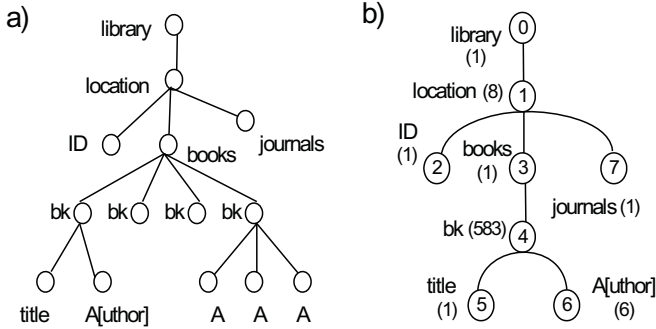


Fig. 1. A sample XML fragment (a) and the corresponding XDG (b)

All elements are encoded as integers and ordered by their distance from the root node. Relationships between two nodes a and b in a document can be easily determined based on their minPIDs [8]. For example, we can consider a document path $d=Library/Loc[5]/books/bk[4]/author[3]$, which is an instance of the rooted label path $p=Library/Loc/books/bk/author$. The number in square brackets identifies the node position among sibling nodes. Following the Fig. 1, the minPID for the example document path is $(\langle 0, 1, 3, 4, 6 \rangle, \langle 5, 4, 3 \rangle)$.

For storage and indexing purposes an efficient minPID encoding is proposed: instead of using numeric sequences to represent the node position, a *position number* is computed and represented using a binary efficient encoding system, this resulting path identifier is called *Micro-Path ID* (μ PID). For its definition see [8].

In Fig. 2 an overview of the data structure proposed in [8] is presented. Based on the previous notions, the index structure proposed in [8] is composed by three indexes: the term index (T-index), the path index (P-index) and the physical

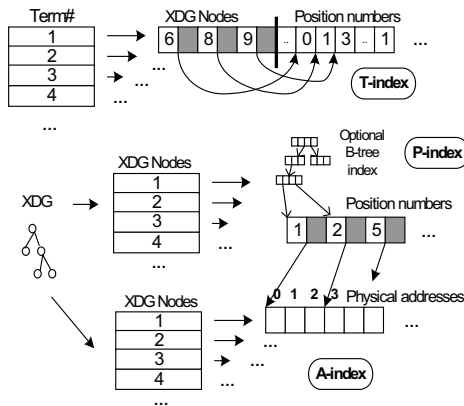


Fig. 2. Overview of the data structure proposed in [8]

address index (A-index). The T-index is created to process keyword based queries, and it is a classical inverted file composed by a term dictionary, and a posting file where each term is associated with the XML fragments containing it expressed by their minimal path identifier.

In Fig. 2, terms are represented as integers. Each term points to the leaf nodes that contain it, each node is represented by its absolute position in the document tree given by the *XDG node#*, and the *position number*. The P-index encodes the remaining nodes of the tree: again it is composed by an inverted file with a dictionary of the *XDG node#*, and a posting file encoding the nodes positions in the tree. The position numbers of the non-leaf nodes can be optionally stored by using an efficient data structure as a B-tree. Finally, the address index is the data structure necessary to have a direct retrieval of source fragments. This structure is used to speed up the execution of simple path queries.

4.2 The Extended Data Structure

In this section we define the proposed extension of the data structure sketched in the previous section: it makes possible to efficiently process flexible constraints like the ones introduced in Section 3.

In Fig. 3 an overview of the two-layered indexing structure explained in this section is given. The first inverted file shown in the figure is a classical term inverted file that encodes the necessary information for the execution of the content-based constraints (such as *cw* and *around*), which can be specified to activate a keyword-based search in an IR style. The second file is the path indexing file that encodes entire XML rooted paths, and contains a link to their physical positions.

By the proposed extension of the data structure described in Section 4.1, the XML paths are encoded using a modified version of the minimal path identifiers. In fact a key difference with respect to the approach in [8] is that our extended structure does not represent an entire XML collection as a single rooted tree, but it manages each document individually, as shown in Fig. 3. Hence a single node instance in the document collection requires two parameters to be identified: the *docID* (which identifies the document containing the target node) and its *rooted label path*. The rooted label path can be expressed by a single natural number replacing each label in the XDG by the corresponding number in document order as previously described. Based on this consideration, the definition of minimal path identifier (minPID) is modified by inserting the notion of document, thus obtaining a triple of values: $minPID = (docID, p, s)$. Considering a document path $p = /Library/Loc[5]/books/bk[4]/author[3]$, and a document $d = doc1$, such a path can be represented as the triple $(doc1, /Library/Loc/books/bk/author, <5,4,3>)$. Moreover, by replacing the label path with *XDG node#s*, p can be concisely represented by the the minPID triple: $(doc1, <0,1,3,4,6>, <5,4,3>)$. The notion of μPID is extended in the same way.

Considering again Fig. 3, the term inverted file is composed by two structures: a term dictionary T where each term is associated with its document frequency

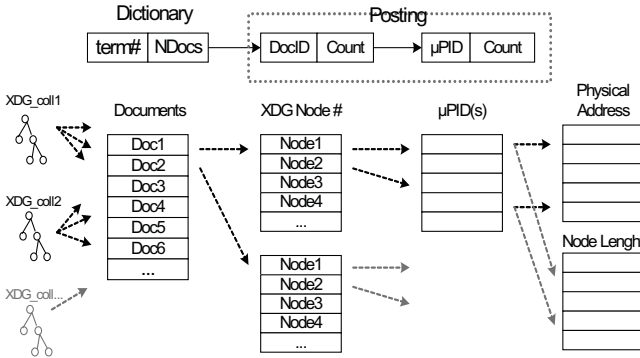


Fig. 3. The new data structure

and with a pointer to the posting file. Each couple $(t, NDOct)$ in the dictionary, where t is a term and $NDOct$ is the number of XML documents indexed by t in the entire heterogeneous collection, points to several blocks of information in the posting file, each block containing: *docId* - document identifier, *dococc* - number of occurrences of t in the document *docId*, *nodeId* - node identifier given by the couple $(XDGnode\#, positionnumber)$, *occ* - number of occurrences of t in the node *nodeId*. By our model an XML document content is formally represented by the set of its index terms, and the document leaf nodes that contain them. In particular we consider XML leaf node contents divided into two classes: *data nodes*, containing dates, numbers etc., and *text nodes*, containing unstructured text descriptions, such as sections, abstracts, etc.

The association of a varying degree of significance with index terms in documents has been widely studied in the field of Information Retrieval. Following [20], where each document element is considered as a retrieval unit, the value of the index term weight $F(nodeId, t)$ of a term in a node content is defined so as to increase with the frequency of term t in the node *nodeId*, and to decrease with the increasing frequency of the term in the document collection. Some possible formulas that can be used to compute the index term weight of a term in a leaf node are also those implemented in the Lemur toolkit [4] in the context of passage retrieval. Other choices are possible and our indexing structure is designed in order to store the parameters needed to compute all the weighting functions proposed in the literature.

The second layer of our indexing structure, the path indexing file, allows the evaluation of the structural constraints specified in queries. Instead of grouping *nodeIds* by their *XDG node#* only as it happens in [8], we rely on a doubly sorted index because we group $\mu PIDs$ by their *docId* and then we sort each group by the *XDG node#s* inside the same document in such a way that contiguous index blocks contain *XDG node#s* having distance one from each other (see Fig. 3). With this mechanism we improve the execution of tree-path queries, while pattern queries are still executed efficiently. This is because query patterns can

be evaluated directly in the term inverted file where identified *nodeIds* directly point to physical addresses. The structure of the path indexing file is shown in Fig. 3. Each document identifier points to the *XDG node#* that contains the node identifiers in the target XML document. Each *XDG node#* points to the *position numbers* that individuate each path, with the same label path, in the target document; then each path points to its physical address. Note that, depending on the type of XML internal representation, we may not perform physical sorting on the *XDG node#*. Alternatively we can use entries to an additional node index such as the one provided by native XML databases [22]. We remark that the additional redirection due to the XML database index will perform better than using pointers to physical file system blocks whenever the underlying storage provides mechanisms that point the *XDG node#s* with distance one from the target *XDG node#*.

The proposed structure allows a two-step evaluation of queries related to structure and content constraints respectively. For the complex and computational expensive flexible structure matching the evaluation is performed only on the subset of XML document fragments that match the content-based constraint part. The content matching is performed by using the efficient inverted-file index data structure that allows to retrieve the list of nodes that match the terms specified in the content-based constraint. This technique reduces the amount of structure-constraint checks that have to be done to evaluate the complete query to retrieve the desired XML node fragments. Our indexing structure allows, given a query, to efficiently locate documents relevant to the *cw* constraint, and to load into the memory only the XDGs necessary to the structure-based constraint evaluation.

4.3 Preliminary Evaluation

In this section we present some preliminary experiments² we performed in order to test the inverted file structure during the evaluation of flexible queries. We used two collections extracted from the INEX [1] data-centric collection 2010 (which is built using the IMDB movie database collection); the first one (C_1) is composed of 1100 documents (up to 41MB) and the second one (C_2) of 12518 documents (523MB). We considered two different scenarios: in the first one we evaluated only the structure-based constraints specified by the *below* and *near* selectors. In the second one we considered also the *cw* constraint, and we performed a two-step evaluation process as described in section 4.2.

In both cases the set of queries was partitioned w.r.t. the frequency f (high or low) of the involved elements nodes in the collection. For each query we have evaluated the efficiency of the data structure in terms of the time to access the μ PIDs related to the nodes specified in the query, as well as the nodes involved in the evaluation process.

The evaluation of the *near* constraint has been executed with a maximum path distance set to 3, while in the evaluation of the *below* constraint the entire

² We used an Intel® Pentium®Core 2 Duo, 2.2 GHz system, with 2GB dual-channel DDR-667Mhz RAM, Windows7 Professional 32 bit and Java5 Virtual Machine.

Table 1. Average execution times of queries with *below* and *near*

	C_1		C_2	
	BELOW	NEAR	BELOW	NEAR
	(ms)	(ms)	(ms)	(ms)
f_{low}	392	489	1347	1324
f_{high}	332.6	332	3208	3370

document tree is checked (no depth constraints are set). The preliminary results show that the average evaluation times for both constraints are reasonable (see Table 1). In the second set of experiments we have taken into account queries involving one structure-based constraints and the constraint *cw*. In this case the query evaluation was performed using the two-step evaluation strategy presented in Section 4.2 and therefore we applied the structure-based constraints evaluation only to the set of document fragments pre-selected by the evaluation of the keyword constraint *cw*. An example of query in the experimental set is “*//person near₃ //otherwork [cw(., music)]*”.

The results shows that the evaluation of the *cw* constraint reduces the collection on which the structure-based constraints are applied to about 33% of the documents and 10% of the document paths, while the execution times are reduced to about 33% w.r.t. the execution of the same structure-based query without the pre-selection technique. More efforts need to be devoted to experiments, also with respect to the evaluation of the effectiveness of the proposed approach; this is a difficult task as it must be outlined that any testbed has been proposed before at INEX [1] for flexible constraints evaluation on XML document collections.

5 Conclusions and Future Work

In this work we have proposed and analyzed an inverted file structure to evaluate query languages that allow the specification of flexible constraints on both document content and structure. The proposed data structure allows a multi-document indexing of both document structure and content; on this data structure we have implemented the execution of the flexible constraints defined in [9]. Our proposal leverage the data structure and the node identification schema proposed in [8] to enhance both structural and content-based constraint evaluation. We have presented some preliminary experiments to show the feasibility of the proposed approach.

Future works will address an extensive evaluation of the proposed data structure as well as the language effectiveness. With respect to the language adopted for our tests we are planning to provide a formal definition of the flexible XPath extension. From a data structure point of view some improvements could be achieved with a new implementation of the indexing and data handling methods by switching from Java to the C/C++ languages. A second effort could be finalized at the implementation of an XPath query engine based on the proposed data structure.

References

1. INitiative for the Evaluation of XML Retrieval, <http://inex.is.informatik.uni-duisburg.de/>
2. XQuery and XPath Full Text 1.0, <http://www.w3.org/TR/xpath-full-text-10/>
3. Ali, M.S., Consens, M.P., Kazai, G., Lalmas, M.: Structural relevance: a common basis for the evaluation of structured document retrieval. In: CIKM 2008, pp. 1153–1162 (2008)
4. Allan, J., Callan, J., Collins-Thompson, K., Croft, B., Feng, F., Fisher, D., Lafferty, J., Larkey, L., Truong, T.N., Ogilvie, P., Si, L., Strohman, T., Turtle, H., Zhai, C.: The lemur toolkit for language modeling and information retrieval (2003)
5. Amer-Yahia, S., Botev, C., Shanmugasundaram, J.: Texquery: a full-text search extension to xquery. In: WWW 2004, pp. 583–594 (2004)
6. Amer-Yahia, S., Cho, S., Srivastava, D.: Tree pattern relaxation. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 496–513. Springer, Heidelberg (2002)
7. Amer-Yahia, S., Lakshmanan, L.V.S., Pandit, S.: Flexpath: flexible structure and full-text querying for xml. In: SIGMOD 2004, pp. 83–94 (2004)
8. Bremer, J., Gertz, M.: Integrating document and data retrieval based on xml. The VLDB Journal 15(1), 53–83 (2006)
9. Campi, A., Damiani, E., Guinea, S., Marrara, S., Pasi, G., Spoleтини, P.: A fuzzy extension of the xpath query language. J. Intell. Inf. Syst. 33(3), 285–305 (2009)
10. Carmel, D., Maarek, Y.S., Mandelbrod, M., Mass, Y., Soffer, A.: Searching xml documents via xml fragments. In: SIGIR 2003, pp. 151–158 (2003)
11. Chinenyanga, T.T., Kushmerick, N.: An expressive and efficient language for xml information retrieval. J. Am. Soc. Inf. Sci. Technol. 53(6), 438–453 (2002)
12. Clarke, C.L.A.: Controlling overlap in content-oriented xml retrieval. In: SIGIR 2005, pp. 314–321 (2005)
13. Cohen, S., Mamou, J., Kanza, Y., Sagiv, Y.: Xsearch: A semantic search engine for xml. In: VLDB, pp. 45–56 (2003)
14. Damiani, E., Marrara, S., Pasi, G.: Fuzzyxpath: Using fuzzy logic an ir features to approximately query xml documents. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 199–208. Springer, Heidelberg (2007)
15. Damiani, E., Marrara, S., Pasi, G.: A flexible extension of xpath to improve xml querying. In: SIGIR 2008, pp. 849–850 (2008)
16. Florescu, D., Kossmann, D., Manolescu, I.: Integrating keyword search into xml query processing. Comput. Netw. 33, 119–135 (2000)
17. Fuhr, N., Grobjoehann, K.: Xirql: a query language for information retrieval in xml documents. In: SIGIR 2001, pp. 172–180 (2001)
18. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: Xrank: Ranked keyword search over xml documents. In: SIGMOD, pp. 16–27 (2003)
19. Li, G., Feng, J., Wang, J., Zhou, L.: Effective keyword search for valuable leas over xml documents. In: CIKM, pp. 31–40 (2007)
20. Lu, W., Robertson, S., MacFarlane, A.: Field-weighted xml retrieval based on bm25. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 161–171. Springer, Heidelberg (2006)
21. Schlieder, T.: Similarity search in xml data using cost-based query transformations. In: WebDB, pp. 19–24 (2001)

22. Schöning, H., Wäsch, J.: Tamino - an internet database system. In: Zaniolo, C., Grust, T., Scholl, M.H., Lockemann, P.C. (eds.) EDBT 2000. LNCS, vol. 1777, p. 383. Springer, Heidelberg (2000)
23. Theobald, A., Weikum, G.: Adding relevance to xml. In: WebDB 2000, pp. 105–124 (2001)
24. Trotman, A., Sigurbjörnsson, B.: Narrowed extended xpath i (nexi). In: Fuhr, N., Lalmas, M., Malik, S., Szilávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 16–40. Springer, Heidelberg (2005)
25. W3C: Xml path language (xpath) 2.0 (November 2007), <http://www.w3.org/TR/xpath20/>
26. W3C: Xquery 1.0: An xml query language (November 2007), <http://www.w3.org/TR/xquery/>

Intuitionistic Fuzzy XML Query Matching

Mohammedsharaf Alzebd¹, Panagiotis Chountas¹, and Krassimir Atanasov²

¹ University of Westminster, ECSE Department, 115 New Cavendish Street
W1W 6UW London, UK

{alzebdm, chountp}@wmin.ac.uk

² Bulgarian Academy of Sciences, CLBME Department

Sofia, Bulgaria

krat@argo.bas.bg

Abstract. As the popularity of XML as a de facto standard for data representation and communication is rising, a need has been identified for efficient XML querying techniques that can overcome the dilemma of diversity in the structure of XML data sources. In this work, we propose our approach of using Intuitionistic Fuzzy Trees (IFTr) to achieve approximate XML query matching, making the query result include not just the XML data trees that exactly match the query, but also the ones that partially match it. Our approach has a potential to return useful query answers while pertaining good performance. Users will have the option to choose between quick and less accurate results or time costing and more accurate results.

Keywords: Intuitionistic Fuzzy Trees, Tree Similarity(Inclusion), Pattern Trees.

1 Introduction

XML is becoming the predominant standard for data representation, especially on the web. However, the heterogeneity in the structure of XML data sources has come to the surface as an obstacle in the way of sharing and exchanging XML data. This is mainly due to fact that the same data can be represented with different data schemas (DTDs) in different XML sources. In addition, the flexibility of XML e.g. having optional elements, results in more diversity in data schemas. Querying such sources using classical query languages, such as XQuery, not just requires knowing the underlying schemas, but it also retrieves only results that exactly match the query, which is inefficient in context of XML-based applications. Approximate query matching and returning ranked list of results is becoming the predominate technique when dealing with XML data sources. This means that not just the exact answer of the query, but also the “similar” answers will be retrieved.

XML (eXtensible Markup Language) is the standard format for structured documents and data on the Web. It is extensible because it is not a fixed format like HTML, which makes it possible to define new tags. Unlike HTML, XML documents consist of data and description of that data (Meta data) in a text format. While HTML was designed to display data, XML was mainly developed to structure, transport and

store data [23]. Because of being the most common tool for data transmissions between heterogeneous systems, XML has gained so much popularity recently, especially in web applications.

As the amounts of data transmitted and stored in XML are rapidly growing, the ability to efficiently query XML is becoming increasingly important. Several XML query languages have been proposed for that purpose such as XML-QL, YATL, Quilt, Lorel and XQuery. In 2007, XQuery, which is an extension of XPath, has been recommended by W3C making it the most popular language for querying XML data. XQuery is designed to allow the construction of concise, flexible and easily understood queries that can operate on diverse XML data sources, including both databases and documents [8]. In the XQuery data model, each document is represented as a tree of nodes, where there is one node called “root” and other nodes that have parent-child and ancestor-descendant relationships. To query XML documents, a number of expressions can be used, the most powerful one is called FLOWER (for-let-where-order by-return) which is similar to the (select-from-where) from the relational SQL. XQuery expressions have provided some advantages over previous XML query languages. However, there are still some performance-related and structural heterogeneity challenges that need to be addressed before the language can be mature enough.

For example, in Fig. 1 below, there are two data trees DT1 and DT2 from two different data sources. Those represent the university domain and have details about departments, staff, research groups, publications and research projects. As shown, the two data trees have different structure than of the pattern tree, which makes it impossible to retrieve data from both of them using traditional XML query languages. TP is a pattern tree for an XML query to return details of departments located in London and have research groups and projects. The query also returns the titles of any publications as well as names of related projects. The schemas here are big and very diverse; a parent/child relationship in one tree can be a sibling relationship in another. Furthermore, different vocabulary is used to describe the same data e.g. the department location is represented by the “location” node in DT1, whereas in DT2 it is called “address”. For all the above reasons, XML query languages need to be extended to resolve such challenges.

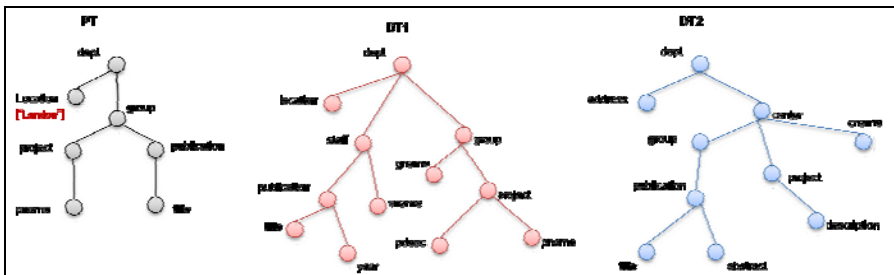


Fig. 1. A pattern tree with two different data trees

In this work, we consider the problem of approximate XML query matching from a database perspective. We believe that pattern tree relaxation approaches have some drawbacks and that an Intuitionistic Fuzzy approach is very useful to mine approximate XML schemas, semantically closer to user perception of the reality. Our approach will investigate initially tree inclusion as one solution to the problem. The main idea is to propose a definition of soft inclusion, meaning that a pattern tree is no more included or not in an XML data tree, but gradually included within it, or to what extent a tree is similar or dissimilar to another in the latter case. Considering the above standard measures of support and confidence of a Pattern Tree PT included in Data Tree DT need to be redefined. When it comes to Intuitionistic fuzzy logic inclusion, arch and node inclusion have to be considered, in case of similarity.

The rest of this paper is structured as follows. In section 2, we introduce the concept of IFTr with a set of definitions. We explain our approach of matching pattern trees against data trees using IFTr in section 3 and in section 4 we present a list of relevant studies and compare them to ours. Finally, in section 5, we conclude the paper and provide further work directions.

2 Intuitionistic Fuzzy Trees (IFTr)

As XML documents follow a tree-structured model, it has become popular to express queries against XML data sources as Pattern Trees (PTs). When a query is executed against a number of XML data trees (DTs) within a forest, the associated pattern tree will be compared to all subtrees within that forest and whenever there is a match (called Witness Trees), the data tree(s) will be retrieved. Two conditions need to be satisfied for successful data retrieval: (i) Matching the structure (Schema) (ii) Matching the predicate (Condition).

A PT is compared against XML subtree using the IFTr approach which consists of comparing the nodes and arcs of the PT against each subtree, and then calculating the similarity based on the number of matching nodes and arcs. Before presenting our proposal, we present a number of definitions.

Intuitionistic Fuzzy Graphs. (IFG) was first introduced by Shannon and Atanasov in 1994 [2]. As a Tree is a special case of a Graph, [3-6] the concept of IFTr defined as an extension of the IFG. Below we introduce a number of definitions to illustrate the IFTr properties as by [7, 22].

Definition 1

Let a set E be fixed. An IFS (Intuitionistic Fuzzy Set) A in E is an object of the following form:

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle \mid x \in E \}$$

Where functions $\mu_A: E \rightarrow [0, 1]$ and $\nu_A : E \rightarrow [0, 1]$ determine the degree of membership and the degree of non-membership of the element $x \in E$, respectively, and for every $x \in E$:

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1$$

Definition 2

Let the oriented graph $G = (V,A)$ be given, where V is a set of vertices and A is a set of arcs. Every graph arc connects one or two graph vertices.

$$A^* = \{ \langle \langle v, w \rangle, \mu_A(v, w), \nu_A(v, w) \rangle \mid \langle v, w \rangle \in V \times V \}$$

The set A^* is called an IFG if the functions $\mu_A: V \times V \rightarrow [0, 1]$ and $\nu_A : V \times V \rightarrow [0, 1]$ define the respective degrees of membership and non-membership of the element $\langle v, w \rangle \in V \times V$ and for all $\langle v, w \rangle \in V \times V$:

$$0 \leq \mu_A(v, w) + \nu_A(v, w) \leq 1$$

Definition 3

Let $G = (V, A)$ be a given IFTr. We can construct its standard incidence matrix. After this, we can change the elements of the matrix with their degrees of membership and non-membership. Finally, numbering the rows and columns of the matrix with the identifiers of the IFTr vertices, we will obtain an IM. For example, the constructed IM for the IFTr in Fig. 2 will be like the following: (See [7] for more details)

	<i>a</i>	<i>b</i>	<i>c</i>
<i>a</i>	$\langle 0,1 \rangle$	$\langle \mu_A(a,b), \nu_A(a,b) \rangle$	$\langle \mu_A(a,c), \nu_A(a,c) \rangle$

Definition 4

Let v be a fixed set of vertices.

Given that $(V \subset v)$ and $(A \subset V \times V)$, An IFTr T over V will be the ordered pair $T = (V^*, A^*)$, where

$$V^* = \{ \langle v, \mu_v(v), \nu_v(v) \rangle \mid v \in V \}$$

$$A^* = \{ \langle g, \mu_A(g), \nu_A(g) \rangle \mid (\exists v, w \in V)(g = \langle v, w \rangle) \in A \}$$

Where $\mu_v(v)$ and $\nu_v(v)$ are degrees of membership and non-membership of the element $v \in V$ and

$$0 \leq \mu_v(v) + v_v(v) \leq 1.$$

$\mu_A(g)$ and $v_A(g)$ are degrees of membership and non-membership of

$$g = \langle v, w \rangle \in A \text{ and } 0 \leq \mu_A(g) + v_A(g) \leq 1$$

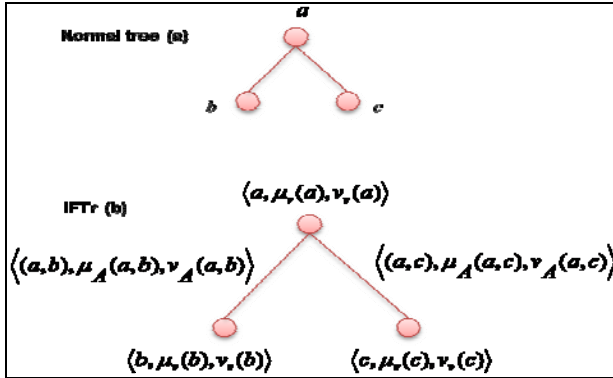


Fig. 2. Normal tree (a) vs IFTr (b)

To clarify the definition of IFTr, Fig. 2 shows a normal tree along with the correspondent IFTr. The latter indicates to what extent the normal tree (a) is included into another normal tree. Stated differently, when two normal trees are compared, the result is an IFTr. In addition to the node labels, the IFTr has functions that define the degree of membership and non-membership of each element of the tree into another tree. For calculating IFTr Inclusion we present a set of definitions. At first we define the followings:

- If n is a node in T then $l(n)$ is a function that defines the label of n .
- If m, n are nodes in T such that n is a child of m , then $A(m, n)$ will be the arch connecting m to n .
- $V(T)$ = a set of all nodes in T
- $A(T)$ = a set of all Arcs in T
- \perp = Null

Definition 5

Let T_1 and T_2 be labeled trees. We define Full Tree Inclusion (\emptyset, T_1, T_2) as an injective function $\emptyset: V(T_1) \rightarrow V(T_2)$ such that for all nodes $m, n \in V(T_1)$,

- $l(n) = l(\emptyset(n))$
- $A(m, n) = A(\emptyset(m), \emptyset(n))$

Definition 6

Let T_1 and T_2 be labeled trees. We define Partial Tree Inclusion (\emptyset, T_1, T_2) as an injective function $\emptyset: V(T_1) \rightarrow V(T_2)$ such that for all nodes $m, n \in V(T_1)$,

- $l(n) = l(\emptyset(n))$ or $\emptyset(n) = \perp$
- $A(m, n) = A(\emptyset(m), \emptyset(n))$ or $A(\emptyset(m), \emptyset(n)) = \perp$

In other words, T_1 can be partially included in T_2 if some nodes and/or arcs of T_1 exist in T_2 .

Definition 7

The degree of inclusion of a tree T_1 in another tree T_2 is $\delta(T_1, T_2)$. We define two factors that determine to which degree T_1 is included in T_2 :

- Support (S) = (# of nodes in T_1 that are included in T_2) / $|T_1|$
- Confidence (C) = (# of arcs in T_1 that are included in T_2) / ($|T_1| - 1$)

Such that: $|T_1|$ is the number of nodes in T_1 ; and ($|T_1| - 1$) is the number of arcs in T_1 i.e. number of nodes - 1.

From the previous terms, S and C, we define the followings:

- Belief (μ) = belief of T_1 being included in T_2 . S.t. $\mu = C$
- Disbelief (ν) = disbelief of T_1 being included in T_2 . S.t. $\nu = 1 - S$
- Hesitation (π) = hesitation of T_1 being included in T_2 . S.t. $\pi = S - C$
- Maximum belief (μ_{\max}) = the maximum belief of T_1 being included in T_2 .

$$\mu_{\max} = C + \pi$$

- Belief (μ) + Disbelief(ν) + Hesitation (π) = 1

Definition 8

- $\text{sim}(n, \emptyset(n))$ = Similarity between two nodes' labels $l(n), l(\emptyset(n))$. Ranges from [0, 1]
- $\text{sim}(A(m, n), A(\emptyset(m), \emptyset(n)))$ = Similarity between two Arcs $A(m, n)$ and $A(\emptyset(m), \emptyset(n))$ Ranges from [0, 1]

Intuitionistic Fuzzy Support: for every node $n \in V(T_1)$ and $\emptyset(n) \in V(T_2)$

$$(S_f) = \sum \text{sim}(n, \emptyset(n)) / |T_1|$$

Intuitionistic Fuzzy Confidence: for every arc $A(m, n) \in A(T_1)$ and $A(\emptyset(m), \emptyset(n)) \in A(T_2)$

$$(C_f) = \sum \text{sim}(A(m, n), A(\emptyset(m), \emptyset(n))) / (|T_1| - 1)$$

In the next section we provide a more detailed explanation on S_f and C_f with an algorithm for calculating them in Fig. 3 below.

IFTr Inclusion Algorithm

// This algorithm calculates S_f and C_f which imply the similarity between two trees T_1 and T_2 .

Input: Two trees T_1, T_2

Output: S_f, C_f

Begin

$S_f = C_f = \text{matchedNodes} = \text{matched_Arcs} = \text{maxSimilarity} = 0;$

// calculate S_f

For each node n in T_1 {

 For each node m in T_2 {

 If $\text{sim}(l(n), l(m)) > \text{maxSimilarity}$

$\text{maxSimilarity} = \text{sim}(l(n), l(m));$

 }

$\text{matchedNodes} += \text{maxSimilarity};$

}

// calculate C_f

For each Arc $A(m,n)$ in T_1

$\text{matched_Arcs} + = \text{arcInclusionDegree}(A(m,n), T_2);$

$S_f = \text{matchedNodes} / |T_2|;$

$C_f = \text{matched_Arcs} / (|T_2| - 1);$

End

Function $\text{sim}(l(n), l(m))$

It calculates similarity between labels of nodes n and m by first mapping the labels to a linguistic ontology and then calculating the distance between them according to the shortest path connecting them within the taxonomy.

Function $\text{arcInclusionDegree}(A(m,n), T_2)$

If $\emptyset(m) = \text{parent}(\emptyset(n))$

 Return 1; //full inclusion of the arc

Else if ($\text{isAncestor}(\emptyset(m), \emptyset(n))$

 Return $(1 / [\text{height}(\emptyset(m)) - \text{height}(\emptyset(n))])$

Fig. 3. IFTr Inclusion Algorithm

3 Intuitionistic Fuzzy Pattern Tree Inclusion

The additional benefit of using IFTr is that it gives more information about how much a pattern tree PT matches an underlying data tree DT. It provides the confirmed minimum degree to which PT is included in DT (C_f), the maximum degree of inclusion in the best case (S_f) and the degree of exclusion ($1 - S_f$) and also the hesitation (π) which implies to which extent we are not sure that there is inclusion.

Once the degree of inclusion is calculated by finding $\langle C_f, S_f \rangle$, If the degree of inclusion is higher than a predefined threshold, then the schemas are semantically close and PT counts as a witness tree; where S_f stands for Intuitionistic Fuzzy Support of an IFTr PT included in DT, and C_f stands for the Intuitionistic Fuzzy Confidence of IFTr PT being semantically included in DT. The reason of using Intuitionistic Fuzzy Techniques is to soften the traditional constraints on finding the degree of inclusion. The “source” tree does not need to be completely included in the “destination” one; it can be partially included.

Here we try to make it even more flexible by considering cases where nodes’ labels and arcs that are not fully matching. In other words, it is not necessary that:

$$l(n) = l(\mathcal{O}(n)) \text{ or } A(m, n) = A(\mathcal{O}(m), \mathcal{O}(n)).$$

We propose two ways of softening matching rules:

- **Soft Nodes Matching:** A linguistic ontology is used to compare labels of correspondent nodes and according to the distance between the labels of the two; the similarity (Semantics closeness) is calculated.
- **Soft Arc Matching:** Achieved by considering cases where separating nodes exist between nodes $\mathcal{O}(m)$, $\mathcal{O}(n)$. This however will decrease the similarity between the two nodes by a certain amount depending on the number of separating nodes.

The degree of inclusion of PT in DT, inc (PT, DT), is calculated in the form of two values $\langle C_f, S_f \rangle$. Those can result in one of the following cases:

1. If S_f & C_f are equal to 1.0, then the tree PT is fully included in DT i.e. that is an exact match and DT is a witness tree.
2. If S_f & C_f are above a user-defined threshold but C_f is less than 1.0, then the tree PT is partially included in DT, and DT is a witness tree.
3. If S_f is high (but less than 1.0) and C_f is low, this implies that the majority of PT nodes are in DT but they are misplaced; which means that there is high hesitation (π) of considering DT as a witness tree.
4. If $S_f = C_f = 0$ this means that there is no inclusion at all and DT is not a witness tree.
5. If S_f and C_f have average values then we can infer that PT is partially included in DT, which means that DT might or might not be a witness tree depending on the user-defined threshold.

Our approach has a potential to provide better performance over pattern relaxation techniques as they avoid running several queries against all underlying XML data trees. Furthermore, using two-valued measure of tree inclusion, i.e. S_f and C_f ,

provides users with the ability to balance between performance and accuracy. For example, a user can choose to retrieve DTs with high S_f if s/he is interested in high performance and not minding less accuracy, or to retrieve DTs with high C_f in case accuracy is more of interest than performance.

4 IFTr vs. Other Approaches

Many previous studies on approximate XML query matching can be found in the literature. The most cited one is probably [10, 14] by Amer-Yehia et al., which presented a number of techniques to relax queries against XML data “Query relaxation” in order to get answers from XML documents that do not 100% match the query. Queries are modeled as tree structures referred to as “Tree Patterns” (or Pattern Trees). Four relaxation techniques were provided: (i) generalize a node (ii) relax an edge (iii) make a leaf node optional and (iv) promote a subtree. The authors also proposed adding weights on the nodes and edges of tree patterns to be used as a score for exact and relaxed match. The overall score of an XML tree is the total weights of all matching nodes and edges.

Many studies followed Amer-Yehia et al. and tried to improve and suggest new relaxation techniques as well as other enhancements. For example, Fazzinga et al [16] proposed an approximate querying approach based on XPath whilst adding transformations concerning textual predicates. It also considered algorithms for combing partial answers from multiple XML sources. The same authors presented rewriting systems for XPath queries as well as a general form of rewriting rules in [17,18]. The aim is to relax the original query and translate it into more general one resulting in non-exact query answers being retrieved, ranked by their degree of relevance to the original query. Another interesting study is [15] which extends [14] by proposing a definition of a cube adapted for XML Data warehouses. This study focuses on two main challenges: Semantic, and computational. Tree pattern relaxation is used to overcome the first, whereas some computational algorithms are proposed for the latter.

The Query Relaxation techniques have managed (to some extent) to overcome the heterogeneity of XML data sources, but running multiple queries on “huge” XML repositories will definitely take the performance down. Therefore, the number of relaxed queries has to be reduced to minimum or some other solutions need to be found.

In [21], an adaptive query relaxation (AQR) approach is proposed which relaxes a query adaptively to agree with the structure of the underlying XML data sources. AQR outperformed the aforementioned studies by avoiding blind pattern tree relaxations. Each generated relaxed query is specific to a data source, thus avoiding queries with no answers. It also introduced node relaxation by using the semantics of node labels based on the taxonomy ontology WordNet [9]. Also considered node similarity measures such as annotation and sibling measures. Reference [19] followed a slightly similar way of retrieving inexact answers by calculating the similarity between a pattern tree and XML subtrees. Authors calculated similarity between

nodes (vertices) not only depending on the label, but also depending on the depth of the node “distance-based similarity” so that the similarity linearly decreases as the number of levels of difference increase.

The following is a comparison between our approach (IFTr) and other previous ones. Four main approaches can be identified in the literature as shown in Table (1) below. (i) Query Relaxation (QR) (ii) Query Rewriting Rules (QRR) (iii) Adaptive Query Relaxation (AQR) and (iv) Tree Similarity Matching (TSM). We carry out the comparison according to the following features:

Performance: This refers to the number of times data trees are scanned (accessed). For QR studies, the value can range from 1 to $2^{(n-1)}$ where n refers to the number of nodes in the pattern tree. On the other hand, the performance of QRR and ATR studies depends on the number of optional elements in the underlying DTD schemas. If that is big, then the number of scans will be high (m). For TSM, three measures of similarity are involved: match-based, level-based and distance based similarity. This can be time consuming and may require multiple scans of data trees. In contrast, the IFTr approach would need one or two scans only. If the user is interested in quick results then the data trees are scanned only one time to calculate the support (S_f). If more accurate results are required then another scan is performed to calculate the confidence (C_f).

Accuracy: While QRR and AQR can provide accurate results as per the authors’ experiments, QR and TSM do not provide such accuracy. For QR, the more the relaxed queries the less the accuracy while in the latter, three measures of similarity are proposed. Those are based on node labels, levels and positions. There are no arc (edge) matching measures which we believe will not provide accurate results.

Use of semantic knowledge: This refers to the inclusion of lexical knowledge (ontology) for resolving the problem of synonyms, homonyms etc.

Adaptation (schema-awareness): this refers to whether the generated relaxed queries are schema-aware i.e. they are generated based on DTD schemas. For TSM and IFTr, there is no generation of relaxed queries. However, those approaches compare the pattern tree to the DTDs of the underlying XML documents which means that they are schema-aware approaches.

Table 1. Comparison between IFTr and other approaches

Approach	Performance	Accuracy	Use of semantic knowledge	Adaptation
QR [14, 15, 26]	$1 - 2^{(n-1)}$	Normal	No	No
QRR [16-18]	1 – m	High	Yes	Yes
AQR[21]	1 – m	High	Yes	Yes
TSM [19]	1 – m	Normal	Yes	Yes
IFTr	1– 2	High	Yes	Yes

As shown in the table above, the IFTr approach is anticipated to provide better results over others in regards to all features. It avoids running blind query relaxations and only requires the underlying data trees to be scanned no more than two times. It also adopts semantic knowledge to achieve high accuracy and schema-aware techniques to further enhance performance. The resultant balance between performance and accuracy is expected to meet different user requirements ranging from quick and less accurate to very accurate but time consuming results in terms of the number of data trees scans.

5 Conclusion and Further Work

The increasing popularity of XML as a standard for data representation resulted in huge amounts of XML documents following different structures. Many previous works proposed query relaxation as a solution to achieve XML approximate query match, which has proved to be a good approach to some extent. However, it consisted of running a number of queries against the entire underlying data, which obviously affected performance. In XML documents with complex schemas, the aforementioned approach is further inappropriate. IFTr is anticipated to achieve approximate query matching on highly varied XML schemas with less influence of performance, in addition to allowing the user to choose between accuracy and performance.

For future work, we intend to extend our approach to enable resolving hierarchical heterogeneities in XML documents by applying transformations on hierarchical levels to enable retrieving combined data from such documents.

References

1. López, F., Laurent, A., Poncelet, P., Teisseire, M.: FTMnodes: Fuzzy tree mining based on partial inclusion. *Fuzzy Sets and Systems* 160, 2224–2240 (2009)
2. Shannon, A., Atanassov, K.: A first step to a theory of the Intuitionistic fuzzy graphs. In: *The First Workshop on Fuzzy Based Expert Systems*, Sofia, pp. 59–61 (1994)
3. Atanassov, K.: On Intuitionistic fuzzy graphs and Intuitionistic fuzzy relations. In: *IFSA World Congress*, Sao Paulo, vol. 1, pp. 551–554 (1995)
4. Atanassov, K.: Temporal intuitionistic fuzzy graphs. *Notes on Intuitionistic Fuzzy Sets* 4(4), 59–61 (1998)
5. Atanassov, K.: *Intuitionistic Fuzzy Sets*. Springer Physica-Verlag, Berlin (1999)
6. Atanassov, K.: On index matrix interpretations of Intuitionistic fuzzy graphs. *Notes on Intuitionistic Fuzzy Sets* 8(4), 73–78 (2002)
7. Chountas, P., Alzebdi, M., Shannon, A., Atanassov, K.: On Intuitionistic fuzzy trees. *Notes on Intuitionistic Fuzzy Sets* 15(2), 30–32 (2009)
8. XQuery 1.0: An XML Query Language, 2nd edn., <http://www.w3.org/TR/xquery/#id-introduction>
9. WordNet, <http://wordnet.princeton.edu/>
10. Amer-Yahia, S., Lakshmanan, L., Pandit, S.: FleXPath: Flexible Structure and Full Text Querying for XML. In: *Proc. Of the 2004 ACM SIGMOD International Conference on Management of Data*, pp. 83–94. ACM, New York (2004)

11. Nierman, H., Jagadish, V.: Structural Similarity between XML Documents and DTDs. In: Sloot, P.M.A., Abramson, D., Bogdanov, A.V., Gorbachev, Y.E., Dongarra, J., Zomaya, A.Y. (eds.) ICCS 2003. LNCS, vol. 2659, pp. 412–421. Springer, Heidelberg (2003)
12. Chen, Y., Chen, Y.: A new tree inclusion algorithm. Elsevier Information Processing Letters 98, 253–262 (2006)
13. Bille, P.: A survey on tree edit distance. Elsevier, Theoretical Computer Science 337, 217–239 (2005)
14. Amer-Yahia, S., Cho, S., Srivastava, D.: Tree Pattern Relaxation. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 496–513. Springer, Heidelberg (2002)
15. Wiwatwattana, N., Jagadish, H.V., Lakshmanan, L.V.S., Srivastava, D.: X³: A cube operator for XML OLAP. In: IEEE 23rd International Conference on Data Engineering, pp. 916–925 (2007)
16. Fazzinga, B., Flesca, S., Pugliese, A.: Retrieving XML Data from Heterogeneous Sources through Vague Querying. ACM Transactions on Internet Technology 9(2), Article 7 (2009)
17. Fazzinga, B., Flesca, S., Furfaro, F.: On the expressiveness of generalization rules for XPath query relaxation. In: IDEAS 2010, Montreal, Canada (2010)
18. Fazzinga, B., Flesca, S., Furfaro, F.: XPath query relaxation through rewriting rules. IEEE Transactions on Knowledge and Data Engineering (2010)
19. Sanz, I., Fischer, F., Mesiti, M., Guerrini, G., Lavori, R.B.: Approximate Subtree Identification in Heterogeneous XML Documents Collections. In: Bressan, S., Ceri, S., Hunt, E., Ives, Z.G., Bellahsene, Z., Rys, M., Unland, R. (eds.) XSym 2005. LNCS, vol. 3671, pp. 192–206. Springer, Heidelberg (2005)
20. Algergawy, A., Nayak, R., Saake, G.: Element similarity measures in XML schema matching. Elsevier, Information Sciences 180, 4975–4998 (2010)
21. Liu, C., Li, J., Xu Yu, J., Zhou, R.: Adaptive relaxation for querying heterogeneous XML data sources. Elsevier, Information Systems 35, 688–707 (2010)
22. Alzebedi, M., Chountas, P., Atanassov, K.T.: Enhancing DWH models with the utilisation of multiple hierarchical schemata. In: IEEE SMC 2010, Istanbul, pp. 488–492 (2010)
23. Introduction to XML, http://www.w3schools.com/xml/xml_what_is.asp

A Cooperative Answering Approach to Fuzzy Preferences Queries in Service Discovery

Katia Abbaci¹, Fernando Lemos², Allel Hadjali¹, Daniela Grigori²,
Ludovic Liétard³, Daniel Rocacher¹, and Mokrane Bouzeghoub²

¹ IRISA/ENSSAT, Rue de Kérampont BP 80518 Lannion, France
{katia.abbaci, allel.hadjali, daniel.rocacher}@enssat.fr

² PRiSM Lab, 45 Av. des États Unis 78000 Versailles, France
{fernando.lemos, daniela.grigori,
mokrane.bouzeghoub}@prism.uvsq.fr

³ IRISA/IUT, Rue Edouard Branly BP 30219 Lannion, France
ludovic.lietard@univ-rennes1.fr

Abstract. In this paper, we propose a novel approach for service retrieval that takes into account the service behavior (described as process model) and relies both on preference satisfiability and structural similarity. User query and target process models are represented as annotated graphs, where user preferences on QoS (Quality of Service) attributes (such as response time, availability and throughput) are modelled by means of fuzzy sets. To avoid empty results, a flexible evaluation method based on fuzzy linguistic quantifiers (such as *almost all*) is introduced. The retrieved results are easily interpreted by the end users thanks to the clear semantics conveyed by that method. Finally, two families of ranking methods are discussed.

Keywords: Cooperative answering, service retrieval, quality of services, fuzzy preferences, linguistic quantifiers.

1 Introduction

Nowadays, an increasing number of companies and organizations are moving towards a service-oriented and model-driven architectures for offering their services on the Web. Searching a specific service within service repositories becomes a critical issue for the success of these architectures. This issue has recently received much attention and many approaches have been proposed [8,25]. Most of these approaches are based on the matchmaking of process inputs/outputs [8], service behavior [2] or ontological knowledge [5]. Unfortunately, these approaches often result in a large number of services offering similar functionalities and behavior. One way to discriminate between such similar services is to consider non-functional requirements such as QoS (Quality of Service) (e.g., response time, throughput, availability and reliability). A recent trend towards quality-aware approaches has been initiated [13,11,8], but remains limited and not satisfactory for generic process model discovery.

On the other hand, several service discovery approaches based on fuzzy set theory have been proposed. For instance, in [11] the authors treat the web service selection

for composition as a fuzzy constraint satisfiability problem. They assign to each QoS criterion five fuzzy sets (such as *poorly acceptable*, *almost acceptable* and *acceptable*) describing its constraint levels. In [15], QoS based service selection is modelled as a fuzzy multiple criteria decision making problem. Linguistic expressions are used to evaluate and to express the weights of importance of QoS criteria. Hafeez et al. [6] present a service selection mechanism allowing the service broker to intelligently select a set of available services from a user query with imprecise constraints defined by fuzzy sets. The query evaluation is based on the aggregation of the obtained degrees over constraints. Şora et al. [11] propose an approach in which they automatically generate fuzzy rules from user preferences and rank the candidate services using a fuzzy inference process.

The above fuzzy approaches only consider the preference satisfiability and ignore the structural similarity of complex web services. Moreover, these works deal only with services as black boxes, i.e., the service behavior level is not investigated. Our goal is to go further these approaches into a unique integrated approach dealing with functional and non-functional requirements and behavior specification in service retrieval.

Starting from the work done in [9], we propose a cooperative approach for handling users process queries where both behavior specification and QoS preferences are specified inside these queries. User preferences on QoS properties are modelled by means of fuzzy sets as they are more suitable to the interpretation of linguistic terms (such as high or fast) that constitutes a convenient way for users to express their preferences. To avoid empty answers for a given query, a flexible evaluation strategy based on fuzzy linguistic quantifiers is introduced.

The remainder of this paper is organized as follows. Section 2 provides some basic background. In Section 3, modelling fuzzy preferences and their evaluation are addressed. Section 4 presents our interpretation of process models similarity based on linguistic quantifiers. In Section 5, service ranking methods are discussed. Section 6 proposes an illustrative example and finally Section 7 concludes the paper.

2 Background

In this section, we provide some basic definitions within the scope of web service selection with preferences, and a short recall on fuzzy sets.

2.1 Fuzzy Sets

A fuzzy set F [17] on the universe X is described by a membership function $\mu_F : X \rightarrow [0, 1]$, where $\mu_F(x)$ represents the **membership degree** of x in F . The set $\{x \in F | \mu_F(x) > 0\}$ (resp. $\{x \in F | \mu_F(x) = 1\}$) represents the **support** (resp. **core**) of F . In practice, the membership function associated to F is often represented by a trapezoid $(\alpha, \beta, \varphi, \psi)$ ¹, where $[\alpha, \psi]$ (resp. $[\beta, \varphi]$) is its support (resp. core).

A Fuzzy set-based approach to preferences queries proposed in [3] relies on the use of fuzzy set membership functions that describe the preference profiles of the user on each attribute domain involved in the query. This is especially convenient and suitable

¹ In our case, the quadruplet $(\alpha, \beta, \varphi, \psi)$ is user-defined to ensure the subjectivity property.

when dealing with numerical domains, where a continuum of values is to be interfaced for each domain with satisfiability degrees in the unit interval scale. Then individual satisfiability degrees associated with elementary conditions are combined (commensurability assumption holds thanks to the membership functions) using a panoply of fuzzy set connectives, which may go beyond conjunctive and disjunctive aggregations (by possibly involving fuzzy quantifiers, if only the satisfiability of the most of the elementary conditions in a query is required).

2.2 Preferences in Process Model Specification

Many languages are currently available to describe service process models, e.g., OWL-S [12]. They represent a process model as a set of primitive activities combined using control flow structures. Then, these languages can be abstracted as a direct graph $G = (V, E)$, where the vertices represent activities or control flow nodes, while the edges represent the flow of execution. In this work, services are specified as graphs annotated with QoS properties and user queries are specified as graphs annotated with preferences.

Figure 1 shows an example of a user query annotated with preferences. The example presents a *global preference* indicating user prefers services providing RSA encryption. Some *activity preferences* are also defined for activities *A* and *B* involving reliability, response time and cost. Figure 2 shows an example of a process model annotated with QoS attributes. The example presents a *global annotation* indicating the security of the process model and *activity annotations* indicating the response time, reliability and cost of some activities. In what follows, we present the formal definitions of our model.

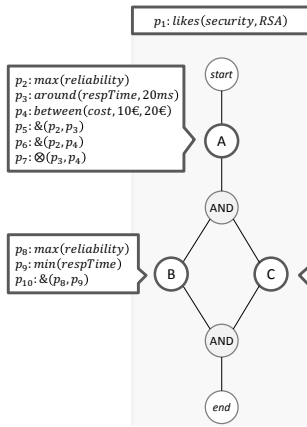


Fig. 1. Query Graph q_1

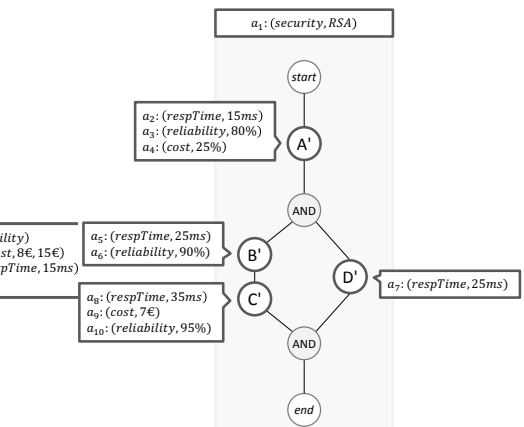


Fig. 2. Target Graph t_1

Definition 1. An *annotation* is a pair (m, r) , where m is a QoS attribute obtained from an ontology O and r is a value for m . It can be specified over a process model graph (*global annotation*) or over an atomic activity (*activity annotation*).

² We abstract from the different units in which a value can be described.

Definition 2. A *preference* is an expression that represents a desire of the user over the QoS attributes of a process model or activity. It can be of one of the following forms³:

- *atomic preferences*:
 - *around* $(m, r_{desired}, \mu_{around})$: for attribute m , this expression favors the value $r_{desired}$; otherwise, it favors the values close to $r_{desired}$. μ_{around} evaluates the degree to which a value r satisfies $r_{desired}$;
 - *between* $(m, r_{low}, r_{up}, \mu_{between})$: for attribute m , this expression favors the values inside the interval $[r_{low}, r_{up}]$; otherwise, it favors the values close to the limits. $\mu_{between}$ evaluates the degree to which a value r satisfies the interval $[r_{low}, r_{up}]$;
 - *max* (m, μ_{max}) : for attribute m , this expression favors the highest value; otherwise, the closest value to the maximum is favored, as example: the maximum of reliability or availability is equal by default to 100%. μ_{max} evaluates the degree to which a value r satisfies the highest value of m ;
 - *min* (m, μ_{min}) : for attribute m , this expression favors the lowest value; otherwise, the closest value to the minimum is favored, as example: the minimum of response time or cost is equal by default to 0. μ_{min} evaluates the degree to which a value r satisfies the lowest value of m ;
 - *likes* $(m, r_{desired})$: for attribute m , this expression favors the value $r_{desired}$; otherwise, any other value is accepted to some extent;
 - *dislikes* $(m, r_{undesired})$: for attribute m , this expression favors the values that are not equal to $r_{undesired}$; otherwise, $r_{undesired}$ is accepted to some extent;
- *complex preferences*:
 - *Pareto preference* $\otimes (p_i, p_j)$: this expression states that the two preference expressions p_i and p_j are equally important;
 - *prioritized preference* $\& (p_i, p_j)$: this expression states that the preference p_i is more important than the preference p_j .

A preference can be specified over a process model graph (**global preference**) or over an atomic activity (**activity preference**).

In [9], this set of preferences has been used to develop a service selection approach based on QoS where preference satisfiability is computed using to a unique distance function for all numerical preferences. This way of doing does not take into account the fact that preferences are context and user-dependent and assumes no commensurability when combining individual satisfiability degrees.

3 A Fuzzy Model to Evaluate Preferences

In this section, we introduce a fuzzy set-based approach to handle the above set of preferences involved in the annotated graph associated with the user query. In particular, we propose a metric, called *satisfiability degree* (δ), that measures how well the annotations of a target process model satisfy the preferences present in the query.

³ Based on a subset of preference operators of the model by [7] that leads to a partial order.

3.1 Atomic Preferences

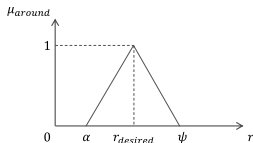
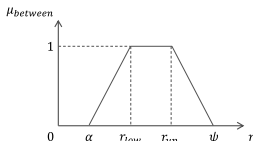
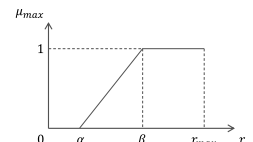
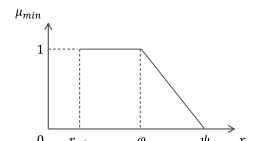
For numerical atomic preferences, the satisfiability degree is obtained using to user-specific membership functions. Table 1 summarizes the fuzzy modelling of numerical preferences of interest. Given a preference p and an annotation $a : (m, r)$, one is interested in computing the degree to which the annotation a satisfies p .

For non-numerical preferences, the satisfiability degree is based on the semantic similarity between concepts. Given an ontology O and two concepts c_1 and c_2 , the semantic similarity wp between c_1 and c_2 is given by [14]:

$$wp(O, c_1, c_2) = \frac{2N_3}{N_1 + N_2 + 2N_3} \tag{1}$$

where c_3 is the least common super-concept between c_1 and c_2 , N_1 is the length of the path from c_1 to c_3 , N_2 is the length of the path from c_2 to c_3 , and N_3 is the length of the path from c_3 to the ontology root. Given a non-numerical preference p and an annotation a , the satisfiability degree $\delta(p, a)$ is calculated as shown in Table 2.

Table 1. Fuzzy modelling of numerical preferences

NUMERICAL PREFERENCE	FUZZY INTERPRETATION
$around(m, r_{desired}, \mu_{around})$	$\mu_{around}(r) = \begin{cases} 0, & r \leq \alpha, r \geq \psi \\ \frac{r - \alpha}{r_{desired} - \alpha}, & \alpha < r < r_{desired} \\ 1, & r = r_{desired} \\ \frac{\psi - r}{\psi - r_{desired}}, & r_{desired} < r < \psi \end{cases}$ 
$between(m, r_{low}, r_{up}, \mu_{between})$	$\mu_{between}(r) = \begin{cases} 0, & r \leq \alpha, r \geq \psi \\ \frac{r - \alpha}{r_{low} - \alpha}, & \alpha < r < r_{low} \\ 1, & r_{low} \leq r \leq r_{up} \\ \frac{\psi - r}{\psi - r_{up}}, & r_{up} < r < \psi \end{cases}$ 
$max(m, \mu_{max})$	$\mu_{max}(r) = \begin{cases} 0, & r \leq \alpha \\ \frac{r - \alpha}{\beta - \alpha}, & \alpha < r < \beta \\ 1, & \beta \leq r \leq r_{max} \end{cases}$ 
$min(m, \mu_{min})$	$\mu_{min}(r) = \begin{cases} 1, & r_{min} \leq r \leq \varphi \\ \frac{\psi - r}{\psi - \varphi}, & \alpha < r < \psi \\ 0, & r \geq \psi \end{cases}$ 

3.2 Complex Preferences

To evaluate a set of complex preferences S_p , first we construct a *preference tree* t_p that represents the semantics of the set S_p . In that tree, the nodes represent atomic preferences and the edges represent a *more important than relation (prioritized preference)*

Table 2. Satisfiability degree of a non-numerical preference p

Non-numerical Preference p	satisfiability Degree $\delta(p, a)$
$likes(m, r_{desired})$	$\delta(p, a) = \begin{cases} 1, & r_{desired} = r \\ wp(O, r_{desired}, r), & otherwise \end{cases}$
$dislikes(m, r_{undesired})$	$\delta(p, a) = 1 - likes(m, r_{undesired})$

from parent to child. Preferences of the same level and having the same parent express *Pareto preference*. Each level i (except the root) of the tree is associated with an importance weight $\omega_i = 1/i$, except $i = 0$ (the smaller i , the more important p_i).

For example, consider the preference tree of q_1 in Figure 3, obtained from the complex preferences of query q_1 . Preferences p_{11} is an atomic preference that is not component of any complex preference. $p_5 : \&(p_2, p_3)$ is a complex preference composed of atomic preferences p_2 and p_3 ; it means that p_2 is more important than p_3 . $p_7 : \otimes(p_3, p_4)$ is a complex preference composed of atomic preferences p_3 and p_4 ; it means that p_3 and p_4 are equally important.

Considering that each atomic preference p_i has a satisfiability degree δ_i , a new satisfiability degree δ'_i is computed taking into account the weight ω_i underlying p_i in the spirit of [3]. δ'_i is defined [4] using the formula (2).

$$\delta'_i = \max(\delta_i, 1 - \omega_i) \tag{2}$$

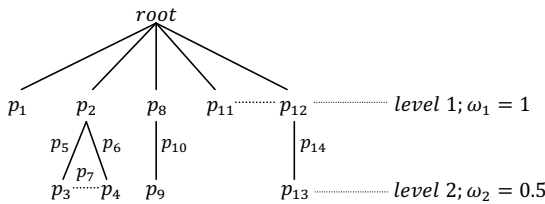


Fig. 3. Sample preference tree

This new interpretation of p_i considers as acceptable any value outside of its support with the degree $1 - \omega_i$. It means that the larger ω_i (i.e., p_i is important), the smaller the degree of acceptability of a value outside the support of p_i . At the end, we calculate the satisfiability degree of user atomic preferences considering their constructors and the complex preferences containing them.

4 Process Model Similarity: A Linguistic Quantifier-Based Method

In this section, we describe a method to compute similarity between process model graphs according to user preferences. We also discuss a method to assess the structural

⁴ We assume here that $\max_{i=1, n} \omega_i = 1$.

similarity between two process model graphs. Both kinds of similarity will be used to rank potential targets of a query as it will be seen in Section 5.

In order to evaluate the structural similarity of two processes q and t , we propose to use a graph matching algorithm, like in [5]. This algorithm returns a mapping M and a set E of edit operations necessary to transform q into t . We consider a mapping M between q and t as a set of pairs (v, w) , such that v is an activity of q and w is an activity of t or the symbol $\$$, which indicates the deletion of v . The edit operations considered are simple graph edit operations: node/edge deletion, node/edge addition and node substitution. Figure 4 illustrates a mapping between a query graph q_1 and a target graph t_1 . In the figure, $SS(v, w)$ denotes the semantic similarity between activities v and w ; we use the metric proposed in [5] that considers the activity name, inputs and outputs. In our work, the preference evaluation explained in Section 3 is applied as follows: the global atomic preferences of q are evaluated against the global annotations of t ; similarly, the atomic preferences of an activity v of q are evaluated against the annotations of an activity w of t , such that $(v, w) \in M$.

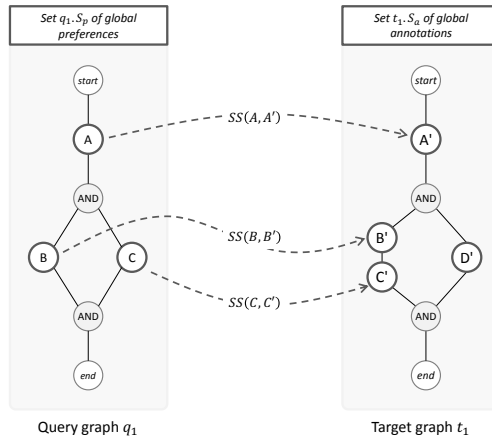


Fig. 4. Sample mapping M between a query graph q_1 and a target graph t_1

In our approach, we rely on the linguistic quantifier “almost all” for the similarity evaluation process. Such a quantifier, which is seen as a relaxation form of the universal quantifier “all”, constitutes an appropriate tool to avoid the empty answers. It allows to retrieve elements that would not be selected when the quantifier “all” is used.

4.1 Preference Satisfiability between Process Models

A natural user interpretation of the similarity between query and target process models according to user preferences is given by the truth degree of the following proposition:

$$\gamma_1: \text{Almost all preferences of } q \text{ are satisfied by } t$$

The above statement is a fuzzy quantified proposition of the form $Q X \text{ are } P$, where (i) Q is a relative quantifier (e.g., almost all, around half, etc.) [4]; (ii) X is a set of elements; (iii) P is a fuzzy predicate. The truth degree δ_γ of $\gamma : Q X \text{ are } P$ is computed according to Yager's method [16]:

- Let $\Omega = \{\mu_1, \dots, \mu_n\}$ be a set of degrees of the elements of X w.r.t. P , ordered in decreasing way: $(\mu_1 \geq \dots \geq \mu_n)$.
- The truth degree δ_γ is given by formula (3), where $\mu_Q(i/n)$ is a membership degree of the element i/n to Q .

$$\delta_\gamma = \max_{1 \leq i \leq n} \min(\mu_i, \mu_Q(i/n)) \tag{3}$$

In our case, $\Omega = \{\mu_1 : \delta'_1, \dots, \mu_n : \delta'_n\}$ is the set of satisfiability degrees of all atomic preferences (i.e. all global and activity atomic preferences) of query q , where δ'_i is the satisfiability degree of an atomic preference p_i computed by formula (2). The semantics of the linguistic quantifier *almost all* is given by (50%, 80%, 100%, 100%). In this case, (i) the user is totally satisfied if at least 80% of preferences are satisfied and (ii) the user is not satisfied at all if at most 50% of preferences are satisfied.

4.2 Structural Similarity between Process Models

Similarly, we can apply the technique based on fuzzy quantifiers to compute a structural similarity degree between two process models. This similarity between a query and target process models can be given by the truth degree of the following propositions:

$$\begin{cases} \gamma_2 : \text{Almost all the activities of } q \text{ are mapped with activities of } t \\ \gamma_3 : \text{Almost no edit operation is necessary to transform } q \text{ into } t \end{cases}$$

The truth degree of proposition γ_2 is obtained from the formula (3), where $\Omega = \{\mu_1 : SS_1, \dots, \mu_n : SS_n\}$ is the set of semantic similarity degrees of all mapped activities of q , and SS_i is the semantic similarity degree of a query activity v mapped with a target activity w . In the case of proposition γ_3 , the expression "almost no edit operation is necessary to transform q into t " is equivalent to the expression "almost all edit operations are *not* necessary to transform q into t ". Therefore, its truth degree is computed as follows:

$$\delta_{\gamma_3} = \max_{1 \leq i \leq n} \min(1 - \mu_i, 1 - \mu_Q(i/n)) \tag{4}$$

In this case, $\Omega = \{\mu_1 : C_1, \dots, \mu_n : C_n\}$ is the set of transformation costs of mapped target activities with the corresponding activities of q , and C_i is the transformation cost of a target activity w into a query activity v .

Thus, the structural similarity between q and t is evaluated as follows:

$$SS = \min(\delta_{\gamma_2}, \delta_{\gamma_3}) \tag{5}$$

Remark. In our approach, we consider particularly formulae (3) and (4) where $\mu_Q(i/n) = i/n$. Thus, the meaning of delivered degrees has a simple and clear semantics for the user [10]. For instance, the evaluation of γ_1 , γ_2 and γ_3 means that:

At least $\delta_{\gamma_1}^*$ % of preferences of q are satisfied by t to at least a degree δ_{γ_1} , at least $\delta_{\gamma_2}^*$ % of activities of q are mapped with t to at least a degree δ_{γ_2} and at least $\delta_{\gamma_3}^*$ % of q does not need edit operation to transform q into t to at least a degree δ_{γ_3} (where $\delta_{\gamma_i}^* = 100 \times \delta_{\gamma_i}$).

5 Process Model Ranking

In this section, given a set of target graphs that are relevant to a query, we discuss some methods to rank-order these graphs according to their structural similarity and preference satisfiability. Let $\delta(q, t, M)$ be the satisfiability degree between query graph q and target graph t obtained by formula (3) according to a mapping M . Similarly, let $SS(q, t, M, E)$ be the structural similarity between q and t obtained by formula (5) w.r.t. M and a set E of edit operations. Two kinds of ranking methods can be used.

Ranking methods based on aggregation. In this first category, ranking methods aggregate both structural similarity and preference satisfiability into a unique degree used to rank-order the target graphs. Two aggregations can be considered:

Weighted average-based aggregation. The weighted average of $SS(q, t, M, E)$ and $\delta(q, t, M)$ is given by equation (6).

$$rank(q, t) = \omega_{SS} \times SS(q, t, M, E) + (1 - \omega_{SS}) \times \delta(q, t, M) \quad (6)$$

s. t. $0 < \omega_{SS} < 1$ is an importance weight assigned to the structural similarity criterion.

Min-combination based aggregation. The min-combination method [17] selects the smallest value of the two similarity degrees $SS(q, t, M, E)$ and $\delta(q, t, M)$, i.e.,

$$rank(q, t) = \min(SS(q, t, M, E), \delta(q, t, M)) \quad (7)$$

Ranking method without aggregation. In this second category, the two distinct similarity degrees are used to rank-order target graphs thanks to the *lexicographic order*. A priority is given to the structural similarity $SS(q, t, M, E)$ while the preference satisfiability $\delta(q, t, M)$ is only used to break ties.

6 Illustrative Example

We give here an example of service discovery for query q_1 of Figure 1. First, we compute the preference satisfiability degree between q_1 and the potential target graphs. To illustrate, we evaluate the preference satisfiability degree between q_1 and target t_1 of Figure 2. We consider the mapping between them as depicted in Figure 4. Then, we apply the ranking methods described in Section 5. See below for more details.

First, we compute the satisfiability degree δ of user preferences as shown in Table 3. Consider, for example, the pair (A, A') in Table 3. The satisfiability degree $\delta(p_2, a_2)$ between preference p_2 and annotation a_2 is obtained by $\mu_{max}[\text{reliability}]$. According to equation (2), a preference tree allows to aggregate the preference degrees of A . The result is presented in column RESULT in Table 3. Second, we apply the truth degree

Table 3. Satisfiability degrees of each pair of matched activities

SATISFIABILITY DEGREE CALCULATION				
ATOMIC PREFERENCES			COMPLEX PREFERENCES	
PRE F.	MEMBERSHIP FUNCTION	δ_i	PREFERENCE TREE	δ'_i
p_1		$\delta(p_1, a_1) = 1$		$\delta'_1 = 1$
p_2		$\delta(p_2, a_3) = 1$		$\delta'_2 = 1$
p_3		$\delta(p_3, a_2) = 0.5$		$\delta'_3 = 0.5$
p_4		$\delta(p_4, a_4) = 0.5$		$\delta'_4 = 0.5$
p_5	The same of preference p_2 .	$\delta(p_5, a_6) = 1$		$\delta'_5 = 1$
p_6		$\delta(p_6, a_5) = 0.9$		$\delta'_6 = 0.9$
p_{11}	The same of preference p_2 .	$\delta(p_{11}, a_{10}) = 1$		$\delta'_{11} = 1$
p_{12}		$\delta(p_{12}, a_9) = 0.75$		$\delta'_{12} = 0.75$
p_{13}		$\delta(p_{13}, a_8) = 0$		$\delta'_{13} = 0.5$

described in Section 4.1 to obtain the global satisfiability degree between q_1 and t_1 , as follows: $\delta_{\gamma_1}(q_1, t_1) = \max(\min(1, \mu_Q(1/9)), \dots, \min(0.5, \mu_Q(9/9))) = 0.67$. This means that at least 67% of preferences are satisfied to at least a degree 0.67.

Assume now that the semantic similarities between activities are given by $SS(A, A') = 0.72$, $SS(B, B') = 0.85$ and $SS(C, C') = 0.66$, and the costs of transformation of target activities are $C(start) = 0$, $C(A') = 0$, $C(AND-split) = 0.1$, $C(B') = 0.2$, $C(C') = 0.2$, $C(D') = 0.4$, $C(AND-join) = 0.1$ and $C(end) = 0$. In a similar way, the structural similarity degree between q_1 and t_1 is obtained as: $\delta_{\gamma_2}(q_1, t_1) = \max(\min(0.85, \mu_Q(1/3)), \dots, \min(0.66, \mu_Q(3/3))) = 0.66$ and $\delta_{\gamma_3}(q_1, t_1) = \max(\min(1 - 0.4, 1 - \mu_Q(1/8)), \dots, \min(1 - 0, 1 - \mu_Q(8/8))) = 0.75$.

Table 4. Structural similarity and preference satisfiability degrees of a set of target graphs **Table 5.** Ranking of target graphs according to the three ranking methods

TARGET GRAPH	STRUCTURAL SIMILARITY SS	SATISFIABILITY DEGREE δ
t_1	0.66	0.67
t_2	0.29	0.72
t_3	0.85	0.40
t_4	0.78	0.35
t_5	0.78	0.21
t_6	0.68	0.72
t_7	0.66	0.72
t_8	0.66	0.35

WEIGHTED AVERAGE	MIN-COMBINATION	LEXICOGRAPHIC ORDER
t_3 $wa = 0.74$	t_6 $mc = 0.68$	t_3
t_6 $wa = 0.69$	t_7 $mc = 0.66$	t_4
t_7 $wa = 0.68$	t_1 $mc = 0.66$	t_5
t_4 $wa = 0.67$	t_3 $mc = 0.40$	t_6
t_1 $wa = 0.66$	t_4 $mc = 0.35$	t_7
t_5 $wa = 0.64$	t_8 $mc = 0.35$	t_8
t_8 $wa = 0.58$	t_2 $mc = 0.29$	t_1
t_2 $wa = 0.40$	t_5 $mc = 0.21$	t_2

Now, $SS(q, t, M, E) = \min(\delta_{\gamma_2}(q_1, t_1), \delta_{\gamma_3}(q_1, t_1)) = 0.66$. It means that at least 66% of query activities are mapped to at least a degree 0.66 and at most 66% of target activities have transformation cost to at most 0.66.

As presented in Table 4, eight potential answers to query q_1 are retrieved. Table 5 summarizes the results of the different ranking methods discussed in Section 5 (where $\omega_{SS} = 0.75$).

7 Conclusion

In this paper, we have proposed an approach for web services selection and ranking where both structural similarity and preference satisfiability are taken into account in the evaluation step. User preferences are modelled thanks to fuzzy predicates while linguistic quantifiers are used as a basis to compute the process model similarity. So, the matchmaking process is achieved in a more cooperative and flexible way. Some ranking methods have been discussed in the scope of services retrieval. We are currently working on a prototype system to evaluate our approach by conducting some experiments.

References

- Şora, I., Lazăr, G., Lung, S.: Mapping a fuzzy logic approach for qos-aware service selection on current web service standards. In: ICC-CONTI, pp. 553–558 (2010)
- Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
- Dubois, D., Prade, H.: Using fuzzy sets in flexible querying: Why and how? In: Proc. of FQAS, pp. 89–103 (1996)
- Glöckner, I.: Fuzzy Quantifiers in Natural Language: Semantics and Computational Models. Der Andere Verlag, Osnabrück (2004)
- Grigori, D., Corrales, J.C., Bouzeghoub, M., Gater, A.: Ranking bpel processes for service discovery. IEEE Transactions on Services Computing 3, 178–192 (2010)
- Hafeez, O., Chung, S., Cock, M.D., Davalos, S.: Towards an intelligent service broker with imprecise constraints: Fuzzy logic based service selection by using sawsdl. Tcss 702 design project in computing and software systems. University of Washington (2008)

7. Kießling, W.: Foundations of preferences in database systems. In: VLDB, pp. 311–322. VLDB Endowment (2002)
8. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with owls-mx. In: Proc. of AAMAS, pp. 915–922 (2006)
9. Lemos, F., Gater, A., Grigori, D., Bouzeghoub, M.: Adding preferences to semantic process model matchmaking. In: Proc. of GAOC (2011)
10. Liétard, L.: A new definition for linguistic summaries of data. IEEE World Congress on Computational Intelligence, Fuzzy-IEEE, Hong-Kong, China (2008)
11. Lin, M., Xie, J., Guo, H., Wang, H.: Solving qos-driven web service dynamic composition as fuzzy constraint satisfaction. Proc. of EEE, 9–14 (2005)
12. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T.R., Sirin, E., Srinivasan, N., Sycara, K.: Owl-s: Semantic markup for web services
13. Mokhtar, S.B., Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y.: Easy: Efficient semantic service discovery in pervasive computing environments with qos and context support. Journal of Systems and Software 81(5), 785–808 (2008)
14. Wu, Z., Palmer, M.S.: Verb semantics and lexical selection. In: Proc. of ACL, pp. 133–138 (1994)
15. Xiong, P., Fanin, Y.: Qos-aware web service selection by a synthetic weight. In: Proc. of FSKD, vol. (3), pp. 632–637 (2007)
16. Yager, R.R.: General multiple-objective decision functions and linguistically quantified statements. International Journal of Man-Machine Studies 21, 389–400 (1984)
17. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
18. Zhang, Y., Huang, H., Yang, D., Zhang, H., Chao, H.-C., Huang, Y.-M.: Bring qos to p2p-based semantic service discovery for the universal network. Personal Ubiquitous Computing 13(7), 471–477 (2009)

Fuzzy Orderings for Fuzzy Gradual Patterns

Malaquias Quintero, Anne Laurent, and Pascal Poncelet

University Montpellier 2
LIRMM - CNRS UMR 5506,
161, rue Ada, F-34392 Montpellier Cedex 5, France
Apizaco Institute of Technology, Mexico
{quinterofl, laurent, poncelet}@lirmm.fr
<http://www.lirmm.fr>

Abstract. In mining *gradual patterns* the idea is to express co-variations of attributes, taking the *direction of change* of attribute values into account. These patterns are such as {*the more A, the more B*}, {*the more A, the more B, the less C*} or {*the higher the speed, the higher the danger*}. These patterns are denoted as $\{A \geq B \geq \}$, $\{A \geq B \geq C \leq \}$ or $\{speed \geq danger \geq \}$ respectively. Such patterns hold if the variation constraints simultaneously hold on the attributes. However, it is often hardly possible to compare attribute values, either because the values are taken from noisy data, or because it is difficult to consider that a small difference between two values is meaningful. In this context, we focus on the use of fuzzy orderings to take this into account. *abstract* environment.

Keywords: Mining *gradual patterns*, fuzzy orderings, fuzzy gradual patterns.

1 Introduction

Given a database D an *association rule* is defined as a rule of the form *If A Then B* expressing the dependency between the so-called itemsets (binary attributes) A , B from the schema of D . The intended meaning of such a rule is that, if A is present in a transaction, then B is likely to be present too. An association rule is of the form:

$$R : I_{sa} \Rightarrow I_{sc}$$

where I_{sa} and I_{sc} are two itemsets. Two measures are usually defined to assess such rules: **The frequency/support** is the frequency of the union of the condition I_{sa} and consequence I_{sc} ie.

$$Freq(R) = Freq(I_{sa} \cup I_{sc})$$

The confidence measures the probability of knowing or occurrence of I_{sc} given I_{sa} , ie.

$$Conf(R) = \frac{Freq(I_{sa} \cup I_{sc})}{Freq(I_{sa})}$$

In the fuzzy case, the presence of an item in a transaction is a matter of degree. Another type of rule, called *gradual dependency*, conveys information in the form of attribute covariations, such as *the higher the age, the higher the salary*, meaning that the age of the persons increases together with its salary. Gradual dependencies consider tendencies across the whole data set, in terms of correlation of the attribute variations. This idea is closely connected to the so called *gradual rules* in fuzzy logic [9].

The automatic extraction of gradual dependencies or gradual association rules is one of the topics addressed in the field of data mining, for the modelling of frequent co-variations over a set of objects described by numerical attributes of data sets, such as biological databases, survey databases, data streams or sensor readings. In mining *gradual dependency* the idea is to express dependencies between the *direction of change* of attribute values.

As for the association rule extraction, the process consists of two steps: first frequent gradual patterns (also known as itemsets) are extracted. Then causality relations between the items are extracted. In mining frequent *gradual itemsets*, the goal is to discover frequent co-variations between attributes [10, 11].

When considering such gradual patterns and gradual rules, it is thus important to be able to count to which extent attributes co-variate. In this context, varied measures have been defined in the literature. However, few works have focused on how to exploit fuzzy orderings for handling noisy data.

For instance, when considering biological data from RNA/DNA chips, it would be semantically false to consider that two close values can be easily ordered. In this paper, we thus focus on an approach that evaluates frequent gradual patterns in terms of the robust rank correlation measure on the basis of fuzzy orderings.

The paper is organized as follows: in Section 2, we introduce the preliminary definitions and related work. The Section 3 is devoted to a review of fuzzy ordering-based rank correlation coefficient. In Section 4, we present our approach. Finally we present in Section 5 our conclusions and future research.

2 Preliminary Definitions and Related Work

In this section, after recalling the definitions of gradual item, gradual itemset, gradual dependencies, rank correlation, fuzzy rank correlation as given in [9,10,11], we present the related works on gradual pattern mining, rank correlation for extracting gradual itemsets, mining gradual dependencies based on fuzzy rank correlation, fuzzy ordering-based rank correlation coefficient, and on parallel frequent gradual pattern mining.

2.1 Preliminary Definitions

Gradual dependencies extraction applies to a data set D defined as a set of tuples T over a schema S of I attributes with m numerical values.

A *gradual item* is defined as a pair (I, θ) where I is an attribute in D and θ a comparison operator in $\{ \geq, \leq \}$. They represent the fact that the attribute values increase (in case of \geq) or decrease (in case of \leq).

A *gradual itemset* is defined as a combination of several gradual items, semantically interpreted as their conjunction $g = \{ (I_1, \theta_1), (I_2, \theta_2), \dots, (I_k, \theta_k) \}$ of cardinality greater than or equal to 2. For example, (Age, \geq) is a gradual item, while $\{(Age, \geq), (Salary, \leq), (Loans, \geq)\}$ is a gradual itemset, with a cardinality equal to 3.

The *support* of a gradual itemset in a data set D can be defined in varied manners [6], [11]. For instance, it can be defined as the number of tuples that can be ordered to support all item comparisons [11].

2.2 Related Works

Two kinds of dependencies can be distinguished: a first category considers linguistic variables represented by fuzzy sets and imposes covariation of the membership degrees across all data, for example, *the more the age is middle-aged, the less the number of cars is low*, where *middle-aged* and *low* refer to modalities of the linguistic variables *age* and *number of cars* respectively. A second, category directly considers the numerical values of the attributes and applies to attribute covariation on the whole attribute universe [11].

There are different interpretations of gradual dependency, as following: (1) *based in regression*, (2) *based in correlation*, (3) *approach based on conflict sets*, and (4) *approach based on the precedence grap*. Consult [11] for more information.

Laurent, Lesot, and Rifqi in [11] present an approach called GRAANK that combines the interpretation of gradual dependency of rank correlation measures and an algorithm on the precedence graph, named GRITE represented by its adjacency matrix, in a bitmap. The proposed algorithm thus follows the principle of the APRIORI algorithm, modifying the step of candidate evaluation, where for all candidate itemsets, compute their support as the sum of their binary matrices divided by $n(n - 1)/2$ where n is the number of objects.

Koh and Hullermeier in [9] present a framework for mining gradual dependencies based on the use of fuzzy rank correlation for measuring the strength of a dependency. The approach is a unification of previous approaches to evaluate gradual dependencies and captures both qualitative and quantitative measures of association as special cases. A gradual dependency $A \rightarrow B$ is evaluated in terms of two measures, namely the number of concordant pairs, CT , and the rank correlation $Fuzzy_\gamma$ as defined in (2). Comparing this approach with the classical setting of association analysis, CT plays the role of the support of a rule, while $Fuzzy_\gamma$ corresponds to the confidence. These measures can also be interpreted within the formal framework proposed by Dubois and Hullermeier in [7], in which every observation (in the case of a pair of points $(A(u), B(u))$ and $(A(v), B(v))$) is considered, to a certain degree, as an *example* of a pattern, as a *counterexample*, or as being *irrelevant* for the evaluation of the pattern. In the framework and the algorithm of Koh and Hullermeier, these degrees are given, respectively, by the degree of concordance, the degree of discordance, and the degree to which the pair is a tie. Formally they define the support and confidence of a gradual dependency $A \rightarrow B$ as follows:

$$supp(A \rightarrow B) = CT \tag{1}$$

$$conf(A \rightarrow B) = Fuzzy\gamma = \frac{CT - DT}{CT + DT} \tag{2}$$

where

$$CT = \sum_{u_i} \sum_{u_j} C(u_i, u_j) \tag{3}$$

$$CT = \sum_{u_i} \sum_{u_j} T(L(A(u_i), A(u_j)), L(B(u_i), B(u_j))) \tag{4}$$

$$DT = \sum_{u_i} \sum_{u_j} D(u_i, u_j) \tag{5}$$

$$DT = \sum_{u_i} \sum_{u_j} T(L(A(u_i), A(u_j)), L(B(u_j), B(u_i))) \tag{6}$$

Laurent et al. in [10] present an efficient parallel mining of gradual patterns and gradual rules on multicore processor based on the algorithm named GRITE (Gradual Itemset Extraction) and a model of parallelization multithreading type master-workers, where only parallelized the evaluation phase of frequent itemsets. In that framework, Laurent et al. consider the support of a gradual itemset P in a database DB as the ratio of the cardinality of P in DB denoted by $\lambda(P, DB)$ over the cardinality of DB denoted by $|DB|$. That is, $supp(P, DB) = \frac{\lambda(P, DB)}{|DB|}$.

Do et al. in [6] present PGLCM (Efficient Parallel Mining of Closed Frequent Gradual Itemsets) based on the parallelization of the GLCM algorithm based on the LCM algorithm (*Linear time Closed itemset Miner*) using the Melinda library. In this framework, Do et al. consider a gradual itemset $P = \{(i_{k_1}, v_{k_1}), \dots, (i_{k_j}, v_{k_j})\}$ where $\{k_1, \dots, k_j\} \subseteq \{1, \dots, n\}$ and the k_1, \dots, k_j are all distinct. Two tuples t and t' can be ordered with respect to P if all the values of the corresponding i_k items from the gradual itemset can be ordered with respect to variation $v \in \{\uparrow, \downarrow\}$ where \uparrow stands for a positive (ascending) variation, \downarrow for a negative (descending) variation and the formal definition of the support of P is $support(P) = \frac{max_{L \in I}(L)}{|R|}$, i.e. it is the size of the longest list of tuples that respects a gradual itemset P , where $L = \{t_1, \dots, t_m\}$ be a list of tuples from a set of tuples R defined over the schema $S = \{I_1, \dots, I_n\}$ of a dataset.

3 An Overview of Robust Rank Correlation Coefficients on the Basis of Fuzzy Orderings

3.1 Rank Correlation Measures: An Overview

Correlation measures are among the most basic tools in statistical data analysis and machine learning. They are applied to pairs of observations ($n \geq 2$) of two variables X and Y

$$(x_i, y_i)_{i=1}^n \tag{7}$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad (8)$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n) \quad (9)$$

of two linearly ordered domains \mathbf{X} and \mathbf{Y} to measure to which extent the two observations comply with a certain model. The most prominent representative is surely *Pearson's product moment coefficient*, often called *correlation coefficient* for short. *Pearson's product moment coefficient* is applicable to numerical data and assumes a linear relationship as the underlying model; therefore, it can be used to detect linear relationships, but no non-linear ones [4].

Rank correlation measures are intended to measure to which extent a monotonic function is able to model the inherent relationship between the two observables. They neither assume a specific parametric model nor specific distributions of the observables. They can be applied to ordinal data and, if some ordering relation is given, to numerical data too [4]. Therefore, rank correlation measures are ideally suited for detecting monotonic relationships, in particular, if more specific information about the data is not available [5], [9]. The two most common approaches are *Spearman's rank correlation coefficient* (short *Spearman's rho*) and *Kendall's tau* (*rank correlation coefficient*).

The goal of a rank correlation measure is to measure the dependence between the two variables in terms of their tendency to increase and decrease in the same or the opposite direction. If an increase in X tends to come along with an increase in Y , then the (rank) correlation is positive. The other way around, the correlation is negative if an increase in X tends to come along with a decrease in Y . If there is no dependency of either kind, the correlation is (close to) 0. Several rank correlation measures are defined in terms of the number C of *concordant*, the number D of *discordant*, and the number N of *tied* data points [9]. For a give index pair $(i, j) \in \{1, \dots, n\}^2$, we say that (i, j) is concordant, discordant or tied depending on whether $(x_i, x_j)(y_i, y_j)$ is positive, negative or 0, respectively. A well-known example is Goodman and Kruskal's *gamma rank correlation*, which is defined as:

$$\gamma = \frac{C - D}{C + D} \quad (10)$$

3.2 Fuzzy Orderings

Fuzzy relation, fuzzy equivalence relation, and fuzzy ordering are concepts that have been introduced with the aim; to model human-like decisions by taking the graduality of human thinking and reasoning into account. Fuzzy orderings have broad utility. They can be applied, for example, when expressing our preferences with a set of alternatives. Compared to crisp orderings, they have greater expressive power. They allow us to express not only that we prefer an alternative to another one, but also the strength of this preference [8]. The study of *similarity*, *fuzzy relation*, *fuzzy ordering*, *similarity relation*, and the notion of *equivalence* was started by Zadeh [12] in 1971, in that paper he defined the notion of *similarity* as a generalization of the notion of equivalence, and a *fuzzy ordering* as a generalization of the concept of ordering.

A fuzzy relation $S : X^2 \rightarrow [0,1]$ is called *similarity relation* on a domain X with respect to a t-norm T , for brevity *T-similarity*, if and only if the following three axioms hold for all $x, y, z \in X$:

- (i) *S*-reflexivity: $\mu_S(x,x)=1$,
- (ii) *S*-symmetry: $\mu_S(x,y)=\mu_S(y,x)$, and
- (iii) *T*-transitivity: $\mu_T(\mu_S(x,y), \mu_S(y,z)) \leq \mu_S(x,z)$.

Where $\mu_S(x,y)$, $\mu_S(y,z)$ and $\mu_S(x,z)$ are the grade of membership of the ordered pairs (x,y) , (y,z) , and (x,z) in S , with respect to a triangular norm (*t-norm*) T .

A fuzzy relation $E : X^2 \rightarrow [0,1]$ is called *fuzzy equivalence relation* on a domain X with respect to a t-norm T , for brevity *T-equivalence*, if and only if the following three axioms are fulfilled for all $x, y, z \in X$:

- (i) *E*-reflexivity: $\mu_E(x,x)=1$,
- (ii) *E*-symmetry: $\mu_E(x,y)=\mu_E(y,x)$, and
- (iii) *T*-transitivity: $\mu_T(\mu_E(x,y), \mu_E(y,z)) \leq \mu_E(x,z)$.

Where $\mu_E(x,y)$, $\mu_E(y,z)$ and $\mu_E(x,z)$ are the grade of membership of the ordered pairs (x,y) , (y,z) , and (x,z) in E , with respect to a triangular norm (*t-norm*) T .

The concept of fuzzy order was introduced by generalizing the notion of (i) reflexivity $\mu_R(x,x)$ for any $x \in X$, (ii) antisymmetry ($\mu_R(x,y)$ and $\mu_R(y,x)$) imply $x = y$, and (iii) transitivity ($\mu_R(x,y)$ and $\mu_R(y,z)$) imply $\mu_R(x,z)$), where R is a fuzzy relation called an *order relation* in X if it satisfies (i), (ii), and (iii). A set X in which an order relation has been given is called an *ordered set* (*semi-ordered set* or *partially ordered set*), i.e. a *fuzzy ordering* is a fuzzy relation which is transitive. A *fuzzy partial ordering*, P , is a fuzzy ordering which is reflexive and antisymmetric ($\mu_P(x, y) > 0$ and $x \neq y$) imply $\mu_P(y, x) = 0$. A *fuzzy linear ordering* is a fuzzy partial ordering in which $x \neq y$ imply $\mu_S(x, y) > 0$ or $\mu_S(y, x) > 0$. A *fuzzy preordering* is a fuzzy ordering which is reflexive. A *fuzzy weak ordering* is a fuzzy preordering in which $x \neq y$ imply $\mu_S(x, y) > 0$ or $\mu_S(y, x) > 0$.

In the last decade Ulrich Bodenhofer [1], [2] and [3] has presented a general framework for comparing fuzzy sets with respect to a general class of fuzzy orderings. This approach includes known techniques based on generalizing the crisp linear ordering of real numbers by means of the extension principle, applicable to any fuzzy subsets of any kind of universe for which a fuzzy ordering is known—no matter whether linear or partial. A approach for fuzzification of the ordering relation and ways to compare fuzzy sets with different heights, and ways of how to refine the ordering relation by lexicographic hybridization with a different ordering method. A formal study of fuzzy orderings with applications to statistical analysis of numerical data, has been made by Bodenhofer and Klawonn [4], [5].

A fuzzy relation $L : X^2 \rightarrow [0,1]$ is called *fuzzy ordering* with respect to a t-norm T and a *T-equivalence* $E : X^2 \rightarrow [0,1]$, for brevity *T-E-ordering*, if and only if the following three axioms are fulfilled for all $x, y, z \in X$:

- (i) *E*-Reflexivity: $\mu_E(x,y) \leq \mu_L(x,y)$
- (ii) *T-E*-Antisymmetry: $\mu_T(\mu_L(x,y), \mu_L(y,x)) \leq \mu_E(x,y)$
- (ii) *T*-transitivity: $\mu_T(\mu_L(x,y), \mu_L(y,z)) \leq \mu_L(x,z)$.

Where $T - E$ -ordering L is strongly complete if $\mu_T(\mu_L(x,y), \mu_L(y,x)) = 1$ for all $x,y \in X$, $\mu_{E_r}(x,y) = \max(0, 1 - \frac{1}{r} * |x - y|)$ is a μ_{T_L} -Equivalence on \mathbf{R} (assume $r > 0$), and $\mu_{T_L}(x,y)$ denoted the Lukasiewicz t -norm.

$$\mu_{T_L}(x,y) = \max(0, x + y - 1) \tag{11}$$

For all $x, y \in X$, and based on the definition of strongly complete fuzzy orderings [4] and [5],

$$\mu_{L_r}(x,y) = \min(1, \max(0, 1 - \frac{1}{r} * (x - y))) \tag{12}$$

is a strongly complete $T_L - E_r$ -ordering on \mathbf{R} . In order to generalize the notion of concordant and discordant pair, a binary fuzzy relation $R : X^2 \rightarrow [0,1]$ is called a *strict fuzzy ordering* with respect to a t -norm T and a T -equivalence E , for brevity *strict T - E -ordering*, if it is irreflexive $\mu_R(x,x) = 0$ for all $x \in X$, T -transitive, and E -extensional $\mu_T(\mu_E(x,x'), \mu_E(y,y'), \mu_R(x,y)) \leq \mu_R(x',y')$, for all $x, y, z \in X$. Given a $T_L - E$ -ordering L strongly complete, it can be proven that the fuzzy relation R_x is defined as:

$$\mu_{R_x}(x_1, x_2) = 1 - \mu_{L_x}(x_2, x_1) \tag{13}$$

Analogously for all $y \in Y$ R_y is defined as:

$$\mu_{R_y}(y_1, y_2) = 1 - \mu_{L_y}(y_2, y_1) \tag{14}$$

3.3 A Fuzzy Ordering-Based Rank Correlation Coefficient

Bodenhofer and Klawonn in [4] and [5] demonstrate that established rank correlation measure are not ideally suited for measuring rank correlation for numerical data that are perturbed by noise, they propose to use robust rank correlation measures based on fuzzy orderings named Fuzzy Rank Correlation and demonstrate that the new measures overcome the robustness problems of existing rank correlation coefficients. The formal description is: Assume that the data are given as in (7), (domain $_x$), and (domain $_y$), where $x_i \in X$ and $y_i \in Y$ for all $i=1, \dots, n$, this means that we have two T_L -equivalences $E_x : X^2 \rightarrow [0,1]$ and $E_y : Y^2 \rightarrow [0,1]$, a strongly complete $T_L - E_x$ -ordering $L_x : X^2 \rightarrow [0,1]$ with a strict $T_L - E_x$ -ordering on X define as in (13) and a strongly complete $T_L - E_y$ -ordering $L_y : Y^2 \rightarrow [0,1]$ with a strict $T_L - E_y$ -ordering on Y define as in (14).

According to the gamma rank correlations measure and given an index pair (i, j) where $i = (x_i, y_i)$ and $j = (x_j, y_j)$, we can compute the degree to which (i, j) is a concordant pair as:

$$C(i, j) = \mu_{T_L}(\mu_{R_x}(x_i, x_j), \mu_{R_y}(y_i, y_j)) \tag{15}$$

And the degree to which (i, j) is a discordant pair as

$$D(i, j) = \mu_{T_L}(\mu_{R_x}(x_i, x_j), \mu_{R_y}(y_j, y_i)) \tag{16}$$

The numbers of concordant pairs CT and discordant pair DT , respectively, as:

$$CT = \sum_{i=1}^n \sum_{j \neq i} C(i, j) \tag{17}$$

$$DT = \sum_{i=1}^n \sum_{j \neq i} D(i, j) \tag{18}$$

So the *fuzzy ordering-based rank correlation measure* γ can be computed as:

$$Fuzzy\gamma = \frac{CT - DT}{CT + DT} \tag{19}$$

Where $\mu_{T_L}(x, y)$, $\mu_{R_x}(x_1, x_2)$, $\mu_{R_y}(y_1, y_2)$, $\mu_{L_x}(x_2, x_1)$ and $\mu_{L_y}(y_2, y_1)$ by fuzzy orderings we can compute as in (11), (13), (14), and (12) respectively.

4 Fuzzy Ordering-Based Rank Correlation Coefficient for Mining of Gradual Itemsets

4.1 Notations

The automatic extraction of gradual dependencies consists of two steps: 1. extraction of frequent gradual itemsets, and 2. extraction of causality relations between the items. In this work, we focus on the first step, and we consider the following notations: A data set \mathcal{D}_S , constituted of \mathcal{N} objects or transactions (data record) denote by $\mathcal{T} = \{t_1, \dots, t_N\}$ described by \mathcal{M} numerical attributes denote by $\mathcal{A} = \{A_1, \dots, A_M\}$. Table of Fig. 1 shows an example data set where $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$ transactions and $\mathcal{A} = \{A_1 : age, A_2 : salary, A_3 : loans, A_4 : cars\}$ attributes, its graphic illustration is shown in the diagram and graphics of Fig. 1.

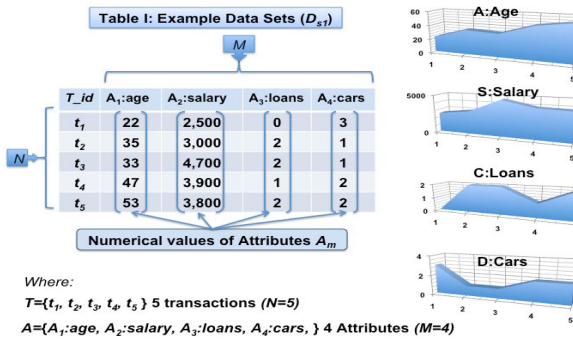


Fig. 1. Notations of a Data Set

In this framework, let us consider a *gradual itemset* $I_S ::= \{I_l\}^k$ where $\{I_l\}^k ::= I_1 \dots I_k$, such that $I_1 \neq I_2 \neq \dots \neq I_k$, for $k ::= 2 \mid 3 \mid \dots \mid \mathcal{M}$, each *gradual item* $I_l ::= \mathcal{A}v$, where $\mathcal{A} ::= A_1 \mid A_2 \mid \dots \mid A_{\mathcal{M}}$, each $A_m ::= id_attribut [vector\ of\ numeric\ values\ u_i]$ for $i=1, 2, \dots, \mathcal{N}$, and $v ::= \geq \mid \leq$, represent a positive (ascending) variation in the numeric values of the attribute A_m (in case $v ::= \geq$) or a negative (descending) variation (in case $v ::= \leq$), see Fig. 2 a). For instance $I_S ::= \{ A_1 \geq A_2 \geq A_4 \leq \}$ is interpreted as a *gradual itemset* of size $k = 3$ illustrated in Fig. 2 b), where for case of the data set of table in Fig. 1 it imposes an ascending variation on the values of attributes $age(u_i, u_j)$ and $salary(u_i, u_j)$ and a descending variation on the values of attribute $cars(u_i, u_j)$ and are concordant pairs simultaneously.

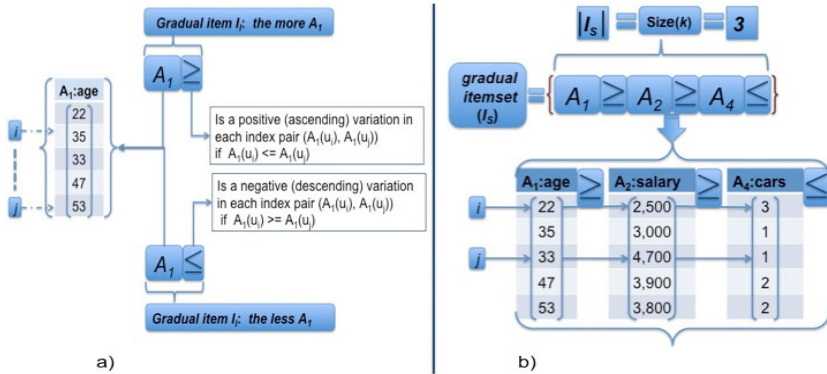


Fig. 2. Illustration: a) Variations of a gradual item, b) A gradual itemset of size $k = 3$

4.2 Algorithm of Extraction of Frequent Gradual Itemsets

In this context we propose an algorithm that evaluates gradual dependencies in terms of a fuzzy rank correlation coefficient, as described in the algorithm 1, where we apply the APRIORI algorithm to generate candidates from the k -itemsets to take advantage of the fact that any subset of a frequent itemset is also a frequent itemset and all infrequent itemsets can be pruned if it has an infrequent subset. We implemented the Fuzzy Ordering-Based Rank Correlation Coefficient ($Fuzzy_{\gamma}$) according to the formal description presented in the previous section, this in order to evaluate candidates itemsets and mining frequent gradual itemset.

4.3 Properties of the Proposed Method

For us, in this work, the problem to address is the automatic extraction of frequent gradual itemsets, in which, relations between the directions of changing the values of the attributes involved are non-linear and/or affected by noise. Consequently, we propose a method of automatic extraction of frequent gradual

Algorithm 1. Fuzzy Ordering-Based Rank Correlation Coefficient

Data: Data set (\mathcal{D}_S), Size(k) and Minimum support (ε)

Result: Frequent Gradual Itemset (\mathcal{I}_F), Support(CT) and Confidence $\mathcal{Fuzzy}\text{-}\gamma$

MainBegin

for $\mathcal{K}=2, 3, 4, \dots, \mathcal{M}$ **do**

- ┌ $\mathcal{I}_s \leftarrow \text{Gen_Cand_Gradual_Itemsets}(\text{Size } \mathcal{K});$
- ┌ */* Compute of rank correlation measure, in:*/*
- ┌ $\text{Fuzzy}\text{-}\gamma\text{-Evaluation}(\text{Itemsets } \mathcal{I}_s, \text{Size } \mathcal{K});$
- ┌ $k++;$

End;

$\text{Fuzzy}\text{-}\gamma\text{-Evaluation}(\text{Itemsets } \mathcal{I}_s, \text{Size } \mathcal{K})$

Begin

$CT \leftarrow \emptyset;$ */* Concordant and Support */*

$DT \leftarrow \emptyset;$ */* Discordant */*

foreach Candidate Gradual Itemset $I_s \in \mathcal{I}_s \times \{ \geq, \leq \}$, Size k **do**

- ┌ */* Compute of concordant \mathcal{C} and discordant \mathcal{D} pair (u_i, u_j) */*
- ┌ **for** $i=0, 1, 2, \dots, \mathcal{N}-1$ **do**
- ┌ ┌ **foreach** $j \in \{ 0, \dots, \mathcal{N}-1 \}$ and $j \neq i$ **do**
- ┌ ┌ ┌ */* Compute: Relationships \mathcal{R}_I of each Item $I \{A \geq \mid A \leq\} \in I_s$ */*
- ┌ ┌ ┌ **for** $ri=0, 1, 2, \dots, k-1$ **do**
- ┌ ┌ ┌ ┌ **if** variation is " \geq " **then**
- ┌ ┌ ┌ ┌ ┌ $\mathcal{R}_C[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_i), I.A_{ri}(u_j));$
- ┌ ┌ ┌ ┌ ┌ $\mathcal{R}_d[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_j), I.A_{ri}(u_i));$
- ┌ ┌ ┌ ┌ **if** variation is " \leq " **then**
- ┌ ┌ ┌ ┌ ┌ $\mathcal{R}_C[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_j), I.A_{ri}(u_i));$
- ┌ ┌ ┌ ┌ ┌ $\mathcal{R}_d[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_i), I.A_{ri}(u_j));$
- ┌ ┌ ┌ ┌ */* Compute: to each index pair $\in \mathcal{R}_I$ is concordant \mathcal{C} */*
- ┌ ┌ ┌ $\mathcal{C}(i, j) \leftarrow \min(\mathcal{R}_C[0], \mathcal{R}_C[1], \dots, \mathcal{R}_C[k-1]);$
- ┌ ┌ $CT \leftarrow CT + \mathcal{C}(i, j);$
- ┌ ┌ */* Compute: to each index pair $\in \mathcal{R}_I$ is discordant \mathcal{D} */*
- ┌ ┌ $\mathcal{D}(i, j) \leftarrow \min(\mathcal{R}_d[0], \mathcal{R}_d[1], \dots, \mathcal{R}_d[k-1]);$
- ┌ ┌ $DT \leftarrow DT + \mathcal{D}(i, j);$
- ┌ ┌ *Support* $\leftarrow CT / (n * (n - 1) / 2);$
- ┌ ┌ **if** *Support* $\geq \text{minSupp}(\varepsilon)$ **then**
- ┌ ┌ ┌ $\mathcal{I}_F \leftarrow \mathcal{I}_F \cup \{I_s\};$
- ┌ ┌ */* Compute the Fuzzy Ordering-Based Rank Correlation Coefficient */*
- ┌ $\mathcal{Fuzzy}\text{-}\gamma \leftarrow (CT - DT) / (CT + DT);$

End;

Table 1. Examples of lists of concordant couples of Gradual Itemsets

Itemset	List of concordant couples	<i>CT</i>	<i>DT</i>	<i>Support</i>	<i>Fuzzy-γ</i>
$A_1 \geq A_2 \geq$	$\{(0,1)(0,2)(0,3)(0,4)(1,3)(1,4)\}$	6	4	6/10	0.2
$A_1 \geq A_2 \geq A_3 \leq$	$\{(0,1)(0,2)(0,3)(0,4)\}$	4	5	4/10	-0.111
$A_1 \geq A_2 \geq A_3 \geq A_4 \leq$	$\{(0,1)(0,2)(0,3)(0,4)\}$	4	6	4/10	-0.2
$A_1 \geq A_3 \leq$	$\{(0,1)(0,2)(0,3)(0,4)\}$	4	5	4/10	-0.111
$A_2 \geq A_3 \leq$	$\{(0,1)(0,2)(0,3)(0,4)\}$	4	4	4/10	0.00

itemsets on the basis of fuzzy orderings. To illustrate this, we consider the data set described in table and graphs of Fig. 1. Table 1 contains the list de concordant couplas, the numbers of concordant pairs *CT* and discordant pair *DT*, the support, and the fuzzy rank correlation coefficient (*Fuzzy-γ*), for several gradual itemsets.

Properties of the proposed method and algorithms are: (i) In order to compute the degree to which each index pair $\mathcal{C}(i, j) \leftarrow \min(\mathcal{R}_C[0], \mathcal{R}_C[1], \dots, \mathcal{R}_C[k - 1])$ are concordant pairs in itemsets $|I_s| > 2$, we exploit the properties of associativity and commutativity of t-norm of (15), (ii) In order to compute the degree to which each index pair $\mathcal{D}(i, j) \leftarrow \min(\mathcal{R}_d[0], \mathcal{R}_d[1], \dots, \mathcal{R}_d[k - 1])$ are discordant pairs in itemsets $|I_s| > 2$, we exploit the properties of associativity and commutativity of t-norm of (16), (iii) Each $\mathcal{R}_C[ri]$, for the concordant case, is computed as $\mathcal{R}_C[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_i), I.A_{ri}(u_j))$, and for the discordant case as $\mathcal{R}_d[ri] \leftarrow \mathcal{R}_I(I.A_{ri}(u_j), I.A_{ri}(u_i))$, (iv)The concordance degrees $\mathcal{C}(i, j)$ are stored in an $|N| \times |N|$ matrix, from which the total number of concordant pairs *CT* of an itemset I_s is computed by summing all entries, and (v) Finally, the support of itemset I_s is computed as: $Support(I_s) = CT / (n * (n - 1) / 2)$, and the set of frequent gradual itemsets \mathcal{I}_F is updated as $\mathcal{I}_F \leftarrow \mathcal{I}_F \cup \{I_s\}$ if $Support(I_s) \geq minSupp(\epsilon)$.

5 Conclusions and Remarks

In this paper, we have presented a review of the basis and new models of fuzzy orderings, also we propose an original approach for extracting gradual itemsets. In our approach apply the APRIORI algorithm to generate candidates from the k -itemsets to take advantage of the fact that any subset of a frequent itemset is also a frequent itemset and all infrequent itemsets can be pruned if it has an infrequent subset, in order to evaluate candidates itemsets and mining frequent gradual itemset we implemented the Fuzzy Ordering-Based Rank Correlation Coefficient (*Fuzzy-γ*) according to the formal description of Bodenhofer and Klawonn [4], [5] and Zadeh [12].

An important aspect to be addressed in future work includes the study of other optimizations in order to improve the efficiency of our approach (for example, the parallelization of our algorithm). Thus, in order to guarantee scalability,

efficient pruning techniques are needed to avoid unnecessary comparisons. We will also study how causality can be defined based on this work, and efficiently extracted.

References

1. Bodenhofer, U.: Orderings of Fuzzy Sets Based on Fuzzy Orderings Part I: The Basic Approach. *Mathware & Soft Computing* 15, 201–218 (2008)
2. Bodenhofer, U.: Orderings of Fuzzy Sets Based on Fuzzy Orderings Part II: Generalizations. *Mathware & Soft Computing* 15, 219–249 (2008)
3. Bodenhofer, U.: Fuzzy Orderings of Fuzzy Sets. In: Proc. 10th IFSA World Congress, Istanbul, pp. 500–507 (July 2003), <http://www.bioinf.jku.at/people/bodenhofer/private/publications/Conferences.html>
4. Bodenhofer, U., Klawonn, F.: Towards Robust Rank Correlation Measures for Numerical Observations on the Basis of Fuzzy Orderings. In: 5th Conference of the European Society for Fuzzy Logic and Technology, pp. 321–327. University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, Ostrava, Czech Republic (2007)
5. Bodenhofer, U., Klawonn, F.: Robust Rank Correlation Coefficients on the Basis of Fuzzy Orderings: Initial Steps. *Mathware & Soft Computing* 15, 5–20 (2008)
6. Do, T.D.T., Laurent, A., Termier, A.: PGLMC: Efficient Parallel Mining of Closed Frequent Gradual Itemsets. In: Proc. International Conference on Data Mining (ICDM), Sydney, Australia, pp. 138–147. IEEE Computer Society Press, Los Alamitos (2010)
7. Dubois, D., Hullermeier, E., Prade, H.: A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery* 13(2), 167–192 (2006)
8. Ismat, B., Samina, A.: Numerical Representation of Product Transitive Complete Fuzzy Orderings. *Journal Elsevier Mathematical and Computer Modelling* 53, 617–623 (2011)
9. Koh, H.-W., Hüllermeier, E.: Mining Gradual Dependencies Based on Fuzzy Rank Correlation. In: Borgelt, C., González-Rodríguez, G., Trutschnig, W., Lubiano, M.A., Gil, M., Grzegorzewski, P., Hryniewicz, O. (eds.) *Combining Soft Computing and Statistical Methods in Data Analysis*. AISC, vol. 77, pp. 379–386. Springer, Heidelberg (2010)
10. Laurent, A., Négrevergne, B., Sicard, N., Termier, A.: Efficient Parallel Mining of Gradual Patterns on Multicore Processors. In: *AKDM2, Advances in Knowledge Discovery and Management*. Springer, Heidelberg (2010)
11. Laurent, A., Lesot, M.-J., Rifqi, M.: GRAANK: Exploiting rank correlations for extracting gradual itemsets. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) *FQAS 2009. LNCS(LNAI)*, vol. 5822, pp. 382–393. Springer, Heidelberg (2009)
12. Zadeh, L.A.: Similarity relations and fuzzy orderings. *Information Sciences* 3(2), 177–200 (1971)

Spearman's Rank Correlation Coefficient for Vague Preferences

Przemysław Grzegorzewski^{1,2} and Paulina Ziemińska²

¹ Systems Research Institute, Polish Academy of Sciences,
Newelska 6, 01-447 Warsaw, Poland

² Faculty of Math. and Information Sci., Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland

Abstract. The problem of measuring association between preference systems in situations with missing information or noncomparable outputs is discussed. A new generalization of Spearman's Rho is suggested. Moreover, it is shown how to apply the suggested coefficient for testing independence.

Keywords: Information retrieval, recommender systems, association measures, Spearman's rank correlation coefficient, independence, preferences, orderings, ranks, IF- sets.

1 Introduction

Many various tasks in broadly meant information retrieval require a comparison of two orderings. Such orderings may represent preferences of particular users with respect to the data sought, the relevance assessment produced by retrieval systems etc.

The former may be exemplified by the recommender systems [10]. In collaborative filtering a recommender system suggests to a user available resources using similarity of his or her preferences to those of the other users. These preferences may be represented by an ordering/ranking set by a user of the resources he or she have seen earlier. If an ordering of given user A is similar to (correlated with) an ordering of another user B then the recommender systems tends to suggest to user A another resources highly preferred by B and not yet seen by A.

The latter use of orderings may be exemplified by the task of retrieval evaluation. For example, rankings of documents produced by several search engines with respect to a query may be evaluated with respect to a reference ranking [1]. The more similar ranking of a given search engine the higher its quality is evaluated.

There are many methods to compare rankings. Among them the statistical approaches play the most important role. Spearman's Rho or Kendall's Tau are the most popular rank correlation measures. These two coefficients are also usually applied in data mining (e.g. a study of associations which may occur between attributes in transactions database). Using rank correlation coefficient

we may state whether variables under study are concordant or discordant and evaluate the strength of the possible association.

However, in practical cases the application of classical measures encounters some difficulties related to incompleteness and non-linearity of orderings. For example, the sets of resources evaluated by particular users of a recommender systems usually differ essentially, and thus the rankings they provide are incomplete with respect to each other. At the same time, many resources may be identically ranked by a user. The same happens in case of the answers of search engines: they may differ due to the fact that the sets of documents indexed by each of them usually differ as well as due to difficulty of taking into account the whole output produced by a search engine with respect to a query.

Of course, one may suggest to remove all "problematic" data. However, in this way we loose to much information and the conclusions of the inference might be misleading. Thus our goal is to suggest methods which could be applied even if the requirements of the classical statistical tools are not completely satisfied, like for preferences which do not form a linear order but only a partial preorder.

The problem of the partial preorders comparison was considered, e.g., by Roy and Slowinski [11]. Hébert et al. [9] extended Kendall's rank correlation to interval and fuzzy data. Kendall's correlation coefficient and other rank-based nonparametric procedures for fuzzy data was also considered by Dencœur et al. [4]. On the other hand, the generalized version of Kendall's Tau for vague preferences was suggested by Grzegorzewski [8] who has also proposed the generalized version of the so-called Kendall's coefficient of concordance [7].

In this paper we propose a novel approach to measuring similarity (correlation) of two rankings. We discuss it in a purely theoretical context but the proposed approach is immediately applicable for various tasks exemplified above. More precisely, we show how to generalize Spearman's rank correlation coefficient into situations with missing information or noncomparable outputs. In our approach we utilize IF-sets which seem to be a very convenient tool for modelling preferences under incompleteness and non-linearity of orderings.

The paper is organized as follows: in Section 2 we recall the definition and basic properties of Spearman's rank correlation coefficient. Next, in Section 3 we mention briefly some information on IF-sets and describe how to apply IF-sets for modelling preferences. Then in Section 4 we suggest a definition of the generalized Spearman rank correlation coefficient and discuss some of his properties. In Sec. 5 we consider some remarks on the distribution of the suggested coefficient and then in Sec. 6 we discuss how to apply it for testing independence. Finally we illustrate the proposed notions in a suitable example.

2 Spearman's Rank Correlation Coefficient

Suppose our data are pairs of n observations on two variables A and B , i.e. $(A_1, B_1), \dots, (A_n, B_n)$. To compute the Spearman correlation coefficient we first rank all the A observations within themselves from smallest to largest (or from largest to smallest). Then we independently rank the values of the second variable

B using the same ranking scheme. In other words, each observation is assigned a rank according to its position relative to the others in its own group. When no ties exist (i.e. when there are no two values of A or two values of B with the same rank) then the Spearman correlation coefficient is defined by the following formula

$$r_s = 1 - \frac{6}{n(n^2 - 1)} \sum_{i=1}^n d_i^2, \tag{1}$$

where $d_i, i = 1, \dots, n$, are the differences in ranks of A_i and B_i (see, e.g. [5]).

Spearman’s correlation coefficient satisfies the usual requirements of good association measure. It can be shown that

$$-1 \leq r_s \leq 1. \tag{2}$$

If variables A and B are perfectly concordant then $r_s = 1$, while for perfect discordance (arrangement B must be the reverse of arrangement A) we get $r_s = -1$. It is equal to zero when there is no relation between A and B . Values between these extremes give a relative indication of the degree of association between A and B . Moreover, r_s is commutative and symmetric about zero. Spearman’s correlation coefficient is also invariant under all order-preserving transformations.

If tied observations also appear then the most common practice for dealing with them, as in most other nonparametric procedures, is to assign equal ranks to indistinguishable observations.

A typical area of application of the Spearman rank correlation coefficient is the comparison of preferences. In such a problem we assume that n elements are ordered according to preferences of the two persons (or search engines) A and B and basing on these two rankings Spearman’s coefficient is calculated to indicate that preferences are concurrent or opposite or to check whether there is any relationship between them. Further on we will use this interpretation of the correlation coefficient. Before we suggest the generalization of the Spearman correlation coefficient let us consider a following example:

Example 1

Consider a typical situation of a recommender system and two users A and B who set their preferences in relation to seven TV channels (say, x_1, \dots, x_7). Suppose, their opinions are as follows: A prefers x_3 , then x_1 and then x_4 ; the next one is both x_2 and x_5 which A apprizes evenly. The last one is x_7 but he has no opinion on x_6 .

The next user B has chosen x_1 as the best one, then he has mentioned x_3 and x_6 both more or less equivalent and next x_2, x_7 and x_5 . He had no opinion on x_4 .

Table 1 presents ranks given by our two users to the seven channels (vectors R^A and R^B), where smaller number means more preferred channel and “?” means that analysts does not have any opinion about this channel.

Table 1. Example: users' preferences

channel	x_1	x_2	x_3	x_4	x_5	x_6	x_7
R^A	2	4	1	3	4	?	5
R^B	1	3	2	?	5	2	4

It is obvious that this kind of data require rank correlation measure to evaluate the association between preferences. Unfortunately, well-known Spearman's coefficient cannot be applied there directly because not all elements have been ranked.

Below we suggest how to cope with such problems. Since our approach utilizes IF-sets we start from recalling some basic concepts and notation.

3 Modelling Preferences

Let X denote a universe of discourse. Then a fuzzy set C in X is defined as a set of ordered pairs

$$C = \{ \langle x, \mu_C(x) \rangle : x \in X \}, \quad (3)$$

where $\mu_C : X \rightarrow [0, 1]$ is the membership function of C and $\mu_C(x)$ is the grade of belongingness of x into C (see [12]). Thus automatically the grade of nonbelongingness of x into C is equal to $1 - \mu_C(x)$. However, in real life the linguistic negation not always identifies with logical negation. This situation is very common in natural language processing, computing with words, etc. Therefore Atanassov [2], [3] suggested a generalization of classical fuzzy set, called an intuitionistic fuzzy set (the name suggested by Atanassov is slightly misleading, because his sets have nothing in common with intuitionism known from logic and hence, in order to avoid terminology problems, we call the Atanassov sets as IF-sets).

An IF-set C in X is given by a set of ordered triples

$$C = \{ \langle x, \mu_C(x), \nu_C(x) \rangle : x \in X \}, \quad (4)$$

where $\mu_C, \nu_C : X \rightarrow [0, 1]$ are functions such that

$$0 \leq \mu_C(x) + \nu_C(x) \leq 1 \quad \forall x \in X. \quad (5)$$

For each x the numbers $\mu_C(x)$ and $\nu_C(x)$ represent the degree of membership and degree of nonmembership of the element $x \in X$ to $C \subset X$, respectively. It is easily seen that an IF-set $\{ \langle x, \mu_C(x), 1 - \mu_C(x) \rangle : x \in X \}$ is equivalent to (3), i.e. each fuzzy set is a particular case of the IF-set. We will denote a family of fuzzy sets in X by $FS(X)$, while $IFS(X)$ stands for the family of all IF-sets in X .

For each element $x \in X$ we can compute, so called, the IF-index of x in C defined as follows

$$\pi_C(x) = 1 - \mu_C(x) - \nu_C(x). \quad (6)$$

It is seen immediately that $\pi_C(x) \in [0, 1] \forall x \in X$. If $C \in FS(X)$ then $\pi_C(x) = 0 \forall x \in X$.

As many contributions show IF-sets form a natural and useful tool for modelling preference systems admitting nonlinear orderings. In particular, Grzegorzewski [6], [7], [8] proposed the method of preference system construction for situations when not all elements under consideration can be ranked according to preference systems A and B . The main point of his idea is that we attribute an IF-set to each preference system. For simplicity of notation we will identify preference systems A and B with the corresponding IF-sets.

Thus let $A = \{ \langle x_i, \mu_A(x_i), \nu_A(x_i) \rangle : x_i \in X \}$ denote an IF-subset of the universe of discourse $X = \{x_1, \dots, x_n\}$, where membership function $\mu_A(x_i)$ indicates the degree to which x_i is the most preferred element according to the preference system A , while nonmembership function $\nu_A(x_i)$ shows the degree to which x_i is the less preferred element according to A . Similarly, let $B = \{ \langle x_i, \mu_B(x_i), \nu_B(x_i) \rangle : x_i \in X \}$ denote an IF-subset of the universe of discourse X , where membership function $\mu_B(x_i)$ and nonmembership function $\nu_B(x_i)$ indicate the degree to which x_i is the most preferred and the less preferred element according to the preference system B , respectively.

To determine all these membership and nonmembership functions let us recall that the only available information on A and B are orderings that admit ties and elements that cannot be ranked. However, one can always specify two functions $w_A, b_A : X \rightarrow \{0, 1, \dots, n-1\}$ defined as follows: for each given $x_i \in X$ let $w_A(x_i)$ denote the number of elements $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ surely worse than x_i , while $b_A(x_i)$ let be equal to the number of elements $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ surely better than x_i in the ordering corresponding to the preference system A . Using functions $w_A(x_i)$ and $b_A(x_i)$ we may determine the requested membership and nonmembership functions μ_A and ν_A . Namely, let

$$\mu_A(x_i) = \frac{w_A(x_i)}{n-1}, \tag{7}$$

$$\nu_A(x_i) = \frac{b_A(x_i)}{n-1}. \tag{8}$$

Similarly, we may easily find two functions $w_B, b_B : X \rightarrow \{0, 1, \dots, n-1\}$ such that for each given $x_i \in X$ a value $w_B(x_i)$ denotes the number of elements $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ surely worse than x_i , while $b_B(x_i)$ is equal to the number of elements $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ surely better than x_i in the ordering corresponding to the preference system B . Thus, as before, the membership and nonmembership functions μ_B and ν_B are given as follows

$$\mu_B(x_i) = \frac{w_B(x_i)}{n-1}, \tag{9}$$

$$\nu_B(x_i) = \frac{b_B(x_i)}{n-1}. \tag{10}$$

It is easily seen that $w_A(x_i), b_A(x_i), w_B(x_i), b_B(x_i) \in \{0, \dots, n-1\}$ because we rank n elements and hence for each element $x_i \in X$ there exist no less than zero and no more than $n-1$ elements which are better (worse) than x_i . Moreover, we admit situations when the same rank is assigned to more than one element (so-called, ties) and elements that are not comparable with the others.

This way we get two well defined IF-sets describing conveniently the preference systems A and B . Without loss of generality let us discuss the properties of A . It is seen that the IF-index $\pi_A(x_i) = 0$ for each $x_i \in X$ if and only if all elements are ranked and there are no ties. Conversely, if there is element $x_i \in X$ that $\pi_A(x_i) > 0$ then there are ties or noncomparable elements in the corresponding preference system. Moreover, more ties or more elements not comparable with the others means higher values of the IF-index are observed. One may also notice that $\pi_A(x_i) = 1$ if and only if element $x_i \in X$ is noncomparable with other element or all elements x_1, \dots, x_n have obtained the same rank.

4 Generalized Spearman’s Coefficient

Using this methodology for modelling preference systems with missing information or noncomparable outputs Grzegorzewski suggested how to generalize Kendall’s correlation coefficient and Kendall’s coefficient of concordance (see [8] and [7], respectively). He also proposed the generalization of the Spearman rank correlation coefficient [6] but that construction was too complicated and not eligible for constructing statistical tests. Below we show another generalization of Spearman’s rank correlation coefficient which, contrary to the one known from the literature, is not so complicated and is useful in dependence test constructing.

Due to Spearman’s rank correlation coefficient representation:

$$r_s(R, S) = 1 - \frac{6 \sum_{i=1}^n D_i^2}{n(n^2 - 1)} = 1 - \xi_n \cdot \partial(S, R), \tag{11}$$

and preferences modelling utilizing IF-sets we may use the following distance between two systems of preferences B and C :

$$d(B, C) = \sum_{j=1}^n [(\mu_B(x_j) - \mu_C(x_j))^2 + (\nu_B(x_j) - \nu_C(x_j))^2]. \tag{12}$$

Using this distance measure, as for the generalized coefficient of concordance, comes also from the classical well known relationship between those two measures.

Definition 1. Let $A = \{(x_j, \mu_A(x_j), \nu_A(x_j)) : x_j \in X; j = 1, \dots, n\}$ and $B = \{(x_j, \mu_B(x_j), \nu_B(x_j)) : x_j \in X; j = 1, \dots, n\}$ denote IF-sets in the universe of discourse X which correspond to nondegenerated orderings R^A and R^B expressed by two experts A and B . Then generalized Spearman’s rank correlation coefficient \tilde{r}_s is given by:

$$\tilde{r}_s(A, B) = 1 - \frac{3(n - 1)}{n(n + 1)} \sum_{j=1}^n [(\mu_A(x_j) - \mu_B(x_j))^2 + (\nu_A(x_j) - \nu_B(x_j))^2]. \tag{13}$$

Basic properties of \tilde{r}_s are given in the following lemmas.

Lemma 1. *If all n objects are univocally ranked by both experts, then the generalized Spearman's rank correlation coefficient \tilde{r}_s (13) is equivalent to the classical Spearman's rank correlation coefficient (1).*

Proof

Suppose that all elements are ranked by both preference systems A and B . If we also assume that there are no ties then there exist a one to one correspondence between membership/nonmembership functions describing the preference systems and usual ranks attributed to elements. Actually we have then ($\forall x_i \in X$):

$$R_i^A = (n - 1)\mu_A(x_i) + 1.$$

Analogically we can write: $R_i^B = (n - 1)\mu_B(x_i) + 1$. It is obvious that with perfect rankings we have the relationships $\nu_A(x_i) = 1 - \mu_A(x_i)$ and $\nu_B(x_i) = 1 - \mu_B(x_i)$. Therefore:

$$\begin{aligned} \tilde{r}_s(A, B) &= 1 - \frac{3(n - 1)}{n(n + 1)} \sum_{i=1}^n \left[\left(\frac{R_i^A - 1}{n - 1} - \frac{R_i^B - 1}{n - 1} \right)^2 + \left(\frac{n - R_i^A}{n - 1} - \frac{n - R_i^B}{n - 1} \right)^2 \right] \\ &= 1 - \frac{3(n - 1)}{n(n + 1)} \cdot \frac{2}{(n - 1)^2} \sum_{i=1}^n (R_i^A - R_i^B)^2 \\ &= 1 - \frac{6 \sum_{i=1}^n (R_i^A - R_i^B)^2}{n(n^2 - 1)} = r_s(R^A, R^B), \end{aligned}$$

which completes the proof.

Lemma 2. *For all $A, B \in IFS(X)$ describing preference systems as was given in (7)-(10) for $X = \{x_1, \dots, x_n\}$ we obtain:*

1. \tilde{r}_s is symmetric, which means $\tilde{r}_s(A, B) = \tilde{r}_s(B, A)$;
2. $\tilde{r}_s(A, B) = 1$ iff preference systems A and B are perfectly concordant and there are no ties;
3. $\tilde{r}_s(A, B) = -1$ iff preference systems A and B are perfectly discordant, which means that arrangement B must be the reverse of arrangement A and there are no ties;
4. $-1 \leq \tilde{r}_s(A, B) \leq 1$.

Proof

1. Symmetry comes obviously from definition (13).
2. The sum

$$S = \sum_{j=1}^n [(\mu_A(x_j) - \mu_B(x_j))^2 + (\nu_A(x_j) - \nu_B(x_j))^2] \tag{14}$$

equals zero iff when $\forall j \in \{1, \dots, n\} : \mu_A(x_j) = \mu_B(x_j)$ and $\nu_A(x_j) = \nu_B(x_j)$, which means when preference systems A i B are perfectly concordant and there are no ties. Then $\tilde{r}_s(A, B) = 1$.

3. The value of S is biggest (equivalently $\tilde{r}_s(A, B)$ is smallest) when $R^A = (1, 2, \dots, n)$ and $R^B = (n, n - 1, \dots, 1)$. These vectors correspond to IF-sets:

$$A = \{(x_1, 0, 1), (x_2, \frac{1}{n-1}, \frac{n-2}{n-1}), \dots, (x_i, \frac{i-1}{n-i}, \frac{n-1}{n-1}), \dots, (x_n, 1, 0)\};$$

$$B = \{(x_1, 1, 0), (x_2, \frac{n-2}{n-1}, \frac{1}{n-1}), \dots, (x_i, \frac{n-1}{n-i}, \frac{i-1}{n-1}), \dots, (x_n, 0, 1)\}.$$

Then $S = \frac{2n(n+1)}{3(n-1)}$, so we obtain $\tilde{r}_s(A, B) = -1$. This situation arises when preference systems A i B are perfectly discordant, which means that arrangement B must be the reverse of arrangement A and there are no ties.

4. From definition (13) we have $S \geq 0$ and therefore $\tilde{r}_s(A, B) \leq 1$, which summarized with point (3) of lemma finishes the proof.

5 Some Notes on the Distribution of \tilde{r}_s

Finding the exact distribution of generalized Spearman’s rank coefficient (13) for any number of values in X is complicated due to complex distributions of membership and nonmembership functions. Combinatorial considerations for small number of values in X lead us to find an exact theoretical distribution of \tilde{r}_s . We assume that each expert’s preference presented as a rank vector is equally probable (with missings and ties). Another assumption is that number of missing values in rank vector do not exceed half of the length of this vector. As an illustration we discuss below the exact distribution and quantiles corresponding to the very limited case of $n = 3$.

For $X \in \{x_1, x_2, x_3\}$ we have $n = 3$ (number of values in X), so number of missing values in rank vector $0 \leq k_3 \leq \lfloor \frac{n}{2} \rfloor = 1$. All possible rank vectors and membership/nonmembership functions in corresponding IF-set for an A expert (without permutations) are shown in Table 2.

Table 2. Possible rank vectors for $n = 3$

rank vector R^A	$\mu(A)$	$\nu(A)$	number of permutations
(1, 2, 3)	$(0, \frac{1}{2}, 1)$	$(1, \frac{1}{2}, 0)$	6
(1, 1, 1)	$(0, 0, 0)$	$(0, 0, 0)$	1
(1, 1, 2)	$(0, 0, 1)$	$(\frac{1}{2}, \frac{1}{2}, 0)$	3
(1, 2, 2)	$(0, \frac{1}{2}, \frac{1}{2})$	$(1, 0, 0)$	3
(?, 1, 2)	$(0, 0, \frac{1}{2})$	$(0, \frac{1}{2}, 0)$	6
(?, 1, 1)	$(0, 0, 0)$	$(0, 0, 0)$	3

For each of these vectors (considering permutations) we calculate the value of generalized Spearman’s rank correlation coefficient and obtain an the exact distribution of \tilde{r}_s given in Table 3 and basic quantiles given in Table 4.

Table 3. Exact distribution of \tilde{r}_s for $n = 3$

k	$P(\tilde{r}_s = k)$
-1	0.012396694
-0.75	0.049586777
-0.5	0.061983471
-0.25	0.173553719
0	0.099173554
0.25	0.198347107
0.5	0.111570248
0.75	0.223140496
1	0.070247934

Table 4. Quantiles of \tilde{r}_s for $n = 3$

range (α)	quantile $\tilde{r}_{s,\alpha}$
0.01	-1
0.05	-0.75
0.95	1
0.99	1

It is worth noting that the distribution is not symmetric around zero. To understand this property one has to realize how missing values and ties affect value of generalized Spearman’s rank correlation coefficient. Ties decrease values of membership and nonmembership functions for tied arguments. Missing values make membership and nonmembership functions equal zero for all assigned arguments and decrease all the other values. Due to lots of possibilities and necessity to consider mutual configuration of all coordinates in both rank vectors (number of which increases fast with growing n), it is hard to show these properties analytically. It is also important to notice that in case of variables dependence Pearson’s coefficient of linear correlation and as a result also Spearman’s coefficient of rank correlation are both not symmetrically distributed. Asymmetry of generalized Spearman’s rank correlation coefficient is intuitive and proved by numerical examples.

For large values of n (number of values in X set) calculating the exact distribution of generalized Spearman’s rank correlation coefficient with combinatorial methods is much more complicated. That is the reason why for $n > 5$ we calculate quantiles (necessary for dependence testing) using numerical methods (an appropriate program was written in MATLAB and R). An exemplary table with quantiles for $n = 7$ is given in Table 5.

Table 5. Quantiles for $n = 7$

n	missing values in rank vectors	$\tilde{r}_{s,0.01}$	$\tilde{r}_{s,0.025}$	$\tilde{r}_{s,0.05}$	$\tilde{r}_{s,0.95}$	$\tilde{r}_{s,0.975}$	$\tilde{r}_{s,0.99}$
7	(0, 0)	-11/14	-39/56	-5/8	37/56	3/4	23/28
	(0, 1)	-4/7	-1/2	-25/56	4/7	9/14	5/7
	(0, 2)	-27/56	-3/7	-3/8	23/56	27/56	31/56
	(0, 3)	-27/56	-3/7	-11/28	1/4	9/28	11/28
	(1, 1)	-13/56	-5/28	-1/8	37/56	41/56	45/56
	(1, 2)	-3/56	0	1/28	37/56	5/7	11/14
	(1, 3)	1/28	1/14	3/28	17/28	9/14	39/56
	(2, 2)	13/56	15/56	17/56	11/14	23/28	7/8
	(2, 3)	11/28	23/56	25/56	45/56	47/56	7/8
	(3, 3)	17/28	17/28	9/14	25/28	51/56	13/14

6 Testing Independence

Suppose we are interested in testing the null hypothesis that there is no association between two preference systems A and B . Then we can precise null hypothesis as:

$$H_0 : \text{preferences of } A \text{ and } B \text{ are independent.} \tag{15}$$

With one-sided alternative hypothesis:

$$H_1^1 : \text{preferences of } A \text{ and } B \text{ are positively dependent (concordant),} \tag{16}$$

we reject H_0 when $\tilde{r}_s \geq \tilde{r}_{s,1-\alpha}$, where $\tilde{r}_{s,\alpha}$ is α -quantile of the \tilde{r}_s distribution.

With one-sided alternative hypothesis:

$$H_1^2 : \text{preferences of } A \text{ and } B \text{ are negatively dependent (discordant),} \tag{17}$$

we reject H_0 when $\tilde{r}_s \leq \tilde{r}_{s,\alpha}$.

With two-sided alternative hypothesis:

$$H_1^3 : \text{preferences of } A \text{ and } B \text{ are dependent,} \tag{18}$$

we reject H_0 when $\tilde{r}_s \geq \tilde{r}_{s,1-\frac{\alpha}{2}}$ or $\tilde{r}_s \leq \tilde{r}_{s,\frac{\alpha}{2}}$. It comes from the fact that \tilde{r}_s coefficient does not have distribution symmetric of zero, which can be easily seen from exact distributions for $n \in \{3, 4, 5\}$.

Therefore we obtain full algorithm of testing association between two preference systems using generalized Spearman's rank correlation coefficient given by (13) and calculated quantiles.

Example 1(cont.)

Lets analyze an example, which shows how to construct preferences model and test dependence hypothesis using generalized Spearman's rank correlation coefficient given by (13). Let us continue the example of the two users who set their preferences in relation to seven TV channels.

Table 6. Example: IF-sets

channel	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$\mu_A(x)$	1/6	3/6	0	2/6	3/6	0	5/6
$\nu_A(x)$	4/6	1/6	5/6	3/6	1/6	0	0
$\mu_B(x)$	0	3/6	1/6	0	5/6	1/6	4/6
$\nu_B(x)$	5/6	2/6	3/6	0	0	3/6	1/6

On the basis of rank vectors we construct corresponding IF-sets. Our set of ordered variables is $X = \{x_1, \dots, x_7\}$, functions μ_A, ν_A, μ_B and ν_B are defined in Table 6.

Then we calculate a value of the generalized Spearman rank correlation coefficient using (13):

$$\begin{aligned} \tilde{r}_s(A, B) &= 1 - \frac{3(n-1)}{n(n+1)} \sum_{j=1}^n [(\mu_A(x_j) - \mu_B(x_j))^2 + (\nu_A(x_j) - \nu_B(x_j))^2] \\ &= 1 - \frac{19}{56} = \frac{37}{56}. \end{aligned}$$

One should see that due to results in Table 5 this value is exactly equal to the 0,95-quantile in the situation taken into consideration, so when $n = 7$ and number of missing values in rank vectors is $(1, 1)$. Coefficient is much greater than zero, so we may assume that there is a monotonic positive dependence between analysts' preferences.

Testing the null hypothesis (15) about independence against the alternative one (16) on significance level $\alpha = 0,05$ we reject H_0 and confirm concordance (i.e. we conclude that there is a significant association between preferences of our two users A and B and their preferences on TV channels are concordant).

7 Conclusions

In the paper we have suggested how to generalize the well-known Spearman's rank correlation coefficient to situations in which not all elements could be ordered. IF-sets have been used for modelling such defects in data sets. It is worth noting that the proposed coefficient is not only a tool applicable in descriptive statistics but could be effectively used for testing independence. Thus our coefficient seems to be a promising tool for the information retrieval, data mining and other applications where missing or noncomparable information is so often.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17(6), 734–749 (2005)
2. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87–96 (1986)

3. Atanassov, K.: *Intuitionistic Fuzzy Sets: Theory and Applications*. Physica-Verlag, Heidelberg (1999)
4. Denceux, T., Masson, M.H., Hebert, P.A.: Nonparametric rank-based statistics and significance tests for fuzzy data. *Fuzzy Sets and Systems* 153, 1–28 (2005)
5. Gibbons, J.D., Chakraborti, S.: *Nonparametric Statistical Inference*. Marcel Dekker, Inc., New York (2003)
6. Grzegorzewski, P.: The generalized Spearman's rank correlation coefficient. In: *Proceedings of the Tenth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Perugia, Italy, July 4-9, pp. 1413–1418 (2004)
7. Grzegorzewski, P.: The coefficient of concordance for vague data. *Computational Statistics and Data Analysis* 51, 314–322 (2006)
8. Grzegorzewski, P.: Kendall's correlation coefficient for vague preferences. *Soft Computing* 13, 1051–1061 (2009)
9. Hebert, P.A., Masson, M.H., Denceux, T.: Fuzzy rank correlation between fuzzy numbers. In: *Proceedings of the 10th IFSA World Congress-IFSA 2003*, Istanbul-Turkey, June 29-July 2, pp. 224–227 (2003)
10. Macdonald, C., He, B., Ounis, I.: Predicting Query Performance in Intranet Search. In: *Proceedings of the ACM SIGIR 2005 Query Prediction Workshop*, Salvador, Brazil (August 19, 2005)
11. Roy, B., Slowinski, R.: Criterion of distance between technical programing and socio-economic priority. *Oper. Res.* 27, 45–60 (1993)
12. Zadeh, L.A.: Fuzzy sets. *Inform. and Control* 8, 338–353 (1965)

Premium Based on Mixture Utility

Jana Špírková

Department of Quantitative Methods
and Information Systems,
Faculty of Economics,
Matej Bel University,
Tajovského 10,
975 90 Banská Bystrica, Slovakia.

Abstract. Aggregation functions and utility functions belong to very interesting parts of modern decision making theory. We develop basic concept of the connection of aggregation functions theory and utility theory to determine gross annual premium in general insurance. We introduce specific values of the gross annual premium on the basis of aggregation of the person's utility functions which were determined empirically based on a short personal interview. Moreover, by specific utility function we determine minimal gross annual premium acceptable for the insurer.

Keywords: Utility function, Mixture utility operator, Risk averse, Risk loving.

1 Introduction

This paper was inspired by the books Modern Actuarial Risk Theory [3] and Actuarial Models - The Mathematics of Insurance [10] where the authors introduce a theoretical model for determination of maximal and minimal premium in insurance. Moreover, in [10] Rotar introduces many types of classical utility functions, for example, positive-power function $u(x) = x^a$ for all $x \geq 0$ and some $a > 0$; negative-power function $u(x) = \frac{1}{x^a}$ for all $x > 0$ and some $a > 0$; logarithmic function $u(x) = \ln x$ for $x > 0$, etc.

However, in real life people do not behave according to theoretical utility functions. There is a psychological problem rather than a mathematical one. Seriousness and also uncertainty of respondent's answers depend on situation, on form of questions asked, on time which respondents have, and on many psychological and social factors. We can find a very interesting approach about utility functions in [4].

In our paper we introduce a possibility of the determination of the person's utility function on the basis of personal interview with virtual money.

Firstly, we develop one type of aggregation operators, so-called mixture utility operators - MU_g , generalized mixture utility operators - GMU_g and ordered generalized mixture utility operators - $OGMU_g$. These operators represent some

extension of mixture operators, generalized mixture operators and ordered generalized mixture operators [2], [5], [6], [11], [12], [13]. Moreover, we aggregate a number of the person’s utility functions obtained from the interview using the mixture utility operators above. Consequently, we determine maximal value of the gross annual premium in general insurance by means of final aggregated utility function.

Our paper is organized as follows: in Section 2 we recall basic properties of utility functions and their applications in insurance [1], [3], [10]. In Section 3 we develop MU_g operator, GMU_g operator and $OGMU_g$ operator and sufficient conditions for their non-decreasingness. In Section 4 we introduce a specific application of the aggregated utility function on the determination of gross annual premium. Moreover, we introduce corresponding minimal premium acceptable for insurer. Finally, in Section 5 some conclusions and indications of our next investigation about mentioned topic are included.

2 Preliminaries

In this section we recall you some basic features of utility function and expected utility, too. We recall some definitions of different types of mixture operators and some sufficient conditions for their non-decreasingness.

2.1 Utility Function

Utility function may be used as a basis for describing individual approaches to risk. Three basic approaches have been characterized. Opposite cases refer to *risk loving* and *risk averse* who accepts favorable gambles only. There is risk-neutral between these two extremes. Risk-neutral behavior is typical of persons who are enormously wealthy. Many people may be both risk averse and loving, depending on the range of monetary values being considered.

The theorem below describes properties of the utility function and its expected value.

Theorem 1. (*Jensen’s inequality*) [3], [10] *Let X be a random variable (with a finite expectation). Then, if $u(x)$ is concave,*

$$E[u(X)] \leq u(E[X]). \tag{1}$$

If $u(x)$ is convex,

$$E[u(X)] \geq u(E[X]). \tag{2}$$

Equality holds if and only if $u(x)$ is linear according to X or $var(X) = 0$.

2.2 Utility of the Insured

Now, suppose that our respondent has two alternatives - to buy insurance or not. Suppose that he owns a capital w and that he values wealth by the utility

function u . Let's assume he is insured against a loss X for a gross annual premium GP . If he is insured that means a certain alternative. This decision gives us the utility value $u(w - GP)$. If he is not insured that means an uncertain alternative. In this case the expected utility is $E[u(w - X)]$. Based on Jensen's inequality (I) we get

$$E[u(w - X)] \leq u(E[w - X]) = u(w - E[X]) \leq u(w - GP). \tag{3}$$

Since utility function u is a non-decreasing continuous function, this is equivalent to $GP \leq P^{max}$, where P^{max} denotes the maximum premium to be paid. This so-called *zero utility premium* is the solution to the following utility equilibrium equation

$$E[u(w - X)] = u(w - P^{max}). \tag{4}$$

The difference $(w - P^{max})$ is also called *certainty equivalent*.

We make a following consideration, (II).

Assume

- $GP = \gamma \cdot C$, ($0 \leq C \leq X$), where C is a claim payment,
- p is a probability of insured event,
- $\Omega = \{0, 1\}$ is a probability space (0 - does not occur, 1 - occurs), $P(1) = p$, $P(0) = 1 - p$,
- $x \in \Omega \rightarrow R^2_+$, where $x_0 = w - \gamma \cdot C$ and $x_1 = w - X + (1 - \gamma) \cdot C$ is a random variable.

In insurance the expected utility after insured event is

$$E[u(x)] = (1 - p) \cdot u(w - \gamma \cdot C) + p \cdot u(w - X + (1 - \gamma) \cdot C). \tag{5}$$

Expected profit EP is given as follows

$$EP = (1 - p) \cdot \gamma \cdot C + p \cdot (1 - \gamma) \cdot C = (\gamma - p) \cdot C. \tag{6}$$

On the basis of the expression (6) we can say that in a case of net annual premium it holds $\gamma = p$.

Therefore, we can rewrite the equation (5) by

$$E[u(x)] = (1 - p) \cdot u(w - p \cdot C) + p \cdot u(w - X + (1 - p) \cdot C). \tag{7}$$

If the insured is risk averse and his utility function is concave. Hence, we can determine maximal utility as follows

$$\frac{\partial u}{\partial C} = (1 - p) \cdot u'(w - p \cdot C) \cdot (-p) + p \cdot u'(w - X + (1 - p) \cdot C) \cdot (1 - p), \tag{8}$$

hence $C = X$, what means that the insured is willing to pay the premium, which is of equal value as the loss.

2.3 Utility of the Insurer

The insurer with utility function $U(W)$ and capital W , with insurance of loss X for a premium GP must satisfy the inequality

$$E[U(W + GP - X)] \geq U(W), \tag{9}$$

and hence for the minimal accepted premium P^{min}

$$U(W) = E[U(W + P^{min} - X)]. \tag{10}$$

D. Bernoulli himself suggested as a good candidate for the “natural” utility function $U(W) = \ln W$, assuming that the increment of the utility is proportional not to be the absolute but to the relative growth of the capital. More specifically, if capital W is increased by a small dW , then the increment of the utility, $du(W)$, is proportional to $\frac{d(W)}{W}$, that is

$$dU = k \cdot \frac{dW}{W} \tag{11}$$

for a constant k . The solution to this equation is

$$U(W) = k \cdot \ln W + C, \tag{12}$$

where C is another constant. The values of k and C do not matter, because an expression (12) represents an affine transformation.

Because it might be difficult to determine utility function for insurance company, we determine minimal premium acceptable for insurer by classic model of utility function $U(W) = \ln W$.

3 Mixture Utility Operator

Mixture operators were introduced in [5], [7], [9] and they are actually weighting arithmetic means weighted by weighting functions $g(x)$. Weights in mentioned mixture operators depend on input values. For more information see also [11]-[13].

However, a mixture operator need not be non-decreasing. Marques-Pereira and Pasi [5] stated first sufficient condition for a weighting function g in order mixture operator is non-decreasing.

In this part we develop so-called mixture utility operators, generalized mixture utility operators and ordered generalized mixture utility operators.

Suppose that each alternative \mathbf{u} is characterized by utility vector

$\mathbf{u} = (u_1, \dots, u_n) \in [0, 1]^n$, where $n \in N - \{1\}$ is the number of aggregated utility values.

Definition 1. Mixture utility operator $MU_g : [0, 1]^n \rightarrow [0, 1]$ is the arithmetic mean weighted by a continuous weighting function $g : [0, 1] \rightarrow]0, \infty[$ given by

$$MU_g(u_1(x), \dots, u_n(x)) = \frac{\sum_{i=1}^n g(u_i(x)) \cdot u_i(x)}{\sum_{i=1}^n g(u_i(x))}, \tag{13}$$

where $(u_1(x), \dots, u_n(x))$ is a vector of utility values for fixed $x, x \in R$.

Observe that due to the continuity of weighting function g , each mixture utility function MU_g is continuous and idempotent.

On the basis of Definition 2. a generalized mixture utility operator can be defined as follows:

Definition 2. Generalized mixture utility operator $GMU_{\mathbf{g}} : [0, 1]^n \rightarrow [0, 1]$ is given by

$$GMU_{\mathbf{g}}(u_1(x), \dots, u_n(x)) = \frac{\sum_{i=1}^n g_i(u_i(x)) \cdot u_i(x)}{\sum_{i=1}^n g_i(u_i(x))}, \tag{14}$$

where $(u_1(x), \dots, u_n(x))$ is a vector of utility values for fixed $x, x \in R$ and $\mathbf{g} = (g_1, \dots, g_n)$ is a vector of weighting functions.

Clearly, generalized mixture utility functions are continuous and idempotent.

Definition 3. Ordered generalized mixture utility function $OGMU_{\mathbf{g}} : [0, 1]^n \rightarrow [0, 1]$ is given by

$$OGMU_{\mathbf{g}}(u_1, \dots, u_n) = \frac{\sum_{i=1}^n g_i(u_{(i)}(x)) \cdot u_{(i)}(x)}{\sum_{i=1}^n g_i(u_{(i)}(x))}, \tag{15}$$

where $\mathbf{g} = (g_1, \dots, g_n)$ is a vector of continuous weighting functions and

$(u_{(1)}(x), \dots, u_{(n)}(x))$ is a non-decreasing permutation of a vector of utility values for fixed $x, x \in R$.

However, mixture utility operators do not have to be non-decreasing.

Following conditions for mixture utility operators $MU_g : [0, 1]^n \rightarrow [0, 1]$, $GMU_{\mathbf{g}} : [0, 1]^n \rightarrow [0, 1]$ and $OGMU_{\mathbf{g}} : [0, 1]^n \rightarrow [0, 1]$ are sufficient conditions for their non-decreasingness. These conditions were determined on the basis conditions of the monotonicity of mixture operators, [5], [11]-[13].

Proposition 1. Let $g : [0, 1] \rightarrow]0, \infty[$ be a non-decreasing smooth weighting function which satisfies the next condition:

$$0 \leq g'(u(x)) \leq g(u(x)) \quad \text{for all } u(x) \in [0, 1], \tag{16}$$

for fixed $x, x \in R$.

Then $MU_g : [0, 1]^n \rightarrow [0, 1]$ is an aggregation operator for each $n \in N, n > 1$.

In the next part we recall more general sufficient conditions mentioned in [6], [13] according to mixture utility operator.

Proposition 2. Let $g : [0, 1] \rightarrow]0, \infty[$ be a non-decreasing smooth weighting function which satisfies the condition:

$$0 \leq g'(u(x))(1 - u(x)) \leq g(u(x)) \quad \text{for all } u(x) \in [0, 1], \quad (17)$$

for fixed $x, x \in R$.

Then $MU_g : [0, 1]^n \rightarrow [0, 1]$ is an aggregation operator for each $n \in N, n > 1$.

Moreover, we have improved sufficient condition ([17]), but constrained by n .

Proposition 3. For a fixed $n \in N, n > 1$, let $g : [0, 1] \rightarrow]0, \infty[$ be a non-decreasing smooth weighting function satisfying the condition:

$$\frac{g^2(u(x))}{(n - 1)g(1)} + g(u(x)) \geq g'(u(x))(1 - u(x)) \quad \text{for all } u(x) \in [0, 1], \quad (18)$$

for fixed $x, x \in R$.

Then $MU_g : [0, 1]^n \rightarrow [0, 1]$ is an aggregation operator.

In the next proposition we introduce a sufficient condition for non-decreasingness of generalized mixture utility operators.

Proposition 4. For a fixed $n \in N, n > 1, i = 1, \dots, n$, let $g_i : [0, 1] \rightarrow]0, \infty[$ be a non-decreasing smooth weighting functions, such that

$$\frac{g_i^2(u(x))}{\sum_{j \neq i} g_j(1)} + g_i(u(x)) \geq g'_i(u(x)) \cdot (1 - u(x)) \quad \text{for all } u(x) \in [0, 1] \quad (19)$$

for fixed $x, x \in R$.

Then $GMU_{\mathbf{g}} : [0, 1]^n \rightarrow [0, 1]$, where $\mathbf{g} = (g_1, \dots, g_n)$, is an aggregation operator.

4 Maximal and Minimal Premium

In practice, the utility function can be determined empirically by a personal interview made by the decision maker. This function may be constructed from the information gleaned from the short interview. The respondent may use this function in any personal decision analysis in which the payoff falls between 0 and 30,000€. We recall the interview which is compiled as follows:

Suppose that you are owner of an estate that has the possible loss of 30,000€ in the future. However, you have a possibility to withdraw from this possible loss under the penalty the amount: 1,000€; 5,000€; 10,000€; 15,000€; 25,000€. Your portfolio manager can provide you with information expressing the probability of the loss the 30,000€.

What would be the biggest probability of the loss, to retain risk of the possible loss above?

Only a few proportioned graphic points are required. We have information (data) from three respondents. They have very similar approach to risk, that means they are risk averse for larger losses and risk loving for smaller losses. Utility function for the first respondent is determined by points (0; 1), (-1, 000; 0.85), (-5, 000; 0.75), (-10, 000; 0.60), (-15, 000; 0.60), (-25, 000; 0.20), (-30, 000; 0.00).

From the second respondent we obtain points as follows: (0; 1), (-1, 000; 0.85), (-5, 000; 0.75), (-10, 000; 0.60), (-15, 000; 0.50), (-25, 000; 0.40), (-30, 000; 0.00), and from the last one (0; 1), (-1, 000; 0.85), (-5, 000; 0.70), (-10, 000; 0.75), (-15, 000; 0.60), (-25, 000; 0.40), (-30, 000; 0.00).

Observe that these utility functions are for larger losses concave and for smaller losses convex, as shown in Figure 11

In this case we have three utility functions which we aggregate by MU_g operator and $OGMU_{\mathbf{g}}$ operator in order to get only one *collective* utility function for people with similar approach to risk.

We decided to aggregate utility values by MU_g operator because we can observe as weights rise continuously according to the utility value. Also, we aggregate individual utility values by means of $OGMU_{\mathbf{g}}$ operator to allocate higher weight to the higher utility value. That means we give the higher weight to the more risk averse utility value. Moreover, we can observe a modification of weights continuously. We use weighting function $g(u(x)) = 0.75u^2(x) + 0.25$ for aggregation with MU_g function and vector of weighting functions $\mathbf{g} = (g_1, g_2, g_3)$, where $g_1(u(x)) = 0.75u^2(x) + 0.25$, $g_2(u(x)) = 0.5u^2(x) + 0.5$, $g_3(u(x)) = 0.2u^2(x) + 0.8$ for aggregation with $OGMU_{\mathbf{g}}$ operator. All used weighting functions fulfill monotonicity conditions (16) - (19).

We decided to use such aggregation in order to see and be able to investigate weight by means of continuous mixture utility function.

Different person's utility functions for our three respondents and mixture utility functions were created by the SPSS system and they are presented in Table 1. This table also provides the values of Adjusted R square, F statistics and significance level for individual utility functions and for aggregated mixture utility operators.

Table 1. Person's utility functions and mixture utility functions

Utility Function	Adjusted F R square	Sig.
$u_1(x) = 5.549 \cdot 10^{-14}x^3 + 2.183 \cdot 10^{-9}x^2 + 4.784 \cdot 10^{-5}x + 0.948$	0.964	54.379 0.004
$u_2(x) = 1.191 \cdot 10^{-13}x^3 + 5.108 \cdot 10^{-9}x^2 + 7.801 \cdot 10^{-5}x + 0.977$	0.966	57.871 0.004
$u_3(x) = 1.003 \cdot 10^{-13}x^3 + 3.762 \cdot 10^{-9}x^2 + 5.425 \cdot 10^{-5}x + 0.955$	0.965	55.515 0.004
$MU_g(x) = 9.301 \cdot 10^{-14}x^3 + 3.726 \cdot 10^{-9}x^2 + 6.004 \cdot 10^{-5}x + 0.960$	1	1.698E6 0.000
$OGMU_{\mathbf{g}}(x) = 9.969 \cdot 10^{-14}x^3 + 4.025 \cdot 10^{-9}x^2 + 6.281 \cdot 10^{-5}x + 0.962$	1	1.030E4 0.000

Remark 1. Expected utility is calculated by the well-known formula

$$E[u(X)] = \sum_{i=1}^n u(x_i) \cdot p_i, \tag{20}$$

where $X = (x_1, x_2, \dots, x_n)$ is a vector of the possible alternatives and p_i is the probability of alternative x_i .

Expected utilities can be calculated by linear function, too which is determined uniquely by points $[-30,000; u(-30,000)]$ and $[0, u(0)]$. In both cases we get the same values of the expected utilities.

In the figure below are the individual utility functions and final *OGMUG* function created by the SPSS system.

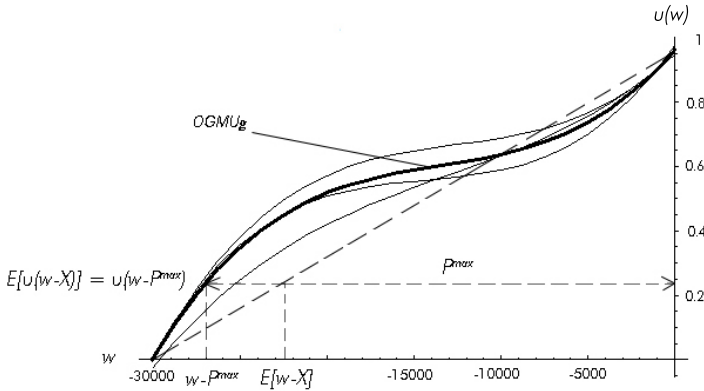


Fig. 1. Utility functions of selected respondents (functions from Table 1)

We determine minimal premium by the means of (10) with respect to utility function for insurer $U(x) = \ln x$ with his basic capital $W = 2,655,513.51\text{€}$ and loss $X = 30,000\text{€}$.

Equation (10) can be rewritten as follows:

$$U(W) = p \cdot U(W + P^{min} - X) + (1 - p) \cdot U(W + P^{min}), \tag{21}$$

and hence

$$W = (W + P^{min} - X)^p \cdot (W + P^{min})^{(1-p)}. \tag{22}$$

We determine individual minimal premiums with corresponding probability by the Mathematica 5 system.

Table 2. Expected Utility, Maximal and Minimal Premium

probability of loss p	$E[MU_g]$	P^{max} according to MU_g	$E[OGMU_g]$	P^{max} according to $OGMU_g$	P^{min}
0.00	0.960000	0.00	0.962000	0.00	0.00
0.01	0.950409	161.35	0.952466	153.29	301.69
0.05	0.912047	841.73	0.914329	799.08	1,508.10
0.1	0.864093	1,786.65	0.866657	1,694.17	3,015.34
0.2	0.768186	4,155.05	0.771314	3,928.81	6,027.24
0.3	0.672279	7,904.69	0.675971	7,463.95	9,035.69
0.4	0.576372	15,656.20	0.580628	15,854.90	12,040.70
0.5	0.480465	21,018.60	0.485285	21,358.90	15,042.40
0.6	0.384558	23,877.00	0.389942	24,116.10	18,040.60
0.7	0.288651	25,896.30	0.294599	26,055.30	21035.50
0.8	0.192744	27,494.30	0.199256	27,590.20	24,027.00
0.9	0.096837	28,835.00	0.103913	28,879.00	27,015.20
1.0	0.00093	30,000.00	0.008570	30,000.00	30,000.00

Remark 2. Utility functions are used to compare investments mutually. For this reason, we can scale a utility function by multiplying it by any positive constant and/or translate it by adding any other constant (positive or negative). This kind of transformation is called a positive affine transformation. All our results would be the same.

5 Conclusion

We have shown how to construct person's utility function and we have calculated maximal premium for loss of 30 000€ according to the mixture utility function. On the basis of mixture utility function and ordered generalized mixture utility function we have determined maximal premium for persons having approximately similar approach to risk. Another utility function would be required if evaluating a decision with more extreme payoffs or if our respondent's attitudes change because of a new job or lifestyle. Moreover, the utility function must be revised from time to time. In our future work we would like to investigate mixture utility operators with other weighting functions, insurer's utility function and to extend and in more detail investigate mentioned model with different interviews and fictive games.

Moreover, we want to investigate asymmetric information in mixture utility theory.

Acknowledgments. This work was supported by Faculty of Economics, Matej Bel University.

References

1. Brunovský, P.: *Mikroekonomics* (2006) (in Slovak), <http://www.iam.fmph.uniba.sk/skripta/brunovsky2/PART0.pdf>
2. Calvo, T., Mesiar, R., Yager, R.R.: Quantitative Weights and Aggregation, *Fellow. IEEE Transaction on Fuzzy Systems* 2(1), 62–69 (2004)
3. Kaas, R., Goovaerts, M., Dhaene, J., Denuit, M.: *Modern Actuarial Risk Theory*. Kluwer Academic Publishers, Boston (2001)
4. Khurshid, A.: The end of rationality? In: *Abstracts, Tenth International Conference on Fuzzy Set Theory and Applications, FSTA 2010*, pp. 10–11. Slovak Republic (2010)
5. Marques-Pereira, R.A., Pasi, G.: On non-monotonic aggregation: Mixture operators. In: *Proceedings of the 4th Meeting of the EURO Working Group on Fuzzy Sets (EUROFUSE 1999) and 2nd International Conference on Soft and Intelligent Computing (SIC 1999)*, Budapest, Hungary, pp. 513–517 (1999)
6. Mesiar, R., Špirková, J.: Weighted means and weighting functions. *Kybernetika* 42(2), 151–160 (2006)
7. Ribeiro, R.A., Marques-Pereira, R.A.: Generalized mixture operators using weighting functions: A comparative study with WA and OWA. *European Journal of Operational Research* 145, 329–342 (2003)
8. Ribeiro, R.A., Marques-Pereira, R.A.: Weights as functions of attribute satisfaction values. In: *Proceedings of the Workshop on Preference Modelling and Applications (EUROFUSE)*, Granada, Spain, pp. 131–137 (2001)
9. Ribeiro, R.A., Marques-Pereira, R.A.: Aggregation with generalized mixture operators using weighting functions. *Fuzzy Sets and Systems* 137, 43–58 (2003)
10. Rotar, V.I.: *Actuarial Models - The Mathematics of Insurance*. Chapman & Hall/CRC Press, Boca Raton, London (2007)
11. Špirková, J.: *Weighting Functions For Aggregation Operator: 3rd International Summer School On Aggregation Operators*, pp. 127–130. Università Della Svizzera Italiana, Lugano (2005)
12. Špirková, J.: Mixture and quasi-mixture operators. In: *IPMU 2006*, France, pp. 603–608 (2006)
13. Špirková, J.: *Dissertation thesis, Weighted aggregation operators and their applications*, Bratislava (2008)
14. Špirková, J.: Mixture utility in general insurance. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) *IPMU 2010. CCIS*, vol. 80, pp. 712–721. Springer, Heidelberg (2010)
15. Yager, R.R.: Generalized OWA Aggregation Operators. *Fuzzy Optimization and Decision Making* 3, 93–107 (2004)

Which Should We Try First? Ranking Information Resources through Query Classification

Joshua Church and Amihai Motro

Department of Computer Science
George Mason University
Fairfax, VA, USA
jchurch2@gmu.edu

Abstract. Users seeking information in distributed environments of large numbers of disparate information resources are often burdened with the task of repeating their queries for each and every resource. Invariably, some of the searched resources are more productive (yield more useful documents) than others, and it would undoubtedly be useful to try these resources first. If the environment is *federated* and a single search tool is used to process the query against all the disparate resources, then a similar issue arises: Which information resources should be searched first, to guarantee that useful answers are streamed to users in a timely fashion. In this paper we propose a solution that incorporates techniques from text classification, machine learning and information retrieval. Given a set of pre-classified information resources and a keyword query, our system suggests a relevance ordering of the resources. The approach has been implemented in prototype form, and initial experimentation has given promising results.

1 Introduction

Users seeking information in distributed environments of large numbers of disparate information resources are often burdened with the task of repeating their queries for each and every resource. Examples include searching for news items on a specific topic among hundreds of news feeds; searching for a job or an apartment in multiple classified ads repositories; or searching for technical advice in a multitude of support sites and discussion groups. Invariably, some of the searched resources are more productive (yield more useful documents) than others, and it would undoubtedly be useful to try these resources first.

If the environment is *federated* and a single search tool is used to process the query against all the disparate resources, then a similar issue arises: Which information resources should be searched first to guarantee that useful answers are streamed to users in a timely fashion.

This issue can be formalized abstractly as follows. Consider a collection $\{R_1, \dots, R_n\}$ of information resources (e.g., document collections), assume a

query Q is processed in the entire collection $R = \cup_1^n R_i$, and let A be its *ranked* answer. Let A_i be the subset of A that originates from R_i (the subsets A_i are not necessarily disjoint) and let ω_i denote the *contribution* of A_i to A . We define the *ranking* of the resource environment R for the given query Q as the ordering of the individual resources R_i according to their ω_i values. Sub-answer contribution could be measured in any of a number of ways, and should reflect both the total number of answer elements in A_i (quantity) and their relative ranking (quality).

The challenge we address in this paper is to design a methodology that *approximates* this order. That is, given a query Q against a collection of information resources R , rank the resources in R in terms of their expected contribution to the query Q . With such a ranking, users who seek answers to queries in a multi-source environment (or meta-searchers that process queries in such environments) can achieve their goals more effectively.

The solution we propose is based on techniques from information retrieval, text classification and machine learning, and it makes several simplifying assumptions. It assumes that each of the information resources in R is *homogeneous*; that is, its documents are on a single subject. Specifically, it assumes that the given information resources have been classified with a pre-determined set of C_1, \dots, C_p categories (labels). Given a query Q , we attempt to *classify* it by the same set of categories; but rather than settle on a single classification, each query Q results in a *ranked list of classifications*. This list, in turn, implies an order of the information resources, which we suggest as an approximation of the order defined above.

Thus, the main challenge is to classify a query; that is, to map Q to a permutation of C_1, \dots, C_p . The main resource in our classification is a *semantic index*. This index is constructed from training documents, in a process that combines content acquisition, feature extraction, and latent semantic analysis (LSA) [5]. It provides the *background knowledge* necessary for classification. Essentially, this semantic index is an approximation of the traditional matrix of features (terms) by documents, in which the number of features has been reduced in a procedure called Single Value Decomposition (SVD). This new representation is known to mitigate the classical problems of synonymy (different terms have the same meaning) and polysemy (a term has multiple meanings). Into this space we also cast the query and compute the documents that are its k nearest neighbors using the traditional *cosine* similarity measure. Since each of the documents has an associated category C_i , the k nearest neighbors provide a multiset of categories. Using a voting approach, this multiset is used to infer a classifying order of the categories. In a final step, each category in the ordered classification of Q is replaced by the resources in R that are associated with this category.

The architecture of the system is described in detail in Section 3. This architecture was implemented in prototype form, weaving together readily available software components to perform the necessary content acquisition, feature extraction, learning and classification.

Our experiments are described and discussed in Section 4. Essentially, the experiments were designed to (1) validate the feasibility of the architecture and

to measure its performance, and (2) to draw conclusions as to the optimal values of two important classification parameters: the number of dimensions (features) used in the semantic index and the number of closest neighbors used in the classification. Our experiments showed that even with moderate amount of training (3533 documents classified by 11 categories), effectiveness of 71% (as measured by the F -measure) can be achieved. These results were achieved with relatively small values for the number of features (200) and the number of closest neighbors (25).

Section 5 concludes the paper with a brief summary and directions for further research. To put our work problem in its appropriate context, we begin with a brief discussion of related work.

2 Background

We assume that readers are familiar with basic concepts of information retrieval [10] and we focus our attention on two active areas of research that relate strongly to this work: database ranking and query classification. We note that database ranking may refer to two different tasks: ranking answers (sets of rows) that are retrieved from a database in response to queries, listing “better rows” first (an early example of this may be found in [11]), or to ranking of a collection of databases with respect to a particular information need. Our work here, and therefore this review, concerns the latter.

Keyword-based selection of multiple structured data sources is the subject of [16]. The authors construct keyword summaries of the databases that participate in the ranking process. Given a query, they compare its keywords to each database summary and measure its proximity to the database’s schema and content. The problem is original and the effectiveness of the approach has been demonstrated. Yet, keyword matching suffers intrinsically from poor precision and recall due to vocabulary mismatch [10]. For example, if users submit synonymous keywords that do not occur in database’s keyword summary, they might miss relevant resources. Our work addresses this issue by using latent semantic analysis, a method known for its ability to handle synonymy and polysemy [7].

Text categorization is an intensively researched area, and query classification, a sub-area, has been very active recently [12,15,13,9,2,3,8]. Our task is essentially to learn a text categorizer that maps short, noisy, and ambiguous sets of keywords to relevant resources [9]. In general, query classification research varies by (1) the machine learning (ML) approach and (2) the type of training data.

A basic necessity to any query classification task is the selection and acquisition of training data that are representative of users’ queries [9,12]. Essentially, a large collection of labeled content that is both general and adaptive enough to categorize queries is hard to find. Training data are often obtained from search engines results [15,3,12,13], click-through data in query logs [2,1], or open directory services [15,1]. Our solution utilizes document feeds that conform to the RSS (Really Simple Syndication) protocol — a widely used format for disseminating content on the Web. Typically, each such feed contains articles on a particular topic. The

assumption is that a sufficiently large set of documents obtained from each feed provide a reliable representation of the feed for the purpose of future classification.

We employ a classification technique called *transfer knowledge* or *background knowledge* which is recognized as an effective method for creating general purpose classifiers [4,14,17]. Background knowledge leverages training data from one classification task to apply to another related classification task [4,14,17]. Although previous research suggested the use of background knowledge [15,3], it considered the query answer as the training material; that is, the search results are examined for patterns that explain the query. Other classifiers described in the literature include rule-based classifiers [2], pattern matchers [13,12,2], Support Vector Machines (SVM) [13,12], and probabilistic classifiers [15,3]. Our classifier's performance is linear in the number of documents, which makes it suitable for large-scale deployment. Moreover, our solution classifies queries without knowledge of the corresponding result set, a feature essential in the application of query classification to resource ranking.

3 System Architecture

Our system consists of two principal phases common to classification tasks: *preparatory indexing* and *request processing*, with the former providing the knowledge necessary for the latter. Figure 1 illustrates this architecture. Preparatory indexing consists of three stages: content acquisition, feature extraction, and semantic indexing. In two additional stages, request processing uses two system assets — the output of the semantic indexing and a classified catalog of the available resources — to assign a set of resources to each user query. A more detailed description of these five stages follows.

3.1 Preparatory Phase

In the first phase, training documents are processed and preserved using the well-known vector space model of information retrieval. The product of this phase is an LSI index to be used for query processing.

Content Acquisition. Initially, users of this system choose a set of *labels* (categories) to be used in classification. For each such label, one or more document collections are selected. Each collection should include documents that correspond primarily to the particular label. This collection provides relevant background to *interpret* the meaning of the label. In this work we chose to work with document collections that are RSS feeds. Custom Java software that incorporates the ROME Java software library is used for managing feeds. Each document is retrieved with an HTTP request, its content is parsed, and the result is saved to a local store.

Feature Extraction. In the next stage, each parsed document is processed to obtain a list of features (terms). It is assumed that the features extracted are characteristic and descriptive of the label associated with the feed.

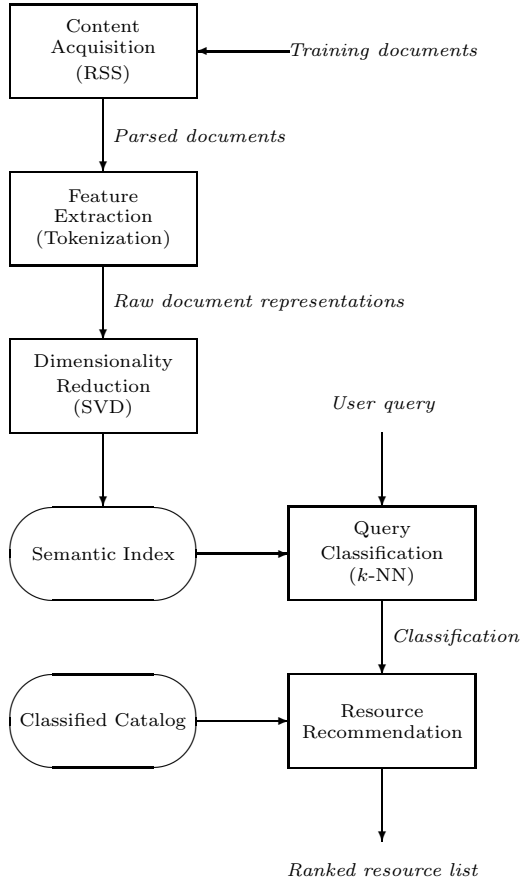


Fig. 1. System architecture

Semantic Indexing. Let the total number of acquired documents be m , and let the total number of extracted features be n . The documents are now represented in a matrix of n rows and m columns, with the value in position (i, j) denoting the *relevance* (significance) of the i 'th feature to the j 'th document. We measure this relevance with the well-known concept of *entropy*: Let $f_{i,j}$ denote the number of occurrences of the i 'th feature in the j 'th document, then $\sum_{j=1}^m (f_{i,j})$ is the total number of occurrences of this feature in all the documents, and $p_{i,j} = f_{i,j} / \sum_{j=1}^m (f_{i,j})$ is the relative frequency of occurrence. The value stored in position (i, j) is $p_{i,j} \cdot \log p_{i,j}$. Typically, each document will contain only a small number of the possible features, resulting in a very sparse matrix. The row-dimensionality of this matrix is then reduced, first by eliminating features that correspond to common words ("stop-words"), and then by using Single Value Decomposition (SVD). A complete explanation of SVD is outside the scope of this paper and may be found in [7].

3.2 Request Processing

The second phase is the repetitive processing of user queries. It involves two stages: query classification and resource ranking.

Query Classification. The output of the preparatory phase is a *semantic index*: a set of vectors representing the documents by means of their features. Initially, each user query is transformed into a similar vector using the same SVD process that was applied to the original matrix. Next, this vector is compared to all the vectors in the semantic index, and, using the well-known *cosine* measure of similarity, its k nearest neighbors (k -NN) are determined [6]. Since each of the k documents originated from a particular collection (feed), it is associated with a particular label. The k labels thus obtained are tallied and *ranked* according to their rate of occurrence in the set. In other words, the documents closest to the query “vote” on its classification.

Resource Ranking. In the final stage we assume that the collection of available resources has been pre-classified using the same set of labels. (Indeed, it may be assumed that the set of labels has been derived from this classification.) This catalog of resources is now used to match the query with a ranked list of resources that correspond to its ranked classification.

It should be noted that, in essence, there are three classifications in this work, and they use the same set of categories: The training documents and the catalog resources are assumed to be pre-classified, whereas user queries are classified by the system.

4 Experimentation and Discussion

To validate the approach outlined in this paper we conducted an experiment of moderate size. Our objective was two fold. The first objective was to validate that the architecture that we proposed can indeed deliver good results. The second objective was to experiment with two important parameters of query classification, namely the number of dimensions with which a document is represented, and the number of closest neighbors that would be used to vote on the classification.

4.1 Datasets

Training Documents. Our system uses 11 different classification labels typical in the newspaper domain; for example, business, sports, health, education, and so on. For training the system, we used RSS feeds of the Washington Post newspaper. RSS feeds are increasingly the dissemination method of choice of on-line resources, and the advantage of using newspaper feeds and labels is that the documents been classified by human editors and thus provide authoritative interpretation for the labels. A total of 34 RSS feeds were sampled and a total of 3533 documents were extracted. Figure 2 shows the breakdown of the 3533

documents by the 11 categories. As can be seen, the distribution of the documents is relatively balanced. The resource catalog used in the final classification stage assigns each resource to one of the 11 categories. A small example of such a catalog is shown in Table 1.

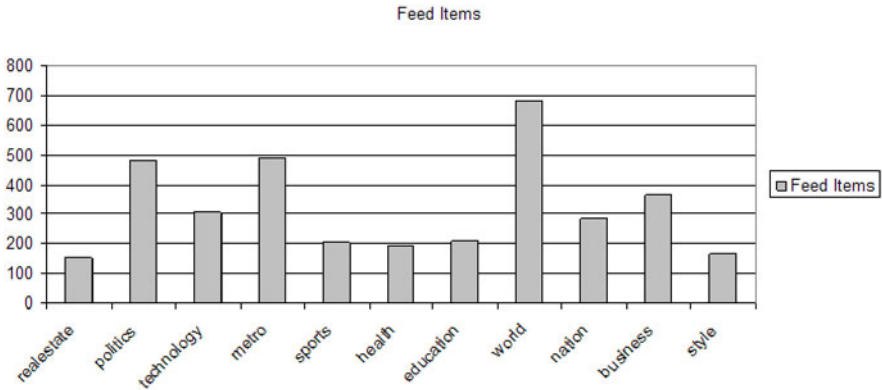


Fig. 2. Histogram of training documents

Table 1. A classified catalog (partial)

Category	Resources
Business	http://www.forbes.com/fdc/rss.html http://feeds.fool.com/usmf/foolwatch http://online.wsj.com/xml/rss/3_7086.xml
Sports	http://sports.yahoo.com/top/rss http://sports.espn.go.com/espn/rss/nfl/news http://content.usatoday.com/marketing/rss/rsstrans.aspx?feedId=sports1
Technology	http://rss.news.yahoo.com/rss/tech http://feeds.wired.com/wired/index http://www.infoworld.com/rss/news.xml

Test Queries. Our experiment used actual user queries submitted to the Google Web search engine. Specifically, the most frequent 100 queries for a given day were collected over a period of two weeks. From this set of 1,400 queries, a random sample of 66 queries was selected, providing for confidence level of 90%. These queries, typically a few keywords each, were classified “manually” using the same 11-label scheme. These authoritative classifications were later used to measure the accuracy of the classifications generated by the system.

4.2 Experimental Results

After the preparatory phase was completed, each of the 66 queries was classified 30 times, using 6 different values for the number of dimensions and 5 different values for the number of closest neighbors. A classification was correct, if the manually-assigned category matched the *top* predicted category. The classification of the set of 66 queries with a specific number of dimensions and neighbors was considered a single experiment, whose success was measured with the *F*-measure (the harmonic mean of the precision and recall). Figure 3 summarizes the results of the 30 experiments.

	dimensions				
neighbors	50	100	200	300	400
1	0.538064	0.552026	0.552026	0.552026	0.509091
10	0.675138	0.675138	0.696651	0.696651	0.61708
25	0.686009	0.686009	0.707071	0.707071	0.629213
50	0.686009	0.686009	0.707071	0.707071	0.629213
100	0.686009	0.686009	0.707071	0.707071	0.629213
200	0.686009	0.686009	0.707071	0.707071	0.629213

Fig. 3. *F*-Measure at various levels

An analysis of the variance of these results (using two-way ANOVA without replication) concluded that there is no significant interaction between the number of dimensions and the number of neighbors chosen, suggesting that they could be optimized independently. Observing the impact of dimensionality on the *F*-measure (Figure 4), it is apparent that increasing the number of dimensions improves performance through 200 dimensions, provides no improvement when the number is increased to 300, and worsens performance substantially thereafter. Observing the impact of the number of neighbors on the *F*-measure (Figure 5), it is apparent that increasing the number of closest neighbors (the number of documents that “vote” on the classification) improves performance through 25 neighbors. Once this number is reached, the quality of the classification remains unchanged. Combined, these three somewhat surprising conclusions suggest keeping the number of dimensions at 200 and the number of neighbors at 25.

4.3 Discussion

The overall results of this experiment are promising. Roughly speaking, the system can classify a query correctly (and recommend the appropriate resources) about 71% of time. And with various extensions and refinements (to be discussed later) we expect even further improvements.

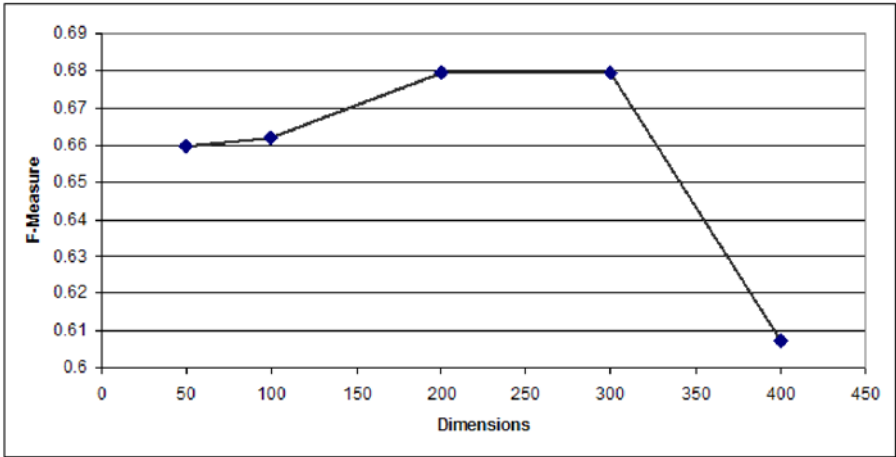


Fig. 4. F-measure vs. dimensions

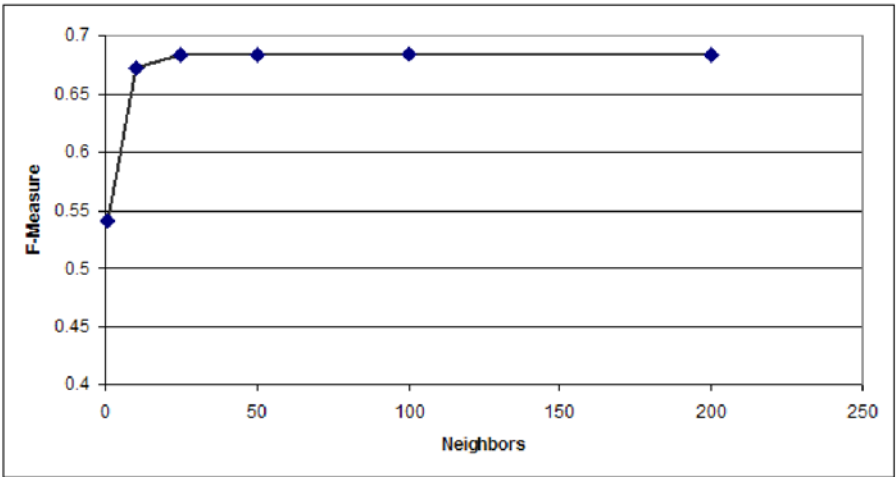


Fig. 5. F-measure vs. neighbors

The results of our experimentation with different classification parameters are also noteworthy. Intuitively, as the number of features used to describe a document increases, the semantics of documents are captured with more fidelity, and hence classification should be more accurate. Similarly, a larger number of close neighbors should be expected to minimize possible “noise” caused by the misclassification of some documents, resulting in a more robust classification. In practice, in both instances, increasing the corresponding parameters proved beneficial, but only to a certain level, beyond which there were no improvements

(and in the case of the dimensionality of vectors, performance eventually started to decline). These two results, combined with the discovery that these two parameters do not interact and could thus be optimized independently, suggest that “more is not always better”. A conclusion that has positive impact on time performance.

5 Conclusions and Future Work

The availability of multiple information resources against which a query may be processed raises the issue of which information resources would prove to be more productive. In this paper we addressed this issue in the context of resources that are collections of documents and queries that are sets of keywords. Specifically, given a large number of document collections and a keyword query, rank the collections in the order of relevance; that is, resources that are more likely to yield documents relevant to the query should be listed earlier.

To address this issue, we proposed and tested a machine learning approach in which we assumed that the given resources have been pre-classified by a set of categories, and the challenge is to correctly classify a given query by the same set of categories. Our query classification method is based on the notion of similarity, and produces not a single category, but a ranked order of categories. These, in turn, suggest a ranking order of the corresponding resources.

Our prototype system combined off-the-shelf tools for RSS feed acquisitions, feature extraction, and latent semantic indexing. The information resources we used were RSS feeds of a major newspaper. Initial experimentation demonstrated that F -measures of 71% can be achieved with moderate size background knowledge.

For this approach to work well, it is important to use sufficient number of training documents and use RSS feeds that accurately characterize the topics that interest the user. Note that available collections of resources need not be static, as they can be updated periodically with “feed crawls”.

There are many opportunities for further research and we mention here just four. First, when classifying a query we ranked the categories by their frequencies in the nearest neighbors set. This corresponds to a simple voting procedure in which all neighbors have equal votes. Another possibility here is to use a *weighted voting scheme*, in which the vote of each neighbor is weighted by its proximity to the query.

Second, we assumed that the information resources in the catalog have been pre-classified. A familiar argument is that this manual classification is laborious and inaccurate. An attractive proposition is to apply similar machine learning techniques to *automate the classification of the resources* as well.

Third, we assumed a pre-determined set of categories (business, sports, politics, and so on). Alternatively, we could *obtain the set of categories* from the classification of the resources (which was suggested above), and then use these in the other two classification processes (document training and user queries).

Finally, the classification of each query was by a ranked order of categories. With a small effort this classification could be converted into a *weighted vector* of categories. On the other hand, each of the given information resources have been assumed to fall into a single category (a somewhat restrictive assumption). An attractive approach is to classify each resource with a similar weighted vector of categories, and then use a similarity measure (such as the cosine) to find the resources that are the *closest neighbors* of the query classification vector.

References

1. Arguello, J., Callan, J., Diaz, F.: Classification-based resource selection. In: Proceedings of CIKM-2009, 18th ACM Conference on Information and Knowledge Management, pp. 1277–1286. ACM, New York (2009)
2. Beitzel, S.M., Jensen, E.C., Lewis, D.D., Chowdhury, A., Frieder, O.: Automatic classification of web queries using very large unlabeled query logs. *ACM Transactions on Information Systems* 25(2), Article 9 (April 2007)
3. Broder, A.Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., Zhang, T.: Robust classification of rare queries using web knowledge. In: Proceedings of SIGIR-2007, 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 231–238. ACM, New York (2007)
4. Do, C.B., Ng, A.Y.: Transfer learning for text classification. In: *Advances in Neural Information Processing Systems* 18, NIPS (2005)
5. Dumais, S.T.: Latent semantic indexing (LSI) and TREC-2. In: Proceedings of the Second Text Retrieval Conference, pp. 105–116. National Institute of Standards and Technology (NIST), Special Publication 500-215 (1993)
6. Feldman, R., Sanger, J.: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, Cambridge (2006)
7. Furnas, G.W., Deerwester, S.C., Dumais, S.T., Landauer, T.K., Harshman, R.A., Streeter, L.A., Lochbaum, K.E.: Information retrieval using a singular value decomposition model of latent semantic structure. In: Proceedings of SIGIR-1988, 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 465–480. ACM, New York (1988)
8. Geng, X., Liu, T.-Y., Qin, T., Arnold, A., Li, H., Shum, H.-Y.: Query dependent ranking using k -nearest neighbor. In: Proceedings of SIGIR-2008, 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 115–122. ACM, New York (2008)
9. Li, Y., Zheng, Z., Dai, H.K.: KDD CUP 2005 Report: facing a great challenge. *SIGKDD Explorations Newsletter* 7(2), 91–99 (2005)
10. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge (2008)
11. Motro, A.: VAGUE: A user interface to relational databases that permits vague queries. *ACM Transactions on Information Systems* 6(3), 187–214 (1988)
12. Shen, D., Pan, R., Sun, J.-T., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Q2C@UST: Our winning solution to query classification in KDD CUP 2005. *SIGKDD Explorations Newsletter* 7(2), 100–110 (2005)
13. Shen, D., Sun, J.-T., Yang, Q., Chen, Z.: Building bridges for web query classification. In: Proceedings of SIGIR-2006, 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 131–138. ACM, New York (2006)

14. Taylor, M.E., Stone, P.: Cross-domain transfer for reinforcement learning. In: Proceedings of ICML-2007, 24th International Conference on Machine Learning, pp. 879–886. ACM, New York (2007)
15. Vogel, D., Bickel, S., Haider, P., Schimpfky, R., Siemen, P., Bridges, S., Scheffer, T.: Classifying search engine queries using the web as background knowledge. ACM SIGKDD Explorations Newsletter 7(2), 117–122 (2005)
16. Yu, B., Li, G., Sollins, K., Tung, A.K.H.: Effective keyword-based selection of relational databases. In: Proceedings of SIGMOD-2007, the 2007 ACM SIGMOD International Conference on Management of Data, pp. 139–150. ACM, New York (2007)
17. Zelikovitz, S., Hirsh, H.: Using LSI for text classification in the presence of background text. In: Proceedings of CIKM-2001, 10th ACM International Conference on Information and Knowledge Management, pp. 113–118. ACM, New York (2001)

Context Modelling for Situation-Sensitive Recommendations

Stewart Whiting and Joemon Jose

School of Computing Science, University of Glasgow,
Scotland, G12 8QQ, UK
{stewh,jj}@dcs.gla.ac.uk

Abstract. Users are finding themselves interacting with increasingly complex software systems and expanding information resources. However many of these systems have little to no awareness of the personally-understood user context which expresses *why* they are being used. In this paper we propose a framework for modelling and proactively retrieving previously accessed and created information objects and resources that are within the context of a user's current situation. We first consider theories of context to understand the discrete aspects of context that may delineate a user's composite situations. With this we develop a framework for modelling user interaction in context along with a re-configurable algorithm for making personal recommendations for desired information objects based upon the environmental, content-based and task sequence contextual similarity of the current situation to past situations. To measure the effectiveness of our approach we use a two week activity log from four real users in a preliminary lab-based evaluation methodology. Initial results suggest the framework as a static personal recommendation algorithm is effective to varying degrees during periods of interaction for users of various characteristics.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval.

General Terms: personal information management, context, framework, recommender.

1 Introduction

A significant proportion of work for many users involves the re-use of existing information objects and resources [7] such as emails, web pages, documents and folders previously found or created. We consider this information interaction to occur within a personally-understood situation, comprised of a number of relevant contextual cues. Finding and recalling diverse information objects repeatedly from multiple sources when needed is often time-consuming and frustrating. Whilst many systems exist to provide ad-hoc search over a user's personal information space, our intention is to provide a system that can proactively assist

the user during information re-use activities without the need to explicitly state information needs through queries.

In this work, we propose the Personal Recommendation Framework (PRF), a context-aware system that can provide “relevant information to the user, where relevancy depends on the user’s task” [5]. The PRF proactively retrieves previously seen and created information objects that are implicitly considered ‘in context’ with the user’s current situation, in order to provide a ranked list of situation-sensitive recommendations. The PRF models ongoing user activity and contextual cues arising during sessions. From this, a number of recommendation algorithms individually attempt to retrieve useful items that were used in a previous similar contextual aspect (e.g. time or place etc.). The output of these separate algorithms is then combined to compose a view of a situation from which the most likely and ‘useful’ overall recommendations can then be extracted and provided to the user.

In Section 1.1 we present related work that addresses the need for systems that proactively assist users through an understanding of their task context. Section 2 provides a brief literature review on the theoretical foundations and critical perspectives of context and situation. Section 3 states our approach to modelling personal information interaction in context. Section 4 outlines the architecture of the PRF action recording, storage and recommendation functions, providing details of the algorithms and methods used. In Sections 5 and 6 we describe a method of collecting and incorporating real user activity in a lab-based study, with an evaluation tool that maintains user privacy. In addition, we propose two possible recall measures for evaluation. Finally, Section 7 discusses preliminary results of the effectiveness of 3 recommender configurations for 4 users of varying characteristics over a 2 week period. We conclude with a summary of our experiences and issues encountered during this study.

1.1 Related Work

There have been many systems developed to incorporate the task and information context in different scenarios. For example, personal agents such as CALVIN [3], an intelligent information retrieval (IR) agent which recommends similar web pages to those recently viewed uses the “WordSieve” algorithm to identify contextually relevant keywords in the interaction stream. From a retrieval perspective the “Stuff I’ve Seen” system [7] emphasises the importance of contextual cues gathered during information interaction being incorporated in results to aid personal information recollection. Rhodes [12] provides a conclusive overview and evaluation of ‘Just-In-Time Information Retrieval agents’ (JITIRs). JITIRs “proactively retrieve and present information based on a person’s local context in an accessible yet non-intrusive manner” in order to assist with ongoing activities. Whilst a number of such systems exist, many take a limited view of context for a specific task, such as word processing or web browsing, ultimately limiting their value [7]. Similarly, their usefulness may also be impaired by the lack of contextual cues when modelling complex human interaction [11]. As such, we propose a more generalised and extensible framework that applies a broader view of situational context to recommendation.

2 Foundations of Context

Within the literature, context is a broad concept that is used to address a number of often weakly related notions, making it a problematic and frequently contended subject. Abowd et al. [1] however provide a fundamental definition of contextual cues and their role in expressing a situation as:

“*Any* information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”

The perceived importance of context in many areas of computing science, most notably in human-computer interaction and information retrieval (IR)/natural language processing has resulted in significant research into the many facets of which context is believed to consist and their effect in the respective areas. Yet, whilst “most people tacitly understand what context is, they find it hard to elucidate” [5]. Considering that the definition of context is itself context-dependent [4], an examination of the information interaction scenario is needed to characterise the principle features. Ingwersen and Järvelin [9] consider the dimensions of this context within the “nested model of context stratification”.

Many perspectives on context exist, but [1] consider the aim of context-awareness is to understand the fundamental “who’s, where’s, when’s and what’s” which may together determine “why”. Considering these together as the user characteristics, locality, timing and interaction activity provides a concrete overview of the user’s ongoing information interaction situations. Many however debate this conveniently simplistic view of context. Dourish [6] comments on the flawed incompatibility between the ‘representational’ positivist and ‘interactional’ phenomenological views of context, noting convenient assumptions such as context being itself a “discrete form of information”, “delineable”, “stable” and that “context and activity are separable”. Consequently, he proposes a notion of context as an ‘interactional’ feature. With this, context is no longer a binary feature but one of relevance and irrelevance, arising from activity itself both dynamically and occasionally with situations defining the scope of contextual features. Whilst context is still very much the “who’s, where’s, when’s and what’s”, these further reflections provide new dimensions from which to understand perhaps the more interactive and user-centred, yet interdependent and intricate nature of context.

3 Modelling Context and Situation

Context can be modelled using a number of methods including key-value, markup scheme, graphical and object-oriented models as well as with logic and ontologies [2]. We use an object-oriented approach to define and model contextual cues and activity as it matches our design architecture. For each user-performed action and event across the desktop, email client and web browser, we record:

- Environmental context features: time (day/hour/morning, afternoon or evening), location (connected wireless networks) and physical features (plugged-in USB drives).
- Content-based user action features: any contained full-text such as email content, web page content etc is indexed (using Lucene.Net¹). Additionally, entities are extracted using LingPipe² and indexed.
- Implicit activity task sequence: the ongoing actions, and therefore proximity of actions to others in each separate session (identified by computer start-up/shut-down). Activity occurring in close proximity may be related as part of the same task.

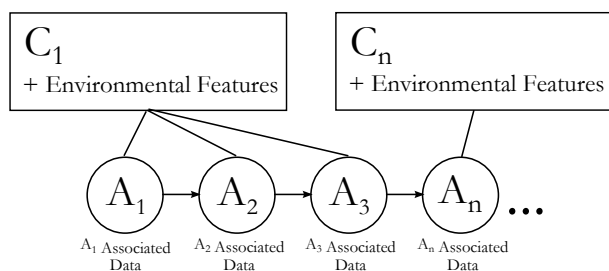


Fig. 1. Context and interaction as a sequential model

In Fig. 1, each user action A_n (where n is an incrementing sequence number) is logged along with any associated action data (such as full-text content or URLs etc). Each A_n has an associated model of the context C_n (where n identifies a unique but possibly shared context representation) within which it occurred. Each of the individual contextual features contained in C_n provides a usually weak indication of the activity likely to take place in that situation. The ability of these contextual clues in defining possible activity is realised when they are combined together, for example time and place, or short-term and long-term informational topics. Anecdotally, a user may primarily work with certain personal information objects during week days when in the office, yet use the same computer for leisure at the weekend. Similarly, when working on a project over a defined period of time, the topic of information being engaged is likely to be to some degree relevant to that of the project.

The central assumption we make is that user activity occurs within a number of transitioning situations. Each situation may be the effect of any number of combined contextual features, which in turn define to some degree the activity. As it would be practically impossible to fully delineate the complex sources of a situation we instead propose an approach that attempts to make the most likely recommendations of previously accessed information objects given the available contextual features considered in combination.

¹ <http://incubator.apache.org/projects/lucene.net.html>

² <http://alias-i.com/lingpipe/>

4 A Context-Aware Personal Recommendation Framework

The overall conceptual framework architecture (summarised in Fig. 2) was specified to be language-agnostic. We however implemented the PRF using the Microsoft .NET Framework allowing it to run on Microsoft Windows desktops for the purpose of our evaluation.

1. Capture of ongoing activity/contextual cues. Transparent to the user, a listener collects and logs interaction and changing contextual data in real-time. By using .NET we were able to use many existing tools and APIs for accessing application interaction and contextual cues. Email activity is recorded using a VSTO³ Microsoft Outlook plug-in. Windows shell activity is gathered using interfaces provided by Microsoft Research PersonalVibe⁴. Web browsing activity is monitored through a managed Microsoft Internet Explorer Browser Helper Object⁵.

2. Stored model of user's activity and context history. The model detailed in Section 3 is used to structure the past and current activity. All data is stored locally maintaining user privacy whilst allowing the recommenders to quickly search and filter the interaction history.

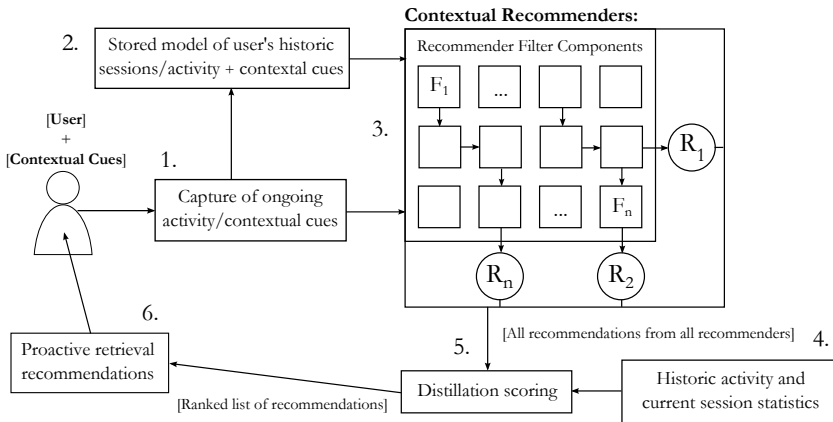


Fig. 2. Personal Recommendation Framework conceptual overview

3. Contextual recommenders. The PRF can contain one or more recommenders in operation at any time. Each recommender R comprises of a pipeline of single filter components F_n which individually filter past actions based on a basic concept of context (such as a dimension of time, location or content/entity

³ Visual Studio Tools for Office: <http://msdn.microsoft.com/en-gb/vsto/>

⁴ <http://research.microsoft.com/en-us/downloads/0ea12e49-8b29-4930-b380-a5a00872d229/>

⁵ <http://archive.msdn.microsoft.com/SpicIE/>

similarity), such that $R_n = (F_1, F_2 \dots F_n)$ (in an ordered arrangement). This reusability and configurability can be used to build complex user-specific recommenders that are effective for possible complex user contexts. For each A_n and C_n to occur, each R_n in the set of operational recommenders $R = \{R_1, R_2 \dots R_n\}$ can optionally provide a set of new arbitrarily scored (confidence level, s) potential action (A) recommendations $r_{R_n} = \{r_{1,A,s}, r_{2,A,s} \dots r_{n,A,s}\}$ based on it's limited view of the current context.

Whilst the PRF is in operation, adaptive feedback is used to set $R_{n,p}$ to prioritise recommenders that are making correct recommendations ($R_{n,p} = 1$ when the PRF first starts). Each R_n maintains history of it's previous three r_{R_n} , if the new A_n to occur exists in any of the last r_{R_n} , then $R_{n,p}$ is multiplicatively increased by 2. Alternatively, if A_n does not exist in any of the previous r_{R_n} , then $R_{n,p}$ is decreased by 1, with a floor of 1.

4. Historic activity collection and current session statistics. Statistics can be obtained from the historic activity and current session of the user and used in the distillation scoring function to determine most likely and useful recommendations to make.

Analysis of a 4 week history of user activity reveals a power law distribution of the frequency of unique actions (see Fig. 3). We hypothesise that the small number of actions that occur very frequently are not good recommendations as the user is likely to already have shortcuts to access them. Similarly, the large number of actions that occur very rarely are not likely to be good repeat action recommendations as the PRF may not have enough information to specify their context, and they may have been part of a one-off task. With the same intention as Luhn's hypothesis on the strong resolving power of medium frequency words [10], we therefore desire to positively bias the actions that have occurred a medium number of times using the *collectionWeighting()* function in distillation scoring. *collectionWeighting()* calculates the probability density function (PDF) from a normal distribution with σ and μ adjusted to weight the rank of the action as desired (the effect of changing σ is seen in Fig. 3). For our study we set $\mu = 0.2 * |Ranks|$ and $\sigma = 1.5 * (\frac{\mu}{2})$ so as to moderately bias the middle range of ranks.

Also used in the distillation scoring function is the *recentSameType()* function. The intention of this function is to count the number of actions of the recommendation type that have occurred in the past 20 user actions. For example, if a user is browsing the internet it would be preferable to recommend further internet browsing actions.

5. Distillation scoring. Scores and ranks the set of all recommendations offered by all operational recommenders to consider the most likely to recommend to the user. Multiple parameters are taken into consideration to provide an aggregate score such that the most likely and useful recommendations are highly weighted. Discussed previously, $R_{n,p}$ is the priority of the recommender that made the recommendation $r_{n,A,s}$. *recommendedCount()* is the number of times the same action A has been recommended in the set of all recommendations. *collectionWeighting()* and *recentSameType()* bias the action based

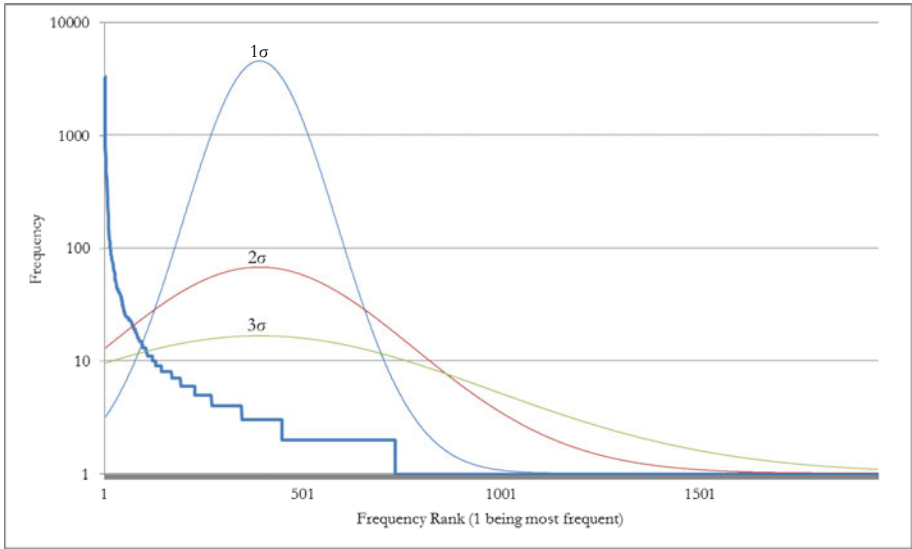


Fig. 3. Unique action power law and normal distribution PDF weighting at $\sigma = 1, 2, 3$ and $\mu = 0.2 * |Ranks|$ over a 4 week user history

on collection and session statistics. A_s is an arbitrary confidence level score provided by the recommender filter process, derived from the frequency of the action recommended amongst others in the recommender’s recommendation set.

$$Score(r_{n,A,s}) = R_{n,p} * recommendedCount(r_{n,A}) * collectionWeighting(r_{n,A}) * recentSameType(A) * A_s$$

6. Proactive retrieval recommendations. For evaluation, we consider the most likely final recommendations to be within the top 8 ranked recommendations. We feel that the need to present any more than 8 recommendations would be counter-productive to a user. For the purpose of this initial study we did not directly present recommendations to the user and so did not need to implement any interface for this task.

5 Evaluation Methodology

Performing a user-study with a fully working prototype would have been expensive and problematic at this early stage [8]. Given this, we chose to develop a hybrid evaluation approach based on the methodology of the ‘Cranfield Experiments’/TREC-style [13] (thus achieving repeatability, scalability and measurability). As there is no pre-existing collection to suit our application, we instead collect real user interaction data for the test collections and the sequential timing of the contained user actions to provide implied relevance judgements. As such, our approach can be summarised as:

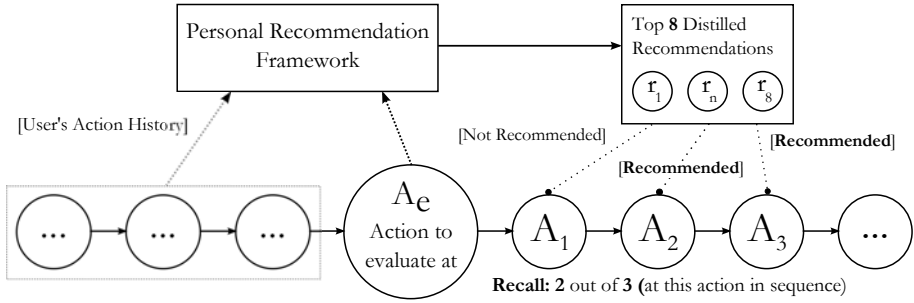


Fig. 4. Evaluation procedure over the user's 2 week past activity collection U_A

1. Recruit user and elicit user's interaction habits through a questionnaire and interview.
2. Install PRF software on user's desktop computer. All data is stored locally.
3. After 2 weeks, the evaluation application is run locally and performs recommendations using different PRF configurations (see Section 5.1) by simulating the 2 weeks of activity again using the logged user actions.
4. Non-identifying recall statistics for each configuration are returned for analysis.

Whilst this approach cannot provide feedback on whether a system is indeed perceived as effective by users in a complex interactive environment (for which only a user-study could provide evidence), it does provide early measures of the degree to which a recommender system is proactively making recommendations of upcoming activity under different conditions.

5.1 Evaluation Metric

Traditional IR recall and precision measures are not directly relevant to this system. We however consider two measures derived from recall in the context of the PRF. Fig. 4 demonstrates our algorithm for calculating individual recall (between 0 and 3) at each user action and context A_e in the user's 2 week history U_A . For each logged action, we simulate the interaction and context again, allowing the PRF to make recommendations that can then be compared with known subsequent user activity. As an overall comparable measure of recall we define R_{ev} , calculated as $\frac{A_r}{A_{ev}}$, where A_r is the number of actions correctly recommended during evaluation and A_{ev} is the number of subsequent actions evaluated against ($\approx 3|U_A|$ due to end of usage sessions in the log).

6 Experimental Settings

Our experimental approach evaluated the effect of 3 variables: user, PRF recommenders and PRF distillation scoring function. We measured the effectiveness

of three PRF configurations: Config 1. All Recommenders + Default Distillation Scoring (including A_s), Config. 2. All Recommenders + Alternative Distillation (excluding A_s) and Config. 3. Environmental Recommenders only + Default Distillation Function) over a 2 week period for 4 separate users.

User characteristics. Although we cannot quantify the scenarios that each of the recruited users work within, we can note a few of their personal characteristics that may have an effect on performance outcome. User 1 is a student, male, age 23 using a laptop. User 2 is a student, male, age 21 using a desktop computer. User 3 is a student, male, age 22 using a desktop computer. User 4 is a retired/part-time home-based worker, male, age 64 using a desktop computer.

7 Results and Discussion

For users 1 to 4 we logged 2007, 895, 848 and 1907 separate actions U_A respectively over the 2 week study period. For each user we calculated overall recall R_{ev} against a possible 5790, 2596, 2376 and 5165 subsequent actions A_{ev} .

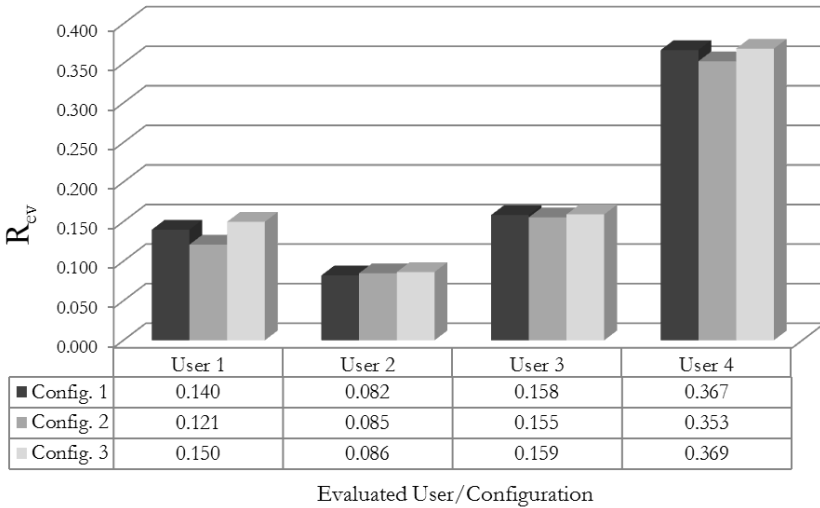


Fig. 5. Overall recall R_{ev} for all users over 3 PRF configurations

Overall performance. In Fig. 5, overall performance measured by R_{ev} is shown by user and PRF configuration. Of interest is the significantly greater R_{ev} obtained for User 4 over all configurations. After informal interview with User 4 it was determined that their work environment is of a relatively static nature with a great deal of repetition in day-to-day tasks. In comparison, User 2 with a much reduced R_{ev} considers their activity to be largely exploratory. Examining

PRF configuration effectiveness would suggest that Config. 3 is marginally more effective than other configurations. This is somewhat surprising, suggesting existing content-based recommenders are adding little, if any R_{ev} improvement. Considering that many content-based context recommendation techniques have been applied successfully in previous systems, it could be concluded that current PRF content-based recommendation implementations are not effective.

Performance over time. Fig. 6 demonstrates the significant variation in individual recommendation recall effectiveness over time. During certain periods of activity the effectiveness peaks, representing periods of repeat activity. This could for example be continuing working with an existing document and analysing known information that has already been gathered and stored, for which the PRF has existing knowledge and so can make effective recommendations. Alternatively, troughs in the effectiveness indicate that the user is performing activity that is unknown and therefore the system cannot assist effectively. The ‘cold-start problem’ of not being able to make recommendations without established knowledge is especially apparent at the start of the 2 weeks (considering the graph shows a forward moving average).



Fig. 6. Smoothed 100-point forward moving average of individual recall at A_e over a two week period for User 1 with PRF Config. 1

Performance over time, varying by configuration. Fig. 7 displays the effectiveness of PRF configurations 1, 2 and 3 plotted alongside each other to see the variance in performance between configurations over time. It is apparent that different configurations are optimal at different times during the evaluation.

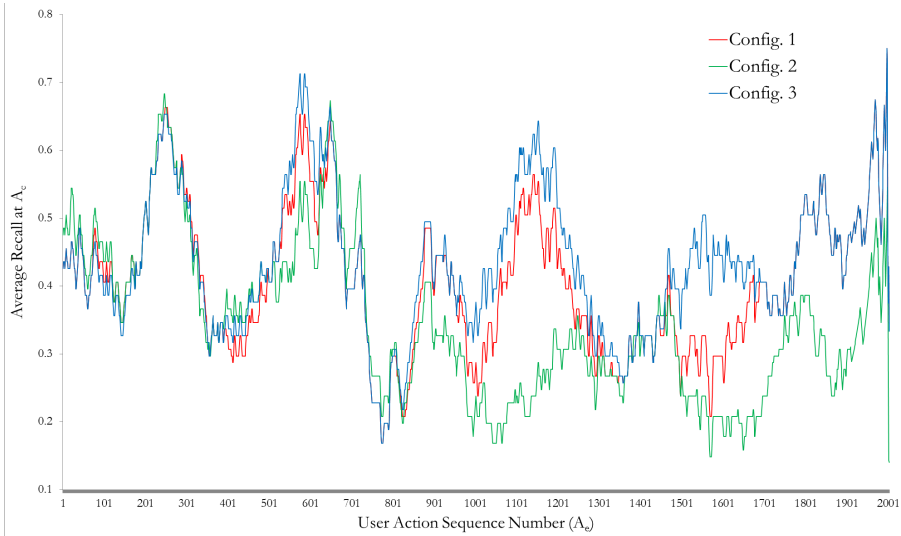


Fig. 7. Smoothed 100-point forward moving average of individual recall at A_e over a two week period for User 1 for all PRF configurations

Configuration 2 is initially optimal, then Configurations 1 and 3 become optimal at varying intervals.

7.1 Discussion

Preliminary results indicate that the recommender framework is making some relevant recommendations. However, given the real-life nature of experiments it is difficult to accurately predict user information needs. User 1 achieved the highest recall, likely due to their work being more routine and so recommendable in comparison to User 2's less defined tasks. R_{ev} provides only an average of overall recall whereas evaluation sequence analysis suggests significant variation in recall (as new and previously seen situations occurred) over all configurations for all users. This analysis strongly supports the need for adaptivity in both recommenders and distillation scoring for varying user characteristics, as well as during a user's ongoing interaction to maximise the effectiveness of any such system throughout time.

8 Conclusion and Future Work

In this work we proposed and implemented an extensible approach to modelling context and information interaction. The objective of our PRF system was to provide recommendations of personal information objects that are mostly likely to be in context with the current situation of the user, thus reducing the need for the user to explicitly express their information needs through queries. Using this system, our study explored the potential for modelling user context and activity and using it for situation-sensitive recommendations. Whilst our proposed

evaluation approach provided a relatively low-cost, confidential and controlled environment to obtain performance data, further investigation is needed to determine the user-perceived effectiveness of such recommendations. Users were highly aware of our access to their personal data and were only willing to participate on condition of their confidentiality, supported by our local evaluation tool. Only a user-based investigation could determine effectiveness and utility in real-use scenarios. As studied more extensively in other work, we consider time and location combined to be significant contextual indicators of the user's information needs. Result analysis strongly supports the need for adaptivity in the PRF to provide optimal effectiveness and efficiency at all times. An automated or semi-automated algorithm with explicit user feedback could be developed to ensure the PRF is using recommenders and a distillation scoring function best-suited to the user and their changing task characteristics.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2, 263–277 (2007)
3. Bauer, T., Leake, D.: A research agent architecture for real time data collection and analysis. In: Proceedings of the Workshop on Infrastructure for Agents, MAS, and Scalable MAS, pp. 61–66 (2001)
4. Demiris, T.: Context revisited: a brief survey of research in context aware multimedia systems. In: Proceedings of the 3rd International Conference on Mobile Multimedia Communications, *MobiMedia 2007*, pp. 66:1–66:5. ICST, Brussels (2007)
5. Dey, A.K.: Understanding and using context. *Personal Ubiquitous Comput.* 5, 4–7 (2001)
6. Dourish, P.: What we talk about when we talk about context. *Personal Ubiquitous Comput.* 8 (2004)
7. Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., Robbins, D.C.: Stuff i've seen: A system for personal information retrieval and re-use. In: *SIGIR 2003 Proceedings*, pp. 72–79. ACM, NY (2003)
8. Elswailer, D., Ruthven, I.: Towards task-based personal information management evaluations. In: *SIGIR 2007 Proceedings*, pp. 23–30. ACM, NY (2007)
9. Ingwersen, P., Järvelin, K.: *The Turn: Integration of Information Seeking and Retrieval in Context*. The Information Retrieval Series. Springer-Verlag New York, Inc., Secaucus (2005)
10. Luhn, H.P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.* 2, 159–165 (1958)
11. Olsson, C., Henfridsson, O.: Designing context-aware interaction: An action research study. In: Srensen, C., Yoo, Y., Lyytinen, K., DeGross, J. (eds.) *Designing Ubiquitous Information Environments: Socio-Technical Issues and Challenges*, vol. 185, pp. 233–247. Springer, Boston (2005)
12. Rhodes, B.J.: *Just-in-Time Information Retrieval*. PhD thesis, Massachusetts Institute of Technology, Supervisor-Maes, Pattie (2000)
13. Voorhees, E.M., Harman, D.K.: *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing, MIT Press (September 2005)

Folksonomy Query Suggestion via Users' Search Intent Prediction

Chiraz Trabelsi, Bilel Moulahi, and Sadok Ben Yahia

Faculty of Sciences of Tunis, University Tunis El-Manar
Campus Universitaire, 1060 Tunis, Tunisia
{chiraz.trabelsi,sadok.benyahia}@fst.rnu.tn

Abstract. Recently, social bookmarking systems have received surging an increasing attention in both academic and industrial communities. The main thrust of these Web 2.0 systems is their easy use that relies on simple intuitive process, allowing their users to label diverse resources with freely chosen keywords *aka* tags. The obtained collection are known under the nickname Folksonomy. As these systems grow larger, however, the users address the need of enhanced search facilities. Today, full-text search is supported, but the results are usually simply listed decreasingly by their upload date. Challenging research issue is therefore the development of suitable prediction framework to support users in effectively retrieving the resources matching their real search intents. The primary focus of this paper is to propose a new users' search intent prediction approach for query tag suggestion. Specifically, we adopted Hidden Markov Models and triadic concept analysis to predict users' search intents in a *folksonomy*. Carried out experiments emphasize the relevance of our proposal and open many issues.

Keywords: Hidden Markov Models, *folksonomies*, Search intent, Triadic concepts analysis.

1 Introduction

Complementing the Semantic Web effort, a new breed of so-called Web 2.0 applications recently emerged on the Web. Indeed, social bookmarking systems, such as *e.g.*, DELICIOUS¹, BIBSONOMY² or FLICKR³ have become the predominant form of content categorization of the Web 2.0 age. The main thrust of these Web 2.0 systems stands in their easy use that relies on simple, straightforward structures by allowing their users to label diverse resources with freely chosen keywords *aka* tags. The resulting structures are called *folksonomies*⁴, that is, "taxonomies" created by the "folk". Considered as a tripartite hyper-graph of tags, users and resources [12], the new data of *folksonomy* systems have become

¹ <http://www.delicious.com>

² <http://www.bibsonomy.org>

³ <http://www.flickr.com>

⁴ <http://www.vanderwal.net/folksonomy.html>

an invaluable source for information retrieval (IR) [13]. Indeed, one of the main services provided by social tagging systems is searching. Searching occurs when the user enters a tag as a query and a, ranked by relevance, list of related resources are yielded to the user. Even though collaborative tagging applications have many benefits, they also present some thriving challenges for Information Retrieval (IR). Actually, the core of many search engines is the ranking algorithm. However, the most currently used ranking algorithms are not straightforwardly adaptable to *folksonomies*. Furthermore, these traditional tools for web information retrieval constitute a hindrance, since they do not take neither social or behavioral facts into account in the retrieval task of resources nor intercepting user's information needs.

A common fact in *folksonomy* search is that a user often performs multiple iterations of query refinement to catch the desired results from a *folksonomy* search engine. Indeed, there is no "standard" or "optimal" way to issue queries to a *folksonomy* search engine, and it is well recognized that query formulation is a bottleneck issue in the usability of search engines. Query suggestion⁵ is thus a promising direction for improving the usability of *folksonomy* search engines. The explicit task of query recommendation is to help users formulate queries that better represent their search intent during *folksonomy* search interactions.

Example 1. (SEARCH INTENT AND CONTEXT). Suppose that a user submits a query "java", then it seems to be hard to determine the user's search intent, *i.e.*, whether the user is interested in the java island, java programming language, or the java song. Hence, without looking at the context of search, the existing methods often suggest many queries for various possible intents, and thus may have a low accuracy in query suggestion. Therefore, if we find that a community of users have submitted a query "Indonesia" before "java", then it is very likely that the user search intent is the Java island of Indonesia. Moreover, we can predict, the most-probable next queries of the current user. Therefore, the query context which consists of the recent queries issued by a community of users, sharing the same interest for a particular topic as the current user, can help to better understand the user's search intent and enable us to make more accurate queries suggestions.

Hence, in this paper, we introduce a novel users' search intent prediction approach to address two major challenges facing information retrieval in *folksonomies*: (i) Discovering and modeling users' search intents in a *folksonomy*; and (ii) Predicting users' next queries by suggesting alternatives queries and recommending a set of resources that fulfills users' information needs concisely and accurately.

Therefore, to tackle this challenging task, we firstly propose, to define a user search intent as a subset of *folksonomy* users who implicitly agree (on subset of resources) on a common conceptualization. From a data mining perspective, the discovery of shared conceptualizations opens a new research field which may prove interesting also outside the *folksonomy* domain: "Closed itemset mining in

⁵ We use "query suggestion" to refer to "*folksonomy* query suggestion".

triadic data”, which is located on the confluence of the research areas of Association Rule Mining and Formal Concept Analysis [7]. Indeed, in *folksonomies*, the usage of tags of users with similar interests tends to converge to a shared vocabulary. To this end, we use an algorithm, called TRIAS, to mine these users’ search intents from a *folksonomy*.

On the other hand, as well recognized by many previous studies, a user often raises multiple queries and conducts multiple rounds of interaction with a search engine to fulfill an information need. For instance, a user u plans to buy a computer, he may decompose his general search task, comparing various computers, into several specification sub-tasks, such as searching the Dell’s computers. In each sub-task, u may bear a particular search intent in mind and formulate a query q to describe the intent. Further, u may selectively choose some related resources to consult. Actually, a Hidden Markov Models (HMM for short) naturally describe such a search process.

Hence, we propose to model each search intent as a state of the HMM, and consider the query and accessed resources as observations generated by the state. The whole search process can be then modeled as a sequence of (auto)-transitions between states.

The remainder of the paper is organized as follows. Section 2 thoroughly scrutinizes the related work. We describe later in Section 3 our probabilistic approach for users’ search intents prediction composed of two major steps, namely, the model-learning step and Matching & prediction step. We dedicate Section 3 for underpinning, through an illustrative example based on a sample taken from a real dataset, the different steps of our approach. The experimental study of our approach is illustrated in Section 4. Section 5 concludes this paper and sketches avenues for future work.

2 Related Work

Folksonomies have recently been studied for their snugness connection with the information retrieval field. The most complete analysis of the DELICIOUS bookmarking system is provided by Heymann et al., [6]. In this respect, the authors investigated the potential benefits of collaborative bookmarking for the web search.

Although search and tagging are apparently for different purposes, the search aims at finding existing information while tagging is to create new information, they are actually very closely related, and can be regarded as two activities governed by the same common information preferences in a user’s mind. Indeed, from comparing categories of tags with query logs and user study, authors in [2,3,9,11] argued that most of the tags can be used for search, and that user’s tagging behavior closely reflects *folksonomy* searching behavior. For example, if a user assigned the tag ”java” to the Apache Lucene homepage (<http://lucene.apache.org>), then we assume that the user will consider this web page as relevant if he issues ”java” as a query.

Hence, defined as a set of assignments, *i.e.*, triples (Resources, Users, Tags), *folksonomies* can be seen as the other side of the ”medal”, *i.e.*, the log files,

baptized by [10] under the name *logsonomies*. As logdata contains queries, clicks and session IDs, the classical dimensions of a *folksonomy* can be reflected: Queries or query words represent tags, session IDs correspond to users identifiers, and the URLs clicked, *i.e.*, resources accessed, by users can be considered as the resources that they tagged with the query words. Currently, to the best of our knowledge, there is no known prior work which has exploited these *logsonomies* for query suggestion in *folksonomy* search.

In the context of users' search intents prediction, HMMs have been recently used for web search [4,5]. Indeed, Cao *et al.*, [4] have developed the notion of variable length hidden markov model (vHMM) and applied it to model query contexts from search session extracted from web log data. Whereas, in [5], the authors proposed a novel sequential query prediction approach that tries to grasp a user's search intent based on his past query sequence mined from massive search engine logs. Because it ignored the three dimensional relationship among users, resources, and queries, the users' tagging behavior was not accurately profiled, and thus the suggestion quality based on the query and the resource data is not satisfactory.

Hence, regardless of their inadaptability to *folksonomies* search, our work has one fundamental difference from these previous HMMs session-based approaches since *logsonomies* provide a three-dimensional dataset (users, queries and resources) instead of a usual two-dimensional web log data (queries and resources). Hence, while previous work focusses on query-resources relations, for HMM modeling, by applying a clustering algorithm, we instead make use of the triadic concept structure which stresses users-queries-resources correlations [7]. Indeed, the triadic concept structure describes three types of sets: (i) the set T of related tags; (ii) the set U of the associated users, *i.e.*, users who have tagged by T ; and (iii) the set of related resources, *i.e.*, which were assigned with T by users U . Hence, triadic concepts allow grouping semantically related tags taking into account the Users' tagging behavior in a *folksonomy*. In fact, in [7,8], the authors have considered a *folksonomy* as a triadic context and proposed an algorithm called TRIAS to get out implicit shared conceptualizations formally sketched by triadic concepts.

In the following, we introduce a novel approach for predicting users' search intents in a *folksonomy* with the following salient features:

- **Folksonomy vs Logsonomy:** Based on the duality hypothesis of search and tagging, two important behaviors of web users [2,3,9,10,11], we make use of tag assignments for learning user behavior and improving search engine accuracy. Thus, defined as a set of tag assignments, *i.e.*, triples (resources, users, tags), *folksonomy* is represented as a search log data, *i.e.*, *logsonomy*.
- **Use of the semantic relatedness embodied in the different frequencies of co-occurrences among users, resources and tags in the *folksonomy*:** instead of using probabilistic models or network analysis techniques for grouping related tags, we mine frequent triadic concepts [7]. Indeed, triadic concepts can be used as an access structure for providing important hidden correlations between queries, resources and users.

- **A Hidden Markov Model (HMM) for learning and then predicting users’ search intents in a *folksonomy*:** On the contrary of the surveyed approaches which consider a 2-dimensional pair relations {query, resource}, missing by the way a part of the total interaction between the three dimensions, *i.e.*, user, query and resource, we introduce a unified framework to concurrently model the three dimensions handled by a HMM [14].

3 Hidden Markov Models for Users’ Search Intent Prediction

Our proposed users’ search intent prediction approach for query suggestion has two steps: **Model learning** and **User search intent prediction**. First of all, for a given *logsonomy*, we proceed by constructing our HMM model. Hence, considering that, in a HMM there are two types of states: the observable states and the hidden ones [14], thereby, we extract user’s queries sequences SL_i issued by each user idu_i for defining the observable states in the HMM, whereas, we mine users’ search intents IT_1 formally represented by triadic concepts for modeling the hidden states.

Thereafter, the results of these previous stages, will be used for the HMM training. Secondly, once the model learning step is performed, we proceed with the user search intent prediction step for identifying the query context and then predict the next user query according to the next HMM state. In the following, we describe these two steps in more details.

3.1 The Model-Learning Step

As we previously mentioned, the model learning step consists of three stages: **User’s queries sequences extraction**, **User search intent mining**, and **HMM training**. Let us firstly start by presenting a simplified definition of a *logsonomy* [10]:

Definition 1. (LOGSONOMY) *A logsonomy \mathcal{L} , related to a folksonomy \mathcal{F} , is a set of tuples $\mathcal{L} = (\mathcal{ID}, \mathcal{Q}, \mathcal{RS}, \mathcal{G})$, where $\mathcal{G} \subseteq \mathcal{ID} \times \mathcal{Q} \times \mathcal{RS}$ represents a triadic relation and each $g \in \mathcal{G}$ can be represented by a triplet: $g = \{(id, q, rs) \mid id \in \mathcal{ID}, q \in \mathcal{Q}, rs \in \mathcal{RS}\}$. Roughly speaking, a user identified by id , retrieved the resource rs queried by the query q .*

Stage 1: User’s queries sequences extraction: In this step, we are interested in extracting the sequences of queries SL_i issued by each user idu_i from users sessions S_i . Hence, we proceed firstly by collecting, from the given *logsonomy*, the users sessions S_i . Thereby, we define a user session as followings.

Definition 2. (USER SESSION)

A user session S_i , related to a user idu_i , is defined as:

$$S_i := \{\{User\ queries\ } q_{S_i,p}\}, rs_{S_i,j}\}.$$

Table 1. Users' sessions example

$S_1 := \{\{q_{1,1}, q_{1,2}, q_{1,3}\}, rs_{1,1}\}; \{\{q_{1,1}, q_{1,4}\}, rs_{1,2}\}$
$S_2 := \{\{q_{2,3}, q_{2,4}\}, rs_{2,3}\}$
$S_3 := \{\{q_{3,2}, q_{3,3}\}, rs_{3,4}\}; \{\{q_{3,4}, q_{3,5}\}, rs_{3,6}\}$

with $rs_{S_i,j} :=$ The resource j accessed by the user idu_i within the session S_i ,
 $q_{S_i,p} :=$ The p^{th} submitted query in S_i .

Table 1 illustrates an example of users sessions. For example, the user session S_2 , highlights that the user idu_2 has retrieved the resource $rs_{2,3}$ after submitting the two queries $q_{2,3}$ and $q_{2,4}$.

Once the users sessions are collected, we extract user's queries sequences by keeping, for each user, the sequences of queries related to his session and discard useless information. An example of user's queries sequences associated to the Table 1 is given in the following: $SL_1 : ((q_{1,1} \Rightarrow q_{1,2} \Rightarrow q_{1,3}); (q_{1,1} \Rightarrow q_{1,4}))$, $SL_2 : (q_{2,3} \Rightarrow q_{2,4})$, $SL_3 : ((q_{3,2} \Rightarrow q_{3,3}); (q_{3,4} \Rightarrow q_{3,5}))$ where SL_i , describes query sequences of the user idu_i .

Stage 2: User search intent mining: The second step of the model-learning step is to mine users' search intents from the *logsonomy*. Hence, since, at one hand, different users may submit different queries to describe the same search intent and on the other hand, different users sharing the same interest for a specific topic, may retrieve different resources even if they submit exactly the same query, therefore we define a user search intent as the common interest shared by a community of users U for a retrieved set of resources R' queried by a certain set of queries Q' .

Consequently, a search intent can be, formally, represented, in a *logsonomy* $\mathcal{L} = (\mathcal{ID}, \mathcal{Q}, \mathcal{RS}, \mathcal{G})$, as a triadic concept $\mathcal{IT} = (U', T', R')$ where $U' \subseteq \mathcal{ID}$, $T' \subseteq \mathcal{Q}$, and $R' \subseteq \mathcal{RS}$ with $U' \times T' \times R' \subseteq \mathcal{G}$. Indeed, mining triadic concepts to discover and model users' search intents, allows to address the sparseness of queries and interpret users' information needs more accurately. The users' search intents are therefore obtained by applying the TRIAS algorithm [7] on the logsonomy \mathcal{L} . TRIAS takes as input the *logsonomy* \mathcal{L} as well as three user-defined thresholds: *id - minsupp*, *q - minsupp* and *rs - minsupp* and outputs the set of all frequent tri-concepts, *i.e.*, search intents, that fulfill these aforementioned thresholds. Roughly speaking, for example, the search intent $IT_1 = \{(id_1, id_3, id_4), (q_4, q_5), (rs_1, rs_2)\}$ highlights that the community of users (id_1, id_3, id_4) share the same interest in the resources (rs_1, rs_2) queried by q_4 and q_5 .

Given the user's queries sequences and the users' search intents, previously extracted, we proceed in the next section with the HMM training .

Stage 3: HMM training: For the last step of our approach, we are interested in training the HMM. Therefore, given the set of hidden states $S = \{s_1, \dots, s_{ns}\}$, we denote the set of distinct queries as $\mathcal{Q} = \{q_1, \dots, q_{nq}\}$, the set of accessed resources

$\mathcal{RS}s = \{rs_1, \dots, rs_{nrs}\}$ and a set of user \mathcal{ID} ; $\mathcal{ID}us = \{idu_1, \dots, idu_{nidu}\}$, where ns is the number of states of the model, nq is the total number of queries, nrs is the total number of resources, $nidu$ is number of users, and SL_i is a state sequence. Our HMM noted $\lambda = (A, B, B', \pi)$, is a probabilistic model defined as follows:

- $\pi = [\dots \pi_i \dots]$, the initial state probability, where $\pi_i = P(s_i)$ is the probability that a state s_i occurs as the first element of a state sequence SL_i .
- $B = [\dots b_j(q) \dots]$, the query emission probability distribution, where $b_j(q) = P(q | s_j)$, denotes the probability that a user, currently at a state s_j , submits a query q .
- $B' = [\dots b_k(rs) \dots]$, the resource emission probability distribution, where $b_k(rs) = P(rs | s_k)$, denotes the probability that a user, currently at a state s_j , accesses the resource rs .
- $A = [\dots a_{ij} \dots]$, the transition probability, where $a_{ij} = P(s_j | s_i)$ that represents the transition probability from a state s_i to another one s_j .

Once the HMM is formalized, we proceed with learning its parameters (A, B, B', π) from a *logsonomy*. This is done by computing the four sets of the HMM parameters: the initial state probabilities $\{P(s_i)\}$, the query emission probabilities $\{P(q_t | s)\}$, the resource emission probabilities $\{P(rs | s_k)\}$, and the transition probabilities $\{P(s_j | s_i)\}$. Hence, inspired from [4], we compute these sets as followings:

1. $\pi_i = P(s_i) = \frac{|\varphi(s_j)|}{|SL_c|}$ with:
 - $SL_c = \cup_{i \in 1, \dots, t} \{E_i\}$ = total set of candidate states sequences to which could be matched a sequence of queries where E_i denotes the set of candidate states that could match a query from a given sequence of queries.
 - $\varphi(s_j)$ = set of states sequences in SL_c starting from s_j .
2. $b_j(q) = P(q | s_j) = \frac{\sum_{rs \in \mathcal{RS}_j} Count(rs, q)}{\sum_{q \in \mathcal{Q}_j} \sum_{rs \in \mathcal{RS}_j} Count(rs, q)}$.
3. $b_k(rs) = P(rs | s_k) = \frac{\sum_{q \in \mathcal{Q}_k} Count(rs, q)}{\sum_{q \in \mathcal{Q}_k} \sum_{rs \in \mathcal{RS}_k} Count(rs, q)}$. where $Count(rs, q)$ = number of times the resource rs is accessed as an answer to the query q in the *logsonomy*.
4. $a_{i,j} = P(s_j | s_i) = \frac{CS(s_i, s_j)}{NC}$ with:
 - NC = the number of occurrences of s_j in SL_c .
 - $CS(s_i, s_j)$ = the number of times the state s_i is followed by the state s_j in SL_c .

3.2 User Search Intent Prediction

Once the model learning step is performed, we proceed with the user search intent prediction step for identifying the query context and then predict the next user query according to the next HMM state. Indeed, when a user submits a query

q , two consecutive stages are carried out: (i) Matching the current user query to its corresponding context according to HMM states; and then (ii) Predicting the next HMM state which represents the user's search intent. Hence, the prediction process starts by looking for the most likely HMM state s_{MS} to which q could better belong. This is done by computing, for each HMM state, the value of the quantity $Mat_i = \pi_i \times b_i(q)$, where π_i is the initial probability of the state s_i and $b_i(q)$ is the emission probability of q at s_i . Therefore, the state with the highest value of Mat_i will define the context of q .

Thereafter, the prediction of the user's search intent is then performed, by looking for the next state s_{NextMS} of s_{MS} . This is obtained by computing the index value $NextMS$ as follows : $NextMS = argmax_j \{a_{\{MS,j\}} \times b_j(q)\}$, where q denotes a query belonging to the state s_j , successor of s_{MS} in the HMM.

Thus, the state s_{NextMS} represents the most probable search intent to which the user may transit after submitting the query q .

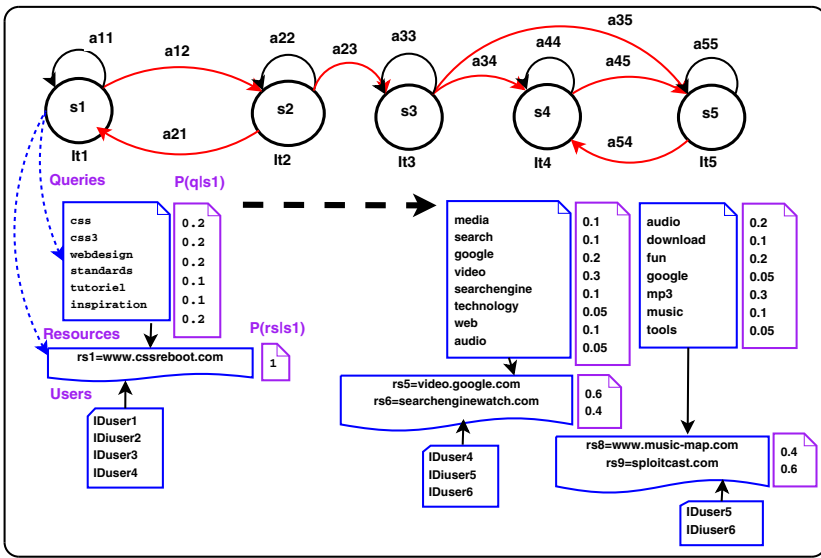


Fig. 1. An example of the proposed approach on a real dataset

Example 2. Fig. 1 represents a HMM with five states $\{s_1, s_2, s_3, s_4, s_5\}$ where each state denotes a user search intent, i.e., It_1, It_2, It_3, It_4 and It_5 , extracted by the algorithm TRIAS from a small real test data collected from DELICIOUS.US⁶. Each search intent is represented by a triplet, i.e., the set of all queries frequently used by a set of users looking for a set of resources. The corresponding transition matrix A, and the distributions of the different probabilities of observation (of resources and queries) are obtained by computing probabilities as described in the HMM training stage. The corresponding HMM, with five states, is shown in

⁶ <http://www.delicious.com>

Fig. 11 Suppose that the generated HMM with five states $\{s_1, s_2, s_3, s_4, s_5\}$ has a transition probability matrix as follows:

$$A = \begin{pmatrix} 0,4 & 0,6 & 0,0 & 0,0 & 0,0 \\ 0,2 & 0,5 & 0,3 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,3 & 0,2 & 0,5 \\ 0,0 & 0,0 & 0,0 & 0,3 & 0,7 \\ 0,0 & 0,0 & 0,0 & 0,6 & 0,4 \end{pmatrix}$$

And let us assume that $\pi = (0,2 \ 0,2 \ 0,2 \ 0,2 \ 0,2)$. Hence, considering the search intent represented by the state s_1 , users have a probability of 0,4 to keep the same search intent and a probability of 0,6 to skip for a new search intent represented by the state s_2 . For example, if a user submits the query "audio", then the prediction process starts by looking for the most likely HMM state to which the query "audio" could better belong. This is obtained by computing for each of the five states, the quantity $Mat_i = \pi_i \times b_i(audio)$ including :

$$Mat_1 = \pi_1 \times b_1(audio) = 0,2 \times 0 = 0; \quad Mat_2 = Mat_3 = 0; \quad Mat_4 = \pi_4 \times b_4(audio) = 0,2 \times 0,05 = 0,01 \text{ and } Mat_5 = \pi_5 \times b_5(audio) = 0,2 \times 0,2 = \mathbf{0,04}.$$

Consequently, s_5 is the state which has the highest probability to represent the user's search intent for the query "audio". Possible states transitions from s_5 are either s_4 or s_5 (i.e., a user may keep the same search intent). Thus, the corresponding candidate queries to be predicted, after the "audio"'s query submission, are computed by the following formula, $argmax_j \{a_{5,j} \times b_j(q)\}$ with $j \in \{4, 5\}$ (i.e., possible state transition from s_5) and q is a query belonging to the search intents represented by s_4 or s_5 states. Furthermore, both the candidate resources (*rs8:www.music-map.com/*) and (*rs9:www.splottcast.com/*), with the respective probabilities 0,6 and 0,4, are recommended to the user.

Otherwise, given that on the one hand : $Max(b_5(q)) = 0,3$ and $Max(b_4(q)) = 0,3$ for all queries q in the fifth and the fourth state respectively, and on the other hand $argmax_j \{a_{5,5} \times b_5(q), a_{5,4} \times b_4(q)\} = argmax_j \{0,12; 0,36\} = 4$, then the search intent to be predicted is represented by the state of index 4 (i.e., s_4). Thus, queries {"video", "media", "google",...} belonging to the search intent represented by s_4 will be suggested to the user in an increasing ranked list of probability. Likewise, the corresponding resources of the considered search intent could be recommended in the same way.

4 Experimental Evaluation

The evaluation of all folksonomies's query suggestion systems is still an open challenge. In fact, as an evidence of the lack of social bookmarking systems that exploit search intents prediction, as far as we know, there is no work with topic published via the scholarly literature. Hence, the evaluation of our approach is a complex task. In order to analyze the accuracy of our approach we adopted the common evaluation measures, namely Precision and Recall [1].

We carried out our experiments on a dataset collected from a real-world social tagging system, *i.e.*, DEL.ICIO.US. The related *logsonomy*⁷, used to conduct our experiments, is around 10 MB in size (compressed) and are freely downloadable⁸ where tags represent queries, users identifiers correspond to session *IDs*, and resources that users tagged with the query words are considered as accessed resources.

4.1 Baseline Models

To the best of our knowledge, search intent prediction (using HMMs) in such social bookmarking systems have never been modeled before. Thus, for enhancing the effectiveness of our approach, we have selected two baselines models for query recommendation. The most popular queries recommender which suggests queries according to their global occurrence in the training data. On the other hand, the most popular query aware recommender, which ranks queries according to their global co-occurrence in the training set, with the query tag in the test set. For each of the algorithms of our evaluation, we briefly describe in the following the specific settings used to run them.

- **Most popular queries recommender:** For each tag query, we counted in how many user sessions it occurs and used the top queries (ranked by occurrence count) as recommendations.
- **Most popular query aware recommender:** These recommenders weight tag queries by their co-occurrence with a given query. We then used the most co-occurrent tag queries as a suggestion.

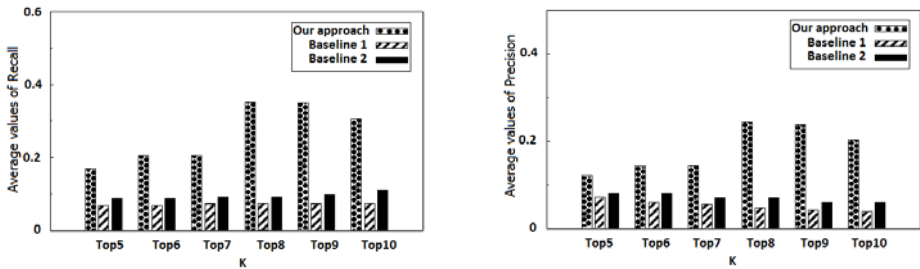


Fig. 2. Left: Averages of *Recall* on the DEL.ICI.OUS dataset; Right: Averages of *Precision* on the DEL.ICI.OUS dataset

4.2 Effectiveness of Our Approach

We evaluate the performance of the proposed approach on query prediction using a supervised learning method. Specifically, we randomly split the dataset, *i.e.*, the *logsonomy*, into a training set and a test set. Hence, for a given sequence

⁷ The *logsonomy* data set contains 99989 triples sessions, *i.e.*, assignments, 18066 queries, 53397 resources and 43419 users.

⁸ <http://data.dai-labor.de/corpus/delicious/>

of queries, the first n queries are used for generating predictions, whereas, the remaining part Q_T of the queries is considered as the set of queries actually formulated by the user, as the ground truth. The performance is then assessed by the measures of *recall* and *precision*. Hence, suppose that for a user query q_T , the proposed approach predicts a list of queries Q_R , thus, the measures of *recall* and *precision* are given as follows:

$$Recall = \frac{|Q_R \cap \{Q_T \setminus q_T\}|}{|Q_T \setminus q_T|}, \quad Precision = \frac{|Q_R \cap \{Q_T \setminus q_T\}|}{|Q_R|}$$

We report in the following results averaged over all user sessions and 6 test runs.

Figure 2 (Left), depicts averages of *recall* for different values of K , *i.e.*, the number of predicted queries, ranging from 5 to 10. Thus, according to the sketched histograms, we can point out that our approach outperforms both baselines. In fact, as expected, the *Recall* values of the individual baselines are much lower than those achieved by our approach. Furthermore, the average *Recall* on the *logsonomy* achieves high percentage for higher value of K . Indeed, for $K = 9$, the average *Recall* is equal to 0,351, showing a drop of 51,85% compared to the average *Recall* for $K = 5$. In this case, for a higher value of K , *i.e.*, $K = 9$, by matching current user's queries with their corresponding contexts, the proposed approach can produce all of the queries that are likely to be formulated by the user.

In addition, according to Figure 2 (Right), the percentage of *Precision* for the proposed model outperforms the two baselines. Our approach achieves the best results when the value of K is around 8. In fact, for $K = 5$, the mean precision, is equal to 12,3%. Whereas, for $K = 8$, it has an average of 24,5% showing a drop of the query prediction accuracy of 49,79% *vs.* an exceeding about 19,6% against the first baseline and around 17,4% against the second one. These results highlight that the proposed approach can better improve query prediction accuracy even for a high number of predicted queries. Moreover, our approach achieves a good coverage, since it produces predictions for 76% of queries contained within the test set Q_T .

5 Conclusion and Future Work

In this paper, we have introduced a novel probabilistic approach for users' search intents prediction by training a HMM from a *logsonomy*, carried out from a real world *folksonomy*. We tackle the challenge of learning a large HMM from hundreds of thousands of user's sessions by summarizing individual queries, resources and users into search intents, formally, represented as triadic concepts. To the best of our knowledge, search intent prediction (using HMMs) in such social bookmarking systems have never been modeled before. Our future research will focus on further study other more sophisticated Markov models such as variable length HMM in a *folksonomy* search. This include modeling hidden states that represent users' search intents, which could be an underlying semantic concept, especially with the help of domain knowledge such as the online ontologies. Indeed, the use of online ontologies may allow prediction systems to find out

how specific the user interest is, and use this information to fine predictions. It remains to be seen whether more sophisticated models can further raise the performance bar for the query prediction in *folksonomies*.

References

1. Baeza-Yates, R., Berthier, R.N.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
2. Benz, D., Hotho, A., Jäschke, R., Krause, B., Stumme, G.: Query Logs as Folksonomies. *Datenbank-Spektrum* 10(1), 15–24 (2010)
3. Bischoff, K., Firan, C.S., Nejdil, W., Paiu, R.: Can all tags be used for search? In: Proc. of the 17th ACM Conf. on Information and Knowledge Management, CIKM 2008, pp. 193–202. ACM Press, Napa Valley (2008)
4. Cao, H., Jiang, D., Pei, J., Chen, E., Li, H.: Towards context-aware search by learning a very large variable length hidden markov model from search logs. In: Proc. of the 18th Intl. Conf. on World wide web, WWW 2009, Madrid, Spain, pp. 191–200 (2009)
5. He, Q., Jiang, D., Liao, Z., Hoi, S.C.H., Chang, K., Lim, E., Li, H.: Web query recommendation via sequential query prediction. In: Proc. of the 2009 IEEE Intl. Conf. on Data Engineering, pp. 1443–1454. IEEE Computer Society, Washington, DC, USA (2009)
6. Heymann, P., Koutrika, G., Garcia-Molina, H.: Can social bookmarking improve web search? In: Proc. of the First ACM International Conference on Web Search and Data Mining, WSDM 2008. ACM, New York (2008)
7. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in folksonomies. *Web Semantics: Science, Services and Agents on the World Wide Web* 6, 38–53 (2008)
8. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS - an algorithm for mining iceberg tri-lattices. In: Proc. of the 6th IEEE Intl. Conf. on Data Mining, ICDM 2006, pp. 907–911. IEEE Computer Society, Hong Kong (2006)
9. Krause, B., Hotho, A., Stumme, G.: A Comparison of Social Bookmarking with Traditional Search. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 101–113. Springer, Heidelberg (2008)
10. Krause, B., Jäschke, R., Hotho, A., Stumme, G.: Logsonomy - social information retrieval with logdata. In: Proc. of the Nineteenth ACM Conf. on Hypertext and Hypermedia, HT 2008, New York, NY, USA, pp. 157–166 (2008)
11. Mei, Q., Jiang, J., Suz, H., Zhai, C.: Search and tagging: Two sides of the same coin? In: Technical Report No. 2919, University of Illinois at Urbana-Champaign (UIUCDCS-R-2007-2919) (2007)
12. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
13. Pan, J., Taylor, S., Thomas, E.: Reducing ambiguity in tagging systems with folksonomy search expansion. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 669–683. Springer, Heidelberg (2009)
14. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE* 77(2), 257–286 (1989)

Factorizing Three-Way Ordinal Data Using Triadic Formal Concepts

Radim Belohlavek, Petr Osička, and Vilem Vychodil

Department of Computer Science
Data Analysis and Modeling Lab
Palacky University, Olomouc
17. listopadu 12, CZ-77146 Olomouc
Czech Republic
{radim.belohlavek,vychodil,osicka}@acm.org

Abstract. The paper presents a new approach to factor analysis of three-way ordinal data, i.e. data described by a 3-dimensional matrix I with values in an ordered scale. The matrix describes a relationship between objects, attributes, and conditions. The problem consists in finding factors for I , i.e. finding a decomposition of I into three matrices, an object-factor matrix A , an attribute-factor matrix B , and a condition-factor matrix C , with the number of factors as small as possible. The difference from the decomposition-based methods of analysis of three-way data consists in the composition operator and the constraint on A , B , and C to be matrices with values in an ordered scale. We prove that optimal decompositions are achieved by using triadic concepts of I , developed within formal concept analysis, and provide results on natural transformations between the space of attributes and conditions and the space of factors. We present an illustrative example demonstrating the usefulness of finding factors and a greedy algorithm for computing decompositions.

1 Motivation

The aim of this paper is to develop a new method of analysis of three-way ordinal data that is based on matrix decomposition. Various types of matrix decomposition are utilized in many methods of data analysis, in particular for two-way data. Related to the topic of our paper are the methods for binary data, see e.g. [4,16,18,23,24]. Recently, there has been a growing interest in analyzing three-way and generally N -way data, i.e. data represented by N -dimensional matrices. [12] provides an up-to-date survey with 244 references, see also [6,13,21]. In [15,4], we developed methods of factor analysis for two-way binary and ordinal data. These methods were extended for three-way binary data in [2]. The purpose of the present paper is to extend the method from [2] to the case of three-way ordinal data.

Such data is represented by a 3-dimensional matrix which is denoted by I in this paper and whose entries, denoted I_{ijt} , are elements of some partially ordered

set $\langle L, \leq \rangle$. In this paper, we interpret the entries as follows (other interpretations are possible):

I_{ijt} is the degree to which object i has attribute j under condition t .

Typical examples of such data are: customer surveys (objects are products, attributes are product features, conditions are customers participating in the survey; I_{ijt} represents the degree to which the customer t considers the feature j important, or being of good quality, for the product i); questionnaires, results of querying a database at different points in time, and the like.

It is well-known [17,24] that the decomposition methods that were designed for real-valued data are not suitable for binary data because they distort the meaning of binary data. The same is true of ordinal data. Therefore, even if the scale L consists of real numbers (such as $L = [0, 1]$), one needs to take care when selecting the matrix decompositions to be employed. In our paper, we generalize the Boolean matrix decomposition from [2]. We assume that the scale $\langle L, \leq \rangle$ is a complete lattice equipped with a binary operation that commutes with suprema (examples for $L = [0, 1]$ are \otimes being min, the usual number-theoretic product, or any left-continuous t-norm, see Section 2.1). This ensures, as we show below, that the decomposition and its results have a clear interpretation.

In particular, we look for a decomposition of a given $n \times m \times p$ matrix I with entries $I_{ijt} \in L$ into a product

$$I = \circ(A, B, C) \tag{1}$$

of an $n \times k$ object-factor matrix A with entries $A_{ik} \in L$, an $m \times k$ attribute-factor matrix B with entries $B_{jk} \in L$, and a $p \times k$ condition-factor matrix A with entries $C_{tk} \in L$, with \circ defined by

$$\circ(A, B, C)_{ijt} = \bigvee_{l=1}^k A_{il} \otimes B_{jl} \otimes C_{tl}, \tag{2}$$

with \bigvee being the supremum in L . It is easily seen that if $p = 1$, $\circ(A, B, C)$ may be identified with the \bigvee - \otimes product, well-known from fuzzy set theory [11], that is employed in [15]. Importantly, the decomposition (2) has the following meaning: The degree to which the object i has the attribute j under the condition t is equal to the truth degree of the proposition “there exists a factor l such that l applies to i , j is a particular manifestation of l , and t is one of the conditions under which l appears”.

In this paper, we describe optimal decompositions (1) of I , i.e. those with the number k of factors as small as possible. We call such k the *Schein rank* of I and denote it by $\rho(I)$. Furthermore, we provide basic geometric insight into the problem and describe natural transformations between the space of attributes \times conditions and the space of factors that enable us to go between the descriptions of objects in these two spaces. We provide an illustrative example, observe that the decomposition problem is NP-hard, and outline an algorithm for computing suboptimal decompositions.

2 Optimal Factorizations Using Triadic Fuzzy Concepts

2.1 Basic Notions from Triadic Concept Analysis

This section provides the notions from triadic concept analysis of data with fuzzy attributes (see [3] for more information, also [15,25] for binary attributes).

We assume that the scale $\langle L, \leq \rangle$ equipped with the binary operation \otimes (mentioned above) form a complete residuated lattice, i.e. a structure

$$\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$$

such that $\langle L, \wedge, \vee, 0, 1 \rangle$ is a complete lattice with 0 and 1 being the least and greatest element of L , respectively; $\langle L, \otimes, 1 \rangle$ is a commutative monoid (i.e. \otimes is commutative, associative, and $a \otimes 1 = a$ for each $a \in L$); \otimes and \rightarrow satisfy so-called adjointness property: $a \otimes b \leq c$ iff $a \leq b \rightarrow c$. Residuated lattices are used in several areas of mathematics, notably in mathematical fuzzy logic. Elements a of L are called truth degrees. \otimes and \rightarrow are (truth functions of) many-valued conjunction and implication. Examples of residuated lattices are well known and we refer the reader e.g. to [9]. Residuated lattices may be used as structures of truth degrees for fuzzy sets; we assume that the reader is familiar with the notions from fuzzy set theory [9,11].

A *triadic fuzzy context* is a quadruple $\langle X_1, X_2, X_3, I \rangle$ where I is a ternary fuzzy relation between X_1 (set of objects), X_2 (set of attributes), and X_3 (set of conditions), with $I(x_1, x_2, x_3)$ being interpreted as the truth degree of “object x_1 has attribute x_2 under condition x_3 ”. Let $\{i, j, k\} = \{1, 2, 3\}$. We denote the degree $I(x_1, x_2, x_3)$ also by $I\{x_i, x_j, x_k\}$. For a fuzzy set $C_k : X_k \rightarrow L$, we define a dyadic fuzzy context $\langle X_i, X_j, I_{C_k}^{ij} \rangle$ by $I_{C_k}^{ij}(x_i, x_j) = \bigwedge_{x_k \in X_k} (C_k(x_k) \rightarrow I\{x_i, x_j, x_k\})$. The concept-forming operators induced by $\langle X_i, X_j, I_{C_k}^{ij} \rangle$ are denoted by \langle^{i,j,C_k} . Therefore, for fuzzy sets $C_i : X_i \rightarrow L$ and $C_j : X_j \rightarrow L$, $x_j \in X_j$, and $x_k \in X_k$, we put

$$\begin{aligned} C_i^{\langle^{i,j,C_k}}(x_j) &= \bigwedge_{x_i \in X_i} C_i(x_i) \rightarrow I_{C_k}^{ij}(x_i, x_j), \\ C_j^{\langle^{i,j,C_k}}(x_i) &= \bigwedge_{x_j \in X_j} C_j(x_j) \rightarrow I_{C_k}^{ij}(x_i, x_j). \end{aligned}$$

A *triadic fuzzy concept* of $\langle X_1, X_2, X_3, I \rangle$ is a triplet $\langle D_1, D_2, D_3 \rangle$ of fuzzy sets $D_i \in L^{X_i}$ such that $D_1 = D_2^{(1,2,D_3)}$, $D_2 = D_3^{(2,3,D_1)}$, and $D_3 = D_1^{(3,1,D_2)}$. D_1 , D_2 , and D_3 are called the *extent*, *intent*, and *modus* of $\langle D_1, D_2, D_3 \rangle$.

The set of all triadic fuzzy concepts of $\langle X_1, X_2, X_3, I \rangle$ is denoted by

$$\mathcal{T}(X_1, X_2, X_3, I)$$

and forms a trilattice [25], called the *concept trilattice* of $\langle X_1, X_2, X_3, I \rangle$ [3].

In what follows, we assume that $X_1 = \{1, \dots, n\}$, $X_2 = \{1, \dots, m\}$ and $X_3 = \{1, \dots, p\}$. For convenience, we identify a fuzzy set A in $\{1, \dots, p\}$ with a p -component tuple $\langle A(1), \dots, A(p) \rangle$ (we use $A_i = A(i)$ and the like, the set of all such tuples is denoted L^p); likewise, we identify ternary fuzzy relations between

$X_1, X_2,$ and X_3 with (3-dimensional) $n \times m \times p$ matrices with degrees from L (the set of all such matrices is denoted by $L^{n \times m \times p}$); the same for binary fuzzy relations and (2-dimensional) matrices.

2.2 Factorization Using Triadic Concepts as Factors

Next, we show how triadic fuzzy concepts of I may be used as factors for decomposition (2). Call a 3-dimensional matrix $J \in L^{n \times m \times p}$ a *cuboidal matrix* (shortly, a *cuboid*) if there exist vectors $A \in L^n, B \in L^m,$ and $C \in L^p$ such that $J_{ijt} = A_i \otimes B_j \otimes C_t,$ or $J = \circ(A, B, C)$ with a small abuse of notation. The role of cuboids for decompositions (2) is the following:

Lemma 1. $I = \circ(A, B, C)$ for an $n \times k$ matrix $A, m \times k$ matrix $B,$ and $p \times k$ matrix C iff I is a \vee -superposition of k cuboids $J_1, \dots, J_k,$ i.e.

$$I = J_1 \vee \dots \vee J_k.$$

In addition, for each $l = 1, \dots, k, J_l = \circ(A_{\downarrow l}, B_{\downarrow l}, C_{\downarrow l}),$ i.e. each J_l is the product of the l -th columns of $A, B,$ and $C.$

Proof. Easy, from definitions. □

It follows that to decompose I using a small number of factors, one needs to find a small number of cuboids contained in I that cover all the entries of $I.$ We say that a cuboid J is contained in I if $J_{ijt} \leq I_{ijt}$ for all $i, j, t.$ As the following lemma shows, triadic fuzzy concepts of I correspond to maximal cuboids contained in $I.$

Lemma 2 (3). $\langle D_1, D_2, D_3 \rangle$ is a triadic concept of I iff $J = \circ(D_1, D_2, D_3)$ is a maximal cuboid contained in $I.$

As we show next, the triadic fuzzy concepts of I may be used as factors for decompositions of I the following way. For a set

$$\mathcal{F} = \{ \langle D_{11}, D_{12}, D_{13} \rangle, \dots, \langle D_{k1}, D_{k2}, D_{k3} \rangle \}$$

of triadic fuzzy concepts of I (we fix this indexing of concepts, i.e. we speak of the l -th concept in \mathcal{F}), we denote by $A_{\mathcal{F}}$ the $n \times k$ matrix in which the l -th column coincides with the extent $D_{l1},$ by $B_{\mathcal{F}}$ the $m \times k$ matrix in which the l -th column coincides with the intent $D_{l2}, C_{\mathcal{F}}$ the $p \times k$ matrix in which the l -th column coincides with the modus D_{l3} of the l -th concept $\langle D_{l1}, D_{l2}, D_{l3} \rangle.$ That is,

$$(A_{\mathcal{F}})_{il} = D_{l1}(i), \quad (B_{\mathcal{F}})_{jl} = D_{l2}(j), \quad (C_{\mathcal{F}})_{tl} = D_{l3}(t).$$

If $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}}),$ we call the triadic concepts from \mathcal{F} *factor concepts.* Given $I,$ our aim is to find a small set \mathcal{F} of factor concepts.

Using triadic concepts of I as factors is intuitively appealing because triadic concepts are simple models of human concepts according to traditional logic

approach [15]. In addition, the extents, intents, and modi of the concepts, i.e. columns of $A_{\mathcal{F}}$, $B_{\mathcal{F}}$, and $C_{\mathcal{F}}$, have a straightforward interpretation: they represent the objects, attributes, and conditions to which the factor concept applies (see Section 3 for particular examples).

The next theorem shows that triadic concepts are universal and optimal factors in that every 3-dimensional matrix can be factorized using triadic concepts and the factorizations that employ triadic concepts as factors are optimal.

Theorem 1 (optimality). *Let I be an $n \times m \times p$ matrix with degrees from L .*

- (1) $\rho(I) \leq \min(nm, np, mp)$.
- (2) *There exists $\mathcal{F} \subseteq \mathcal{T}(X_1, X_2, X_3, I)$ with $|\mathcal{F}| = \rho(I)$ for which*

$$I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}}).$$

Proof. Sketch: (1) One can show that I can be factorized by a set of triadic concepts, each of which is a so-called ij -join induced by an object and an attribute. Such set has at most nm concepts, whence $\rho(I) \leq nm$. In a similar manner, one proves $\rho(I) \leq np$ and $\rho(I) \leq mp$.

(2) One can show that if $I = \circ(A, B, C)$ with inner dimension k , then the k cuboids of which I is the \vee -superposition are each contained in some triadic fuzzy concepts of I . These triadic fuzzy concepts are again cuboids and are contained in I . It can be shown that for the set \mathcal{F} of these concepts, $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$. Clearly, $|\mathcal{F}| \leq k$. □

Theorem 1 means that when looking for factors for decompositions of I , one may restrict the search to triadic concepts. Theorem 1 is the basis for the decomposition algorithm presented in Section 4.

2.3 Transformations Between the Attribute×Condition Space and the Factor Space

Given a decomposition $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$ of an $n \times m \times p$ matrix I for some k -element set \mathcal{F} of factor concepts, one can ask for a transformation of a description of a given object in attribute×condition space $L^{m \times p}$ into a description in factor space L^k . For the dyadic case, such transformations are described in [4] and were utilized in [19,20] for improving classification of binary data.

In the attribute×condition space, the object $i \in X_1$ is represented by the $m \times p$ matrix $I_{i_}$ corresponding to the dyadic context $\langle X_2, X_3, I_{\{1/i\}}^{23} \rangle$ (see Section 2.1), called i -th (object) dyadic cut. In the factor space, i is represented by the i -th row $A_{i_}$ of A .

Consider the transformations $g : L^{m \times p} \rightarrow L^k$ and $h : L^k \rightarrow L^{m \times p}$ defined for $P \in L^{m \times p}$ and $Q \in L^k$ by

$$(g(P))_l = \bigwedge_{j=1}^m \bigwedge_{t=1}^p (B_{jl} \otimes C_{tl} \rightarrow P_{jt}) \tag{3}$$

$$(h(Q))_{jt} = \bigvee_{l=1}^k (Q_l \otimes B_{jl} \otimes C_{tl}) \tag{4}$$

for $l \in \{1, \dots, k\}$, $j \in \{1, \dots, m\}$ and $t \in \{1, \dots, p\}$.

The previous two operators have the following interpretation. (3) says that the degree to which object i applies to factor l equals to a degree to which i has every attribute j under every condition t such that j is a manifestation of l and t is one of the conditions under which l appears; (4) says that a degree to which object i has attribute j under condition t equals the degree to which there is a factor l such that l applies to i , j is a manifestation of l , and l is one of the conditions under which l appears.

The suitability of g and h as natural transformations between attribute \times condition and factor spaces is demonstrated by the following theorem.

Theorem 2. For $i \in \{1, \dots, n\}$:

$$g(I_{i_}) = A_{i_} \text{ and } h(A_{i_}) = I_{i_}.$$

That is, g maps the object dyadic cuts of I to the rows of A and vice versa, h maps the rows of A to the object dyadic cuts of I .

Proof. $h(A_{i_}) = I_{i_}$ follows directly from $I = \circ(A, B, C)$. Since $A = A_{\mathcal{F}}$, $B = B_{\mathcal{F}}$, and $C = C_{\mathcal{F}}$, the l -th columns of A, B and C coincide with the extent D_{l1} , intent D_{l2} , and modus D_{l3} of a triadic concept $\langle D_{l1}, D_{l2}, D_{l3} \rangle \in \mathcal{F}$, respectively.

$$\begin{aligned} (g(I_{i_}))_l &= \bigwedge_{j=1}^m \bigwedge_{t=1}^p (B_{jl} \otimes C_{tl} \rightarrow (I_{i_})_{jt}) = \\ &= \bigwedge_{j=1}^m \bigwedge_{t=1}^p ((D_{l2})_j \otimes (D_{l3})_t \rightarrow I_{ijt}) = \\ &= (D_{l2}^{(1,2,D_{l3})})_i = (D_{l1})_i = A_{il}. \end{aligned}$$

□

The following theorem shows that g and h form an isotone Galois connection.

Theorem 3. For $P, P' \in L^{m \times p}$ and $Q, Q' \in L^k$:

$$P \leq P' \Rightarrow g(P) \leq g(P'), \tag{5}$$

$$Q \leq Q' \Rightarrow h(Q) \leq h(Q'), \tag{6}$$

$$h(g(P)) \leq P, \tag{7}$$

$$Q \leq g(h(Q)). \tag{8}$$

Proof. By routine verification using (4) and (3), and basic properties of residuated lattices. □

(5)–(8) are natural properties of transformations between attributes \times condition and factor spaces. For example, (5) shows that the higher the degree to which an object has attributes under conditions, the higher the degree to which factors apply to the object, while (6) states analogous relationship in the opposite direction.

A geometry behind the transformations is described by the following assertion. For $P \in L^{m \times p}$ and $Q \in L^k$, put

$$\begin{aligned} g^{-1}(Q) &= \{P \in L^{m \times p} \mid g(P) = Q\}, \\ h^{-1}(P) &= \{Q \in L^k \mid h(Q) = P\}. \end{aligned}$$

Recall that $S \subseteq L^s$ is called *convex* if $V \in S$ whenever $U \leq V \leq W$ for some $U, W \in S$.

Theorem 4. (1) $g^{-1}(Q)$ is a convex partially ordered subspace of the attribute and condition space and $h(Q)$ is the least element of $g^{-1}(Q)$.
 (2) $h^{-1}(P)$ is a convex partially ordered subspace of the factor space and $g(P)$ is the largest element of $h^{-1}(P)$.

Proof. By standard application of the properties of isotone Galois connections. □

According to Theorem 4 the space $L^{m \times p}$ of attributes and conditions and the space L^k of factors are partitioned into an equal number of convex subsets. The subsets of the space of attributes and conditions have least elements and the subsets of the space of factors have greatest elements. g maps every element of any convex subset of the space of attributes and conditions to the greatest element of the corresponding subset of the factor space, whereas h maps every element of some convex subset of the space of factors to the least element of the corresponding convex subset of the space of attributes and conditions.

3 Illustrative Example

In this section, we present an illustrative example of factorization. We consider input data containing information about potential car buyers and their motivation for the purchase of a particular type of car. Such data is usually obtained via a customer survey. We assume that customers expressed the degrees of their motivation using a 3-element scale (not at all, partly, significantly).

We represent the data by a triadic fuzzy context $\langle X_1, X_2, X_3, I \rangle$, where $X_1 = \{a, b, c, d, e, f, g, h\}$ is a set of cars, $X_2 = \{hp, sp, ac, pr, mc, sa\}$ is a set of car characteristics: horse power, space (i.e. the car is spacious), acceleration/speed, price, monthly cost, safety; and $X_3 = \{A, B, C, D, E\}$ is a set of customers participating in the survey. The fact that x is related to y under z to the degree $I(x, y, z)$ is interpreted as “the attribute y motivates the customer z for the purchase of x to the degree $I(x, y, z)$ ”. We represent the scale of degrees used in the survey by a 3-element Lukasiewicz chain $\{0, \frac{1}{2}, 1\}$, with the following interpretation:

- 0 ... not at all,
- $\frac{1}{2}$... partly,
- 1 ... significantly.

We consider I given by the table in Fig. 1. The rows of the table correspond to cars, the columns correspond to attributes under the various conditions (customers).

In such data, there exists a three-element set $\mathcal{F} = \{F_1, F_2, F_3\}$ of factor concepts. We fix the order of objects, attributes and conditions to the order in which they appear in Fig. 1 in order to represent the extents, intents, and modi of the factor concepts by their characteristic vectors. For example, the characteristic vector of the extent of F_1 is $\langle 1, \frac{1}{2}, \frac{1}{2}, 0, 1, 0, \frac{1}{2}, 1 \rangle$ which means that car a belongs

	A					B					C					D					E						
	hp	sp	ac	pr	mc	hp	sp	ac	pr	mc	hp	sp	ac	pr	mc	hp	sp	ac	pr	mc	hp	sp	ac	pr	mc		
a	1	0	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	1	0	1	0	0
b	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0		
c	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0		
d	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0		
e	1	0	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	1	0		
f	0	0	0	0	0	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	0	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$		
g	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0		
h	1	0	1	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	0	1	0	0		

Fig. 1. Triadic context

to the extent of F_1 to the degree 1, car b to the degree $\frac{1}{2}$ and so forth. The factor concepts in \mathcal{F} are represented by the following triplets of the characteristic vectors of their extents, intents, and modi (the vectors are separated by ;):

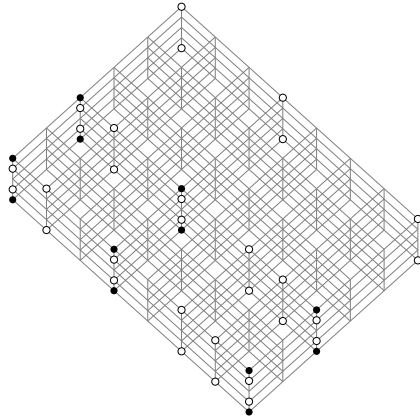
$$\begin{aligned}
 F_1 & \dots \langle 1, \frac{1}{2}, \frac{1}{2}, 0, 1, 0, \frac{1}{2}, 1; 1, 0, 1, 0, 0, \frac{1}{2}; 1, \frac{1}{2}, 0, \frac{1}{2}, 1 \rangle, \\
 F_2 & \dots \langle 0, \frac{1}{2}, 0, \frac{1}{2}, 1, 1, \frac{1}{2}; \frac{1}{2}; \frac{1}{2}, 1, 0, \frac{1}{2}, \frac{1}{2}, 1; 0, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2} \rangle, \\
 F_3 & \dots \langle 0, 0, \frac{1}{2}, 1, 0, 1, \frac{1}{2}, \frac{1}{2}; 0, 0, 0, 1, 1, \frac{1}{2}; 0, 0, 1, 0, \frac{1}{2} \rangle.
 \end{aligned}$$

Using \mathcal{F} , we obtain the following 8×3 object-factor matrix $A_{\mathcal{F}}$, 6×3 attribute-factor matrix $B_{\mathcal{F}}$, and 5×3 conditions-factor matrix $C_{\mathcal{F}}$:

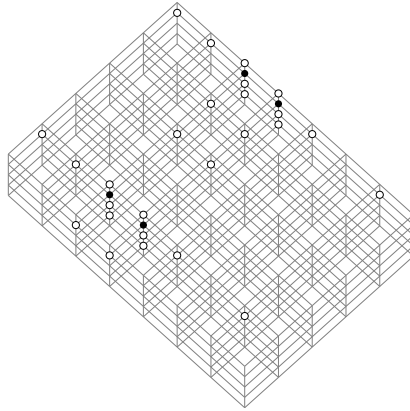
$$A_{\mathcal{F}} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad B_{\mathcal{F}} = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{2} & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix}, \quad C_{\mathcal{F}} = \begin{pmatrix} 1 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 \\ 1 & 1 & \frac{1}{2} \end{pmatrix}.$$

One can check that $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$, i.e., I decomposes into three (two-dimensional) matrices using three factors. Note that the meaning of the factors can be seen from the extents, intents, and modi of the factor concepts. For instance, F_1 can be interpreted as “the ability to go fast”. Indeed, F_1 is manifested by the attributes horse power and speed to the degree 1, and by safety to the degree $\frac{1}{2}$. The factor F_2 is manifested by space and safety to the degree 1, and by horse power, price, and monthly cost to the degree $\frac{1}{2}$. This suggests that F_2 can be interpreted as “being a family car”. The high degree manifestations of F_3 are price and monthly cost, leading to a possible interpretation as “cost-effectiveness”. As a result, by finding the factors set $\mathcal{F} = \{F_1, F_2, F_3\}$, we have explained the structure of the input data set I using three factors which describe the attractivity of cars to customers in terms of their characteristics.

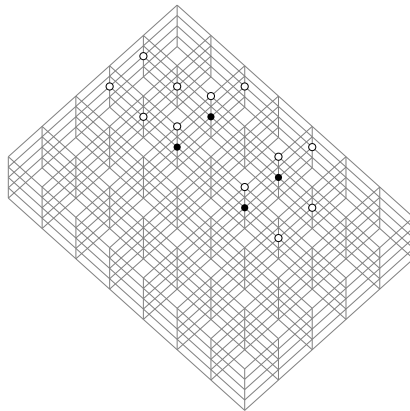
Let us recall that the factor concepts $\mathcal{F} = \{F_1, F_2, F_3\}$ can be seen as maximal cuboids in I . Indeed, I itself can be depicted as three-dimensional box where the



F_1 : "ability to go fast"



F_2 : "being a family car"



F_3 : "cost-effectiveness"

Fig. 2. Geometric meaning of factors

Algorithm 1. COMPUTEFACTORS(X, Y, Z, I)

```

1  compute  $\mathcal{B}(X, Y, Z, I)$ ;
2  set  $\mathcal{S}$  to  $\mathcal{B}(X, Y, Z, I)$ ;
3  set  $\mathcal{F}$  to  $\emptyset$ ;
4  set  $U$  to  $X \times Y \times Z$ ;
5  while  $U \neq \emptyset$  do
6    select  $\langle A, B, C \rangle \in \mathcal{S}$  which maximizes  $|U \cap S_{\langle A, B, C \rangle}|$ ;
7    add  $\langle A, B, C \rangle$  to  $\mathcal{F}$ ;
8    set  $U$  to  $U \setminus S_{\langle A, B, C \rangle}$ ;
9    remove  $\langle A, B, C \rangle$  from  $\mathcal{S}$ ;
10 end
11 return  $\mathcal{F}$ 

```

axes correspond to cars, their characteristics, and customers. Figure 2 shows the three factors depicted as cuboids. White and black circlets in Figure 2 correspond to elements in I . Namely, a white circlet is present on the intersection of $x \in X$, $y \in Y$, and $z \in Z$ in the diagram iff $\circ(A_{-i}, B_{-i}, C_{-i})(x, y, z) = \frac{1}{2}$. Furthermore, the circlet is black iff $\circ(A_{-i}, B_{-i}, C_{-i}) = 1$. That is, for a factor F_i , the circle depicts the degree to which x belongs to the extent of F_i , y belongs to the intent of F_i , and z belongs to the modus of F_i .

4 Algorithms

Due to the above results, the problem of finding a minimal decomposition of $\langle X, Y, Z, I \rangle$ can be seen as a problem of finding a minimal subset $\mathcal{F} \subseteq \mathcal{T}(X, Y, Z, I)$ of formal concepts that cover I . As a result, we can reduce the problem of finding a matrix decomposition to the set-covering problem in the following way. The universe U that should be covered corresponds to $X \times Y \times Z$. The family \mathcal{S} of subsets of the universe U that is used for finding a cover contains for each triadic concept in $\mathcal{T}(X, Y, Z, I)$ a set of indices which the triadic concept covers. More precisely, $\mathcal{S} = \{S_{\langle A, B, C \rangle} \mid \langle A, B, C \rangle \in \mathcal{T}(X, Y, Z, I)\}$, where $S_{\langle A, B, C \rangle} = \{(i, j, k) \mid A_i \otimes B_j \otimes C_k = I_{ijk}\}$. In this setting, we are looking for $\mathcal{C} \subseteq \mathcal{S}$ as small as possible such that $\bigcup \mathcal{C} = U$. Thus, finding factor concepts is indeed an instance of the set-covering problem. It is well known that the set covering optimization problem is NP-hard and the corresponding decision problem is NP-complete. However, there exists a greedy approximation algorithm for the set covering optimization problem which achieves an approximation ratio $\leq \ln(|U|) + 1$, see [7]. This gives us a “naive” greedy-approach algorithm for computing all factor concepts.

Algorithm 1, implementing the above-mentioned greedy approach in our setting, first computes a set of all triadic concepts which are stored in \mathcal{S} , see lines 1–2. Then it iteratively selects triadic concepts from \mathcal{S} , maximizing their overlap with the remaining tuples in U , see lines 5–9. Notice that the size of the overlap of $\langle A, B, C \rangle$ with U is the number of yet uncovered indices at which the cuboid corresponding to $\langle A, B, C \rangle$ has the same value as I . More precisely, it is the number of elements of $U \cap S_{\langle A, B, C \rangle}$.

The drawback of Algorithm 1 is that it first computes a possibly large set of triadic concepts and then it selects a small subset of it as the set of factor concepts. This difficulty can be overcome by computing the factor concepts “on demand”. This can be done in a way analogous to the one described in [5]. Due to the lack of space, a description of such an algorithm, as well as its experimental evaluation and comparison with Algorithm 1, is postponed to the extended version of this paper.

5 Conclusions and Future Work

We presented a method for factorization of three-way ordinal data. The method uses triadic formal concepts of the input data as factors. Due to the clear interpretation of such concepts, the factor model and the factors have a transparent meaning. We proved that the factorizations using triadic concepts are optimal. Furthermore, we provided natural transformations between the space of attributes and conditions, and the space of factors. We outlined an approximation algorithm that utilized a greedy approach to the set cover problem.

The topics left for the extended version of this paper and for future research include:

- A more efficient version of the greedy algorithm for computing decompositions. The aim is to overcome the computation of all the triadic concepts. Instead, the algorithm is to compute, using a greedy strategy, a hopefully good factor concept.
- Investigation of approximate factorizability, i.e. the problem of finding a set of concepts such that $\circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$ equals I at least to a specified degree. An interesting issue is whether the factors computed under this criterion will differ from the factors computed under the criterion of exact factorization.
- Experimental study of the factor model presented in this paper involving domain experts. In particular, comparing the present method with other methods for reducing dimensionality of three-way data.
- Further study of algorithms, in particular their computational complexity, approximability characteristics, and performance evaluation.

Acknowledgment. Supported by Grant No. P202/10/0262 of the Czech Science Foundation and by research plan MSM 6198959214.

References

1. Belohlavek, R.: Optimal decomposition of matrices with entries from residuated lattices. *J. Logic and Computation* (to appear, preliminary version appeared in *Proc. IEEE Intelligent Systems*, pp. 15-2–15-7 (2008))
2. Belohlavek, R., Glodeanu, C.V., Vychodil, V.: Optimal factorization of three-way binary data using triadic concepts. (submitted, preliminary version appeared in *Proc. IEEE GrC 2010*, pp. 61–66 (2010))
3. Belohlavek, R., Osicka, P.: Triadic concept analysis of data with fuzzy attributes. In: *Proc. 2010 IEEE International Conference on Granular Computing*, San Jose, California, August 14–16, pp. 661–665 (2010)

4. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Computer and System Sci.* 76(1), 3–20 (2010)
5. Belohlavek, R., Vychodil, V.: Factor analysis of incidence data via novel decomposition of matrices. In: Ferré, S., Rudolph, S. (eds.) *ICFCA 2009. LNCS(LNAI)*, vol. 5548, pp. 83–97. Springer, Heidelberg (2009)
6. Cichocki, A., et al.: *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. J. Wiley, Chichester (2009)
7. Cormen, T.H., et al.: *Introduction to Algorithms*, 2nd edn. MIT Press, Cambridge (2001)
8. Ganter, B., Wille, R.: *Formal Concept Analysis. Mathematical Foundations*. Springer, Berlin (1999)
9. Hájek, P.: *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht (1998)
10. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS – An Algorithm for Mining Iceberg Tri-Lattices. In: *ICDM 2006*, pp. 907–911 (2006)
11. Klir, G.J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice-Hall, Englewood Cliffs (1995)
12. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Review* 51(3), 455–500 (2009)
13. Kroonenberg, P.M.: *Applied Multiway Data Analysis*. J. Wiley, Chichester (2008)
14. Kuznetsov, S., Obiedkov, S.: Comparing performance of algorithms for generating concept lattices. *J. Experimental and Theoretical Artificial Intelligence* 14(2–3), 189–216 (2002)
15. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) *ICCS 1995. LNCS*, vol. 954, pp. 32–34. Springer, Heidelberg (1995)
16. Mickey, M.R., Mundle, P., Engelman, L.: Boolean factor analysis. In: Dixon, W.J. (ed.) *BMDP Statistical Software Manual*, vol. 2, pp. 849–860. University of California Press, Berkeley (1990)
17. Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The Discrete Basis Problem. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *PKDD 2006. LNCS (LNAI)*, vol. 4213, pp. 335–346. Springer, Heidelberg (2006)
18. Nau, D.S., Markowsky, G., Woodbury, M.A., Amos, D.B.: A Mathematical Analysis of Human Leukocyte Antigen Serology. *Math. Biosciences* 40, 243–270 (1978)
19. Outrata, J.: Preprocessing input data for machine learning by FCA. In: Kryszkiewicz, M., Obiedkov, S. (eds.) *Proc. CLA 2010*, vol. 672, pp. 187–198. University of Sevilla, CEUR WS (2010)
20. Outrata, J.: Boolean factor analysis for data preprocessing in machine learning. In: Draghici, S., et al. (eds.) *Proc. ICMLA 2010, Intern. Conf. on Machine Learning and Applications*, pp. 899–902. IEEE, Washington, DC (2010)
21. Smilde, A., Bro, R., Geladi, P.: *Multi-way Analysis: Applications in the Chemical Sciences*. J. Wiley, Chichester (2004)
22. Stockmeyer, L.J.: The set basis problem is NP-complete. IBM Research Report RC5431, Yorktown Heights, NY (1975)
23. Tang, F., Tao, H.: Binary principal component analysis. In: *Proc. British Machine Vision Conference 2006*, pp. 377–386 (2006)
24. Tatti, N., Mielikäinen, T., Gionis, A., Mannila, H.: What is the dimension of your binary data? In: *ICDM 2006*, pp. 603–612 (2006)
25. Wille, R.: The basic theorem of triadic concept analysis. *Order* 12, 149–158 (1995)

On Possibilistic Skyline Queries

Patrick Bosc, Allel Hadjali, and Olivier Pivert

Irisa – Enssat, University of Rennes 1
Technopole Anticipa 22305 Lannion Cedex France
{bosc,hadjali,pivert}@enssat.fr

Abstract. This paper deals with Skyline queries in the context of possibilistic databases, where uncertain attribute values are represented by possibility distributions. In this framework, Skyline queries aim at computing the extent to which any tuple from a given relation is possibly/certainly not dominated by any other tuple from that relation. Beside the interpretation of possibilistic Skyline queries, a basic algorithm suited to their evaluation is provided.

1 Introduction

In database research, the last two decades have witnessed a growing interest in preference queries on the one hand, and uncertain databases on the other hand.

Motivations for introducing preferences inside database queries are manifold [1]. First, it has appeared to be desirable to offer more expressive query languages that can be more faithful to what a user intends to say. Second, the introduction of preferences in queries provides a basis for rank-ordering the retrieved items, which is especially valuable in case of large sets of items satisfying a query. Third, a classical query may also have an empty set of answers, while a relaxed (and thus less restrictive) version of the query might be matched by items in the database.

Approaches to database preference queries may be classified into two categories according to their qualitative or quantitative nature [1]. In the latter, preferences are expressed quantitatively by a monotone scoring function, and the overall score is positively correlated with partial scores. Since the scoring function associates each tuple with a numerical score, tuple t_1 is preferred to tuple t_2 if the score of t_1 is higher than the score of t_2 . Representatives of this family of approaches are top- k queries [2] and fuzzy-set-based approaches (e.g., [3]). In the qualitative approach, preferences are defined through binary preference relations. Since such relations can be defined in terms of scoring functions, the qualitative approach is more general than the quantitative one. Representatives of qualitative approaches are those relying on a dominance relationship, e.g. Pareto order, in particular *Preference SQL* [4], Skyline queries [5] and the approach presented in [6].

In this paper, a qualitative view of preference queries is adopted, namely the Skyline approach. One also considers databases where some attribute values

may be ill-known, which brings us back to the second topic mentioned above, i.e., uncertain databases.

Since the late 70's, many authors have made diverse proposals to model and handle databases involving uncertain or incomplete data. In particular, the last two decades have witnessed a profusion of research works on this topic. Even though most of them consider probability theory as the underlying uncertainty model, some approaches rather rely on possibility theory [7]. The initial idea which consists in applying possibility theory to the modeling of uncertain databases goes back to the early 80's [8]. More recent advances can be found in [9]. In contrast with probability theory, one expects the following advantages when using possibility theory:

- the qualitative nature of the model makes easier the elicitation of the degrees attached to the various candidate values;
- in probability theory, the fact that the sum of the degrees from a distribution must equal 1 makes it difficult to deal with incompletely known distributions.

However, we do not claim in this paper that the possibilistic framework is “better” than the probabilistic one, but that it constitutes an interesting alternative inasmuch as it captures a different kind of uncertainty (of a qualitative nature). An example — drawn from [10] — is that of a person who witnesses a car accident and who does not remember for sure the model of the car involved. In such a case, it seems reasonable to model the uncertain value by means of a possibility distribution, e.g., $\{1/\text{Mazda}, 1/\text{Toyota}, 0.7/\text{Honda}\}$ — where 0.7 is a numerical encoding in a usually finite possibility scale — rather than with a probability distribution which would be artificially normalized.

Not many works deal with both preference queries and uncertain databases, except a few approaches which consider top- k query processing in a probabilistic database context [11,12,13], an approach which deals with fuzzy queries to possibilistic databases [14], and some works about Skyline queries in the presence of missing values [15] and probabilistic data [16]. In the possibilistic database framework considered further on, we will see that Skyline queries aim at computing the extent to which any tuple from a given relation is possibly/certainly not dominated by any other tuple. This idea was initially suggested in [17], but the authors did not develop it (they just stated that “in principle, it would be possible to derive a degree of possibility and a degree of necessity for an element x to be an element of the skyline”). As an example of a possibilistic database, we will consider a used cars database resulting from the fusion of different (more or less reliable) data sources.

The remainder of this paper is structured as follows. Section 2 consists of a reminder about Pareto order and Skyline queries. Section 3 recalls the basic notions of possibility theory and introduces the possibilistic database model used further. In Section 4, the interpretation of Skyline queries in a possibilistic database context is dealt with. Section 5 is devoted to query processing and provides a basic evaluation algorithm. In Section 6, we deal with the case where the Skyline query involves a preselection condition. Finally, Section 7 recalls the main contributions and outlines some perspectives for future research.

2 About Skyline Queries

Let us first recall the general principle of Skyline queries, which are based on the use of Pareto order. Let $\{G_1, G_2, \dots, G_n\}$ be a set of the atomic preferences. We denote by $t >_{G_i} t'$ (resp. $t \geq_{G_i} t'$) the statement “tuple t satisfies preference G_i better than (resp. at least as good as) tuple t' ”. Using Pareto order, a tuple t dominates another tuple t' iff

$$\forall i \in \{1, \dots, n\}, t \geq_{G_i} t' \text{ and } \exists k \in \{1, \dots, n\}, t >_{G_k} t'.$$

In other words, t dominates t' if it is at least as good as t' regarding every preference, and is strictly better than t' regarding at least one preference. The following example uses the syntax of the language *Preference SQL* [4], which is a typical representative of a Pareto-based approach.

Example 1. Let us consider a relation *car* of schema (*make, category, price, color, mileage*) whose extension is given in Table 1 and the query:

```
select * from car where mileage ≤ 20,000
preferring (category = ‘SUV’ else category = ‘roadster’) and (make = ‘VW’
else make = ‘Ford’ else make = ‘Opel’);
```

The idea is to retain the tuples which are not dominated in the sense of the *preferring* clause. Here, $t_1, t_4, t_5,$ and t_7 are discarded since they are Pareto-dominated by t_2 and t_3 . On the other hand, t_2 and t_3 are incomparable and the final answer is $\{t_2, t_3\}$. \diamond

In the following, we will first consider “basic” Skyline queries which do not include any *where* clause, but just aim at determining the tuples from the database which are not dominated by any other, in the sense of a set of partial preferences. As mentioned above, the case of Skyline queries involving a preselection condition (as in Example 1) will be dealt with in Section 6.

Table 1. An extension of relation *car*

	<i>make</i>	<i>category</i>	<i>price</i>	<i>color</i>	<i>mileage</i>
t_1	Opel	roadster	4500	blue	20,000
t_2	Ford	SUV	4000	red	20,000
t_3	VW	roadster	5000	red	10,000
t_4	Opel	roadster	5000	red	8,000
t_5	Fiat	roadster	4500	red	16,000
t_6	Renault	sedan	5500	blue	24,000
t_7	Seat	sedan	4000	green	12,000

3 About Possibilistic Databases

3.1 Basic Notions about Possibility Theory

Possibility theory [7418] offers a qualitative model for uncertainty where a piece of information is represented by means of a possibility distribution encoding

a complete preorder over the possible situations. More formally, a possibility distribution is a function π from a domain X to the unit interval $[0, 1]$ and $\pi(a)$ expresses the degree to which a is a possible value for the considered variable. The normalization condition imposes that at least one of the values of the domain (a_0) is completely possible, i.e., $\pi(a_0) = 1$ in case of consistent information. When the domain is discrete, a possibility distribution can be written $\{\pi_1/a_1, \dots, \pi_n/a_n\}$ where a_i is a candidate value and π_i its possibility degree. Any event E is characterized by two measures: its possibility Π (expressing the fact that E may more or less occur) and its necessity N (expressing that E will occur more or less for sure). The necessity N of E is defined as: $N(E) = 1 - \Pi(\overline{E})$ where \overline{E} is the event opposite to E . The following results, where E, E_1 and E_2 denote events, are of interest further:

- $\Pi(E_1 \cup E_2) = \max(\Pi(E_1), \Pi(E_2))$
- $\Pi(E_1 \cap E_2) = \min(\Pi(E_1), \Pi(E_2))$ if E_1 and E_2 are logically independent
- $N(E_1 \cap E_2) = \min(N(E_1), N(E_2))$
- $N(E_1 \cup E_2) = \max(N(E_1), N(E_2))$ if E_1 and E_2 are logically independent
- $\Pi(E) < 1 \Rightarrow N(E) = 0$.

The two measures Π and N provide a total order over the set of regular (non fuzzy) events. The events can be rank-ordered according to Π for those which are not at all certain and according to N for those which are completely possible.

3.2 Possibilistic Databases

In contrast to a regular database, a possibilistic relational database D may have some attributes which take imprecise values. In such a case, a possibility distribution is used to represent all the more or less acceptable candidate values for the attribute. The first version of a possibilistic database model was introduced in [19]. From a semantic point of view, a possibilistic database D can be interpreted as a set of usual databases (also called worlds or interpretations) W_1, \dots, W_p , denoted by $rep(D)$, each of which being more or less possible. This view establishes a straightforward semantic connection between possibilistic and regular databases. This relationship is particularly interesting since it offers a canonical approach to the definition of queries addressed to possibilistic databases [9]. Any world W_i is obtained by choosing a candidate value in each possibility distribution appearing in D . One of these (regular) databases, let us say W_k , is supposed to correspond to the actual state of the universe modeled. Any world W_i corresponds to a conjunction of independent choices and according to previous formulas, the degree assigned to it is the minimum of the degrees tied to each of the chosen candidate values in the original possibilistic database D . Therefore, at least one of the worlds is completely possible, i.e., is assigned the degree of possibility $\Pi = 1$.

Example 2. Let us consider the possibilistic database D involving a relation im whose schema is $IM(\#i, ac, date, loc)$. Relation im describes satellite images of airplanes and each image, identified by a number ($\#i$), taken on a certain

Table 2. A possibilistic relation im

$\#i$	ac	$date$	loc
i_1	$\{1/a_1, 0.6/a_2\}$	$\{1/d_1, 0.7/d_3\}$	c_1
i_3	$\{1/a_3, 0.3/a_4\}$	d_1	c_2

location (loc) a given day ($date$) is supposed to include a single plane (ac). With the extension of im given in Table 2, eight worlds can be drawn, W_1, W_2, \dots and W_8 , since there are two candidates for ac (resp. $date$) in the first tuple of im and two candidates for ac in the second one. Each of these worlds involves one of the eight regular relations im_1 to im_8 given hereafter, issued from the possibilistic relation im .

$$\begin{aligned}
 im_1 &= \{\langle i_1, a_1, d_1, c_1 \rangle, \langle i_3, a_3, d_1, c_2 \rangle\} & \Pi &= 1 \\
 im_2 &= \{\langle i_1, a_1, d_3, c_1 \rangle, \langle i_3, a_3, d_1, c_2 \rangle\} & \Pi &= 0.7 \\
 im_3 &= \{\langle i_1, a_1, d_1, c_1 \rangle, \langle i_3, a_4, d_1, c_2 \rangle\} & \Pi &= 0.3 \\
 im_4 &= \{\langle i_1, a_1, d_3, c_1 \rangle, \langle i_3, a_4, d_1, c_2 \rangle\} & \Pi &= 0.3 \\
 im_5 &= \{\langle i_1, a_2, d_1, c_1 \rangle, \langle i_3, a_3, d_1, c_2 \rangle\} & \Pi &= 0.6 \\
 im_6 &= \{\langle i_1, a_2, d_3, c_1 \rangle, \langle i_3, a_3, d_1, c_2 \rangle\} & \Pi &= 0.6 \\
 im_7 &= \{\langle i_1, a_2, d_1, c_1 \rangle, \langle i_3, a_4, d_1, c_2 \rangle\} & \Pi &= 0.3 \\
 im_8 &= \{\langle i_1, a_2, d_3, c_1 \rangle, \langle i_3, a_4, d_1, c_2 \rangle\} & \Pi &= 0.3
 \end{aligned}$$

The value Π specified is that of each world W_i and as it is expected, one of them is completely possible. \diamond

When dealing with an incomplete database D , a practical issue is that of the efficiency of the querying process. A naive way of doing would be to make explicit all the worlds of D (at least when they are finite) in order to query each of them. Such an approach is intractable in practice and it is of prime importance to find a more realistic alternative. As we will see in Sections 4 and 5, it is possible to evaluate Skyline queries in a “compact” way, i.e., without computing the worlds of the possibilistic database.

Remark. In this paper, we do not aim at defining a closed (compositional) querying framework, we rather assume that Skyline queries are stand-alone queries (their result is not used as the input of any subsequent operation). Otherwise, one would need to define a representation system in the sense of [20], which notably implies using a more sophisticated database model such as that described in [9]. This is left for future work.

4 On a Possibilistic Skyline

Let (A_1, \dots, A_n) be the schema of the relation queried. Let Ar_1, \dots, Ar_p be the attributes concerned by a preference in the query and res the result of the query. We denote by $>_{Ar_k}$ the preference relation defined over the domain of attribute Ar_k . According to the definition of Pareto order (cf. Subsection 2),

we say that a precise tuple t_i is dominated by another precise tuple t'_j , denoted by $t_i \prec t'_j$ iff:

$$\forall k \in \{1, \dots, p\}, t'_j.Ar_k \geq_{Ar_k} t_i.Ar_k \text{ and } \exists q \in \{1, \dots, p\}, t'_j.Ar_q >_{Ar_q} t_i.Ar_q$$

and we say that a precise tuple t_i is non-dominated by another precise tuple t'_j , denoted by $t_i \not\prec t'_j$ iff $\neg(t_i \prec t'_j)$ holds.

The degree of possibility $\Pi(t)$ that a tuple t from res be non-dominated by any other tuple t' from res is computed as follows. For each interpretation π_i/t_i of t , one computes the possibility that for every tuple $t' \neq t$, there exists an interpretation t'_j of t' which does not dominate t_i . The final degree $\Pi(t)$ is the maximum of these degrees, computed over all the interpretations of t . This leads to:

$$\Pi(t) = \max_{\pi_i/t_i \in \text{int}(t)} \Pi(\pi_i/t_i) \quad (1)$$

where $\text{int}(t)$ denotes the set of interpretations of t and

$$\Pi(\pi_i/t_i) = \min(\pi_i, \min_{t' \in res \setminus \{t\}} \Pi(t_i \not\prec t'))$$

with

$$\Pi(t_i \not\prec t') = \begin{cases} 0 & \text{if } \{\pi_j/t'_j \in \text{int}(t') \mid t_i \not\prec t'_j\} = \emptyset, \\ \max_{\pi_j/t'_j \in \text{int}(t') \mid t_i \not\prec t'_j} \pi_j & \text{otherwise.} \end{cases} \quad (2)$$

Let us now define the degree of possibility $\Pi'(t)$ that tuple t be dominated by any other tuple t' . One has:

$$\Pi'(t) = \max_{\pi_i/t_i \in \text{int}(t)} \Pi'(\pi_i/t_i) \quad (3)$$

where $\Pi'(\pi_i/t_i) = \min(\pi_i, \max_{t' \in res \setminus \{t\}} \Pi(t_i \prec t'))$ and

$$\Pi(t_i \prec t') = \begin{cases} 0 & \text{if } \{\pi_j/t'_j \in \text{int}(t') \mid t_i \prec t'_j\} = \emptyset, \\ \max_{\pi_j/t'_j \in \text{int}(t') \mid t_i \prec t'_j} \pi_j & \text{otherwise.} \end{cases} \quad (4)$$

In other words, one computes the extent to which an interpretation of t' Pareto-dominates an interpretation of t .

Example 3. Let us consider a relation of schema (*make, category*), the preferences ($VW > Ford > Opel$) and ($SUV > roadster > others$) and the tuples:

$$\begin{aligned} t_1 &= \langle \{1/Opel, 0.8/VW\}, roadster \rangle \\ t_2 &= \langle Ford, \{1/SUV, 0.7/sedan\} \rangle \\ t_3 &= \langle \{1/VW, 0.6/Opel\}, roadster \rangle. \end{aligned}$$

Let us compute $\Pi(t_1)$. The interpretations of t_1 are

$$t_{11} = 1/\langle Opel, roadster \rangle \text{ and } t_{12} = 0.8/\langle VW, roadster \rangle,$$

We get $\Pi(t_{11} \not\prec t_2) = 0.7$ (corresponding to the interpretation $\langle \text{Ford, sedan} \rangle$ of t_2) and $\Pi(t_{11} \not\prec t_3) = 0.6$ (corresponding to the interpretation $\langle \text{Opel, roadster} \rangle$ of t_3).

For the second interpretation of t_1 , we get $\Pi(t_{12} \not\prec t_2) = 1$ (which corresponds to the interpretation $\langle \text{Ford, SUV} \rangle$ of t_2 that is completely possible and does not dominate t_{12}) and $\Pi(t_{12} \not\prec t_3) = 1$ (which corresponds to the interpretation $\langle \text{VW, roadster} \rangle$ of t_3) Finally:

$$\Pi(t_1) = \max(\min(1, \min(0.7, 0.6)), \min(0.8, \min(1, 1))) = 0.8. \quad \diamond$$

The degree of necessity that a tuple t from res be non-dominated by any other tuple t' from res is equal to 1 minus the degree of possibility that t be dominated by a tuple t' from res :

$$N(t) = 1 - \Pi'(t). \quad (5)$$

Example 4. Let us come back to the data and query from Example 3. Concerning the computation of $N(t_1)$, we get

- $\Pi(t_1 \prec t_2) = \min(1, 1) = 1$ which corresponds to the pair $1/\langle \text{Opel, roadster} \rangle$ for t_1 and $1/\langle \text{Ford, SUV} \rangle$ for t_2 .
- $\Pi(t_1 \prec t_3) = \min(1, 1) = 1$ corresponding to the pair $1/\langle \text{Opel, roadster} \rangle$ for t_1 and $1/\langle \text{VW, roadster} \rangle$ for t_3 .

Finally:

$$N(t_1) = 1 - \max(\Pi(t_1 \prec t_2), \Pi(t_1 \prec t_3)) = 1 - \max(1, 1) = 0. \diamond$$

For every tuple t from res , one computes $\Pi(t)$ and $N(t)$ according to Formulas (4) and (5). Since dominance is a Boolean property, one has (cf. Subsection 3.1):

$$N(t) > 0 \Rightarrow \Pi(t) = 1.$$

It is thus possible to rank-order the tuples using the following strategy: i) rank decreasingly according to N first, ii) for the tuples such that $N = 0$, rank decreasingly according to Π .

Example 5. Let us consider a possibilistic database describing used cars (cf. Table 3). This database is assumed to result from the fusion of different more or less reliable data sources. Let us consider the query:

```
select * from car
preferring (category = 'SUV' else category = 'roadster')
and (make = 'VW' else make = 'Ford' else make = 'Opel')
and (color = 'red' else color = 'black' else color = 'blue').
```

The evaluation yields:

- $\Pi(t_1) = 0.5$ and $N(t_1) = 0$.
- $\Pi(t_2) = 1$ and $N(t_2) = 0.3$.

Table 3. An extension of the possibilistic relation *car*

<i>make</i>	<i>category</i>	<i>color</i>	<i>horsepower</i>
t_1 {1/Opel, 0.8/VW}	roadster	{1/blue, 0.5/black}	{1/75, 0.8/85}
t_2 Ford	{1/SUV, 0.7/sedan}	red	110
t_3 {1/VW, 0.4/Opel}	roadster	{1/red, 0.8/blue}	{1/90, 0.8/95, 0.6/110}
t_4 {1/Opel, 0.3/Ford}	{1/roadster, 0.6/SUV}	red	{1/95, 0.8/100, 0.4/115}

- $\Pi(t_3) = 1$ and $N(t_3) = 0.4$.
- $\Pi(t_4) = 0.6$ and $N(t_4) = 0$.

Let us denote by $(N(t), \Pi(t))/t$ an element of the result. The ranked result is:

$$(0.4, 1)/t_3 \succ (0.3, 1)/t_2 \succ (0, 0.6)/t_4 \succ (0, 0.5)/t_1. \quad \diamond$$

5 About Query Evaluation

The algorithm which straightforwardly follows from Equations (II) and (5) is:

for every tuple t of r do

$\Pi(t) \leftarrow 0; \Pi_0 \leftarrow 0;$

for every interpretation π_i/t_i of t **do**

$\Pi_1 \leftarrow \pi_i; \Pi_2 \leftarrow 0;$

for every tuple $t' \neq t$ of r **do**

$\Pi_3 \leftarrow 0; \Pi_4 \leftarrow 0;$

for every interpretation π_j/t'_j of t' **do**

if $t_i \neq t'_j$ **then** $\Pi_3 \leftarrow \max(\Pi_3, \pi_j)$ **else** $\Pi_4 \leftarrow \max(\Pi_4, \pi_j)$ **endif;**

done;

$\Pi_1 \leftarrow \min(\Pi_1, \Pi_3); \Pi_2 \leftarrow \max(\Pi_2, \Pi_4);$

done;

$\Pi(t) \leftarrow \max(\Pi(t), \Pi_1); \Pi_0 \leftarrow \max(\Pi_0, \min(\pi_i, \Pi_2));$

done;

$N(t) \leftarrow 1 - \Pi_0;$

done.

Obviously, the data complexity of this algorithm is in $\theta(m^2)$ where m denotes the cardinality of r . The evaluation implies to compute all of the interpretations of t and t' for all the pairs (t, t') , but one can assume that the number of interpretations of an uncertain tuple is in general limited, and this computation does not entail any overhead in terms of disk access anyway. The good thing is that one does not have to compute the interpretations (worlds) of the relations themselves — but let us recall that this is because we treat Skyline queries as stand-alone queries. If a full compositional framework were to be defined, the story might be different, though (this is a matter for future work).

The previous algorithm can be improved by introducing some pruning conditions, which may reduce the number of calculations to be performed. For instance, when evaluating a tuple t , the innermost loop can be stopped as soon as both Π_3 and Π_4 equal 1. If the user is interested only in the necessity degrees, the evaluation of a tuple t can be stopped as soon as a tuple t' such that $\Pi(t \prec t') = 1$ is encountered (then $N(t)$ equals 0). Furthermore, if one looks for the tuples which belong to the skyline with a certainty degree at least equal to α (which probably corresponds to the most useful type of query in practice), it is possible to stop the evaluation of a tuple t as soon as a tuple t' such that $\Pi(t \prec t') > 1 - \alpha$ is encountered.

It remains to be studied whether some techniques proposed in the context of Skyline queries on classical data (for instance those based on presorting, see, e.g., [21]), or the algorithms proposed in the context of databases with missing values [15] or probabilistic data [16] could be adapted to the possibilistic database framework.

6 Skyline Queries Involving a Preselection

In the language *Preference SQL* [4], queries may involve a *where* clause (cf. Example [1]), which is used to filter out those tuples which are not concerned by the computation of the skyline. In the context considered here, the presence of such a selection condition may produce a relation involving *more or less possible/certain tuples* (if the selection criterion applies to attributes that take uncertain values). It is important to emphasize that the presence of such a selection condition in the query does not have any impact on the validity of the approach, as long as the attributes involved in the selection condition are *independent* from those concerned by the preferences. Otherwise, a strong representation system would be necessary in order to have a sound computation of the dominance degrees. In the context considered in this paper (i.e., stand-alone skyline queries), the only impact of a *where* clause is that one has to take into account the possibility and the necessity degrees attached to a tuple in the result of the selection during the computation of the skyline, which implies modifying Formulas (1) and (3).

Let us denote by $\Pi_{\in}(t)$ (resp. $N_{\in}(t)$) the possibility (resp. necessity) degree — whose computation will be detailed further — attached to a tuple t in the result of the preselection. Let us emphasize that $N_{\in}(t) = \alpha$ means that it is $1 - \alpha$ possible that tuple t does not exist in the considered relation. In the following formulas, $int(t)$ denotes the set of interpretations of t restricted to the attributes involved in the *preferring* clause. We get:

$$\Pi(t) = \min(\Pi_{\in}(t), \max_{\pi_i/t_i \in int(t)} \Pi(\pi_i/t_i)) \quad (6)$$

where

$$\Pi(\pi_i/t_i) = \min(\pi_i, \min_{t' \in res \setminus \{t\}} \Pi_c(t_i \not\prec t'))$$

with

$$\Pi_c(t_i \not\prec t') = \max(1 - N_{\in}(t'), \min(\Pi_{\in}(t'), \Pi(t_i \not\prec t')))$$

and $\Pi(t_i \not\prec t')$ is defined as in Equation (2).

Besides, the degree of possibility $\Pi''(t)$ that tuple t be dominated by any other tuple t' is now defined as:

$$\Pi''(t) = \max(1 - N_{\in}(t), \min(\Pi_{\in}(t), \max_{\pi_i/t_i \in \text{int}(t)} \Pi''(\pi_i/t_i)))$$

with

$$\Pi''(\pi_i/t_i) = \min(\pi_i, \max_{t' \in \text{res} \setminus \{t\}} \min(\Pi_{\in}(t'), \Pi(t_i \prec t')))$$

and $\Pi(t_i \prec t')$ is defined as in Equation (4).

As to Formula (5), it becomes:

$$N(t) = 1 - \Pi''(t). \quad (7)$$

For the sake of clarity, we restrict the scope to selection conditions ψ of the type

$$A_1 \theta_1 v_1 \wedge \dots \wedge A_p \theta_p v_p$$

where each θ_i is a comparator, and each v_i is a constant. Let us mention, however, that any type of selection condition could be dealt with (as long as the atomic predicates are logically independent). Let us denote by ψ_i the condition $A_i \theta_i v_i$. According to the axioms of possibility theory, we have:

$$\begin{aligned} \Pi_{\in}(t) &= \min(\Pi_{\psi_1}(t.A_1), \dots, \Pi_{\psi_p}(t.A_p)) \\ N_{\in}(t) &= \min(N_{\psi_1}(t.A_1), \dots, N_{\psi_p}(t.A_p)) \end{aligned}$$

where

$$\begin{aligned} \Pi_{\psi_i}(t.A_i) &= \max_{\pi_k/t_{ik} \in \text{int}(t_i)} \min(\pi_k, \mu_i(t_{ik})) \\ N_{\psi_i}(t.A_i) &= 1 - \max_{\pi_k/t_{ik} \in \text{int}(t_i)} \min(\pi_k, 1 - \mu_i(t_{ik})) \end{aligned}$$

with $\mu_i(t_{ik}) = 1$ if $t_{ik}.A_i \theta_i v_i$ holds, 0 otherwise.

Example 6. Let us consider the relation from Table 3 and the query (whose preference part is the same as in Example 5):

select * **from** *car* **where** *horsepower* ≥ 100
preferring (*category* = ‘SUV’ **else** *category* = ‘roadster’)
and (*make* = ‘VW’ **else** *make* = ‘Ford’ **else** *make* = ‘Opel’)
and (*color* = ‘red’ **else** *color* = ‘black’ **else** *color* = ‘blue’).

The result of the preselection is the relation $\{(1, 1)/t_2, (0.6, 0)/t_3, (0.8, 0)/t_4\}$ where an element is denoted by $(\Pi_{\in}(t_i), N_{\in}(t_i))/t_i$. The computation of the skyline yields:

- $\Pi(t_2) = 1$ and $N(t_2) = 0.3$.
- $\Pi(t_3) = 0.6$ and $N(t_3) = 0$.
- $\Pi(t_4) = 0.7$ and $N(t_4) = 0$.

Let us denote by $(\Pi(t), N(t))/t$ an element of the result. The ranked result is:

$$(1, 0.3)/t_2 \succ (0.7, 0)/t_4 \succ (0.6, 0)/t_3. \quad \diamond$$

7 Conclusion

In this paper, we have considered the situation where Skyline queries are submitted to a possibilistic database, i.e., an uncertain database where ill-known values are represented by possibility distributions. In this context, Skyline queries aim at computing the extent to which any tuple from a given relation is possibly/certainly not dominated by any other tuple. We have established the formulas allowing for the computation of the pair of degrees attached to each tuple of the result, and devised an algorithm which makes it possible to process such queries without computing the worlds of the possibilistic database involved (which would clearly be intractable).

Among the perspectives opened by this work, let us mention:

- the investigation of query optimization methods, with the particular aim of determining if some techniques suitable for Skyline queries in other contexts could be of any use in a possibilistic database framework;
- the extension of the present approach to a *graded* dominance relation [22,23,24];
- the conception of an optimized method for finding the tuples t' which α -certainly dominate a given tuple t , see [25] for similar queries in a probabilistic database context.
- the study of the feasibility of the definition of a strong representation system for possibilistic databases, including Skyline queries in its language.

References

1. Hadjali, A., Kaci, S., Prade, H.: Database preferences queries – A possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 291–310. Springer, Heidelberg (2008)
2. Bruno, N., Chaudhuri, S., Gravano, L.: Top-k selection queries over relational databases: mapping strategies and performance evaluation. *ACM Trans. on Database Systems* 27, 153–187 (2002)
3. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. *IEEE Trans. on Fuzzy Systems* 3, 1–17 (1995)
4. Kießling, W., Köstler, G.: Preference SQL — Design, implementation, experiences. In: Proc. of VLDB 2002, pp. 990–1001 (2002)
5. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. of ICDE 2001, pp. 421–430 (2001)
6. Chomicki, J.: Preference formulas in relational queries. *ACM Transactions on Database Systems* 28(4), 427–466 (2003)
7. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1, 3–28 (1978)
8. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. *Information Sciences* 34(2), 115–143 (1984)
9. Bosc, P., Pivert, O.: About projection-selection-join queries addressed to possibilistic relational databases. *IEEE Trans. on Fuzzy Systems* 13(1), 124–139 (2005)

10. Benjelloun, O., Das Sarma, A., Halevy, A., Theobald, M., Widom, J.: Databases with uncertainty and lineage. *VLDB Journal* 17(2), 243–264 (2008)
11. Ré, C., Dalvi, N., Suciu, D.: Efficient top-k query evaluation on probabilistic data. In: *Proc. of ICDE 2007*, pp. 886–895 (2007)
12. Soliman, M., Ilyas, I., Chang, K.C.: Top-k query processing in uncertain databases. In: *Proc. of ICDE 2007*, pp. 896–905 (2007)
13. Zhang, X., Chomicki, J.: On the semantics and evaluation of top-k queries in probabilistic databases. In: *Proc. of DBRank 2008*, pp. 556–563 (2008)
14. Bosc, P., Pivert, O.: From Boolean to fuzzy algebraic queries in a possibilistic database framework. In: *Proc. of the 13th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2004)*, pp. 25–29 (2004)
15. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline query processing for incomplete data. In: *Proc. of ICDE 2008*, pp. 556–565 (2008)
16. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: *Proc. of VLDB 2007*, pp. 15–26 (2007)
17. Hüllermeier, E., Vladimirskiy, I., Prados Suárez, B., Stauch, E.: Supporting case-based retrieval by similarity skylines: Basic concepts and extensions. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) *ECCBR 2008. LNCS (LNAI)*, vol. 5239, pp. 240–254. Springer, Heidelberg (2008)
18. Dubois, D., Prade, H.: *Possibility Theory*. Plenum, New York (1988)
19. Prade, H.: Lipski’s approach to incomplete information databases restated and generalized in the setting of Zadeh’s possibility theory. *Information Systems* 9(1), 27–42 (1984)
20. Imielinski, T., Lipski, W.: Incomplete information in relational databases. *J. of the ACM* 31(4), 761–791 (1984)
21. Bartolini, I., Ciaccia, P., Patella, M.: Efficient sort-based skyline evaluation. *ACM Trans. Database Syst.* 33(4), 1–49 (2008)
22. Zadrozny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: *Proc. of DEXA 2006 Workshops*, pp. 415–419 (2006)
23. Goncalves, M., Tineo, L.J.: Fuzzy dominance skyline queries. In: Wagner, R., Revell, N., Pernul, G. (eds.) *DEXA 2007. LNCS*, vol. 4653, pp. 469–478. Springer, Heidelberg (2007)
24. Hadjali, A., Pivert, O., Prade, H.: On different types of fuzzy skylines. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) *ISMIS 2011. LNCS (LNAI)*, vol. 6804, pp. 581–591. Springer, Heidelberg (2011)
25. Fung, G.P.C., Lu, W., Du, X.: Dominant and K nearest probabilistic skylines. In: Zhou, X., Yokota, H., Deng, K., Liu, Q. (eds.) *DASFAA 2009. LNCS*, vol. 5463, pp. 263–277. Springer, Heidelberg (2009)

A Fuzzy Valid-Time Model for Relational Databases within the Hibernate Framework

Jose Enrique Pons, Olga Pons Capote, and Ignacio Blanco Medina

Department of Computer Science and Artificial Intelligence
Universidad de Granada

Escuela Técnica Superior de Ingeniería Informática

C/Periodista Daniel Saucedo Aranda s/n

E-18071, Granada-Spain

{jpons,opc,iblanco}@decsai.ugr.es

Abstract. Time in databases has been studied for a long time. Valid time databases capture when the objects are true in the reality. The proposed model allows both representing and querying time in a fuzzy way. The representation and the underlying domain are defined as well as some fuzzy temporal operators. The implementation of the model is developed within the Hibernate framework. The Hibernate framework acts as an abstraction for the running database. Therefore, any relational database supported by the framework can now represent fuzzy valid time in its schema.

1 Introduction

In the present work we introduce a fuzzy temporal model for fuzzy relational databases. The model is database independent, due the representation, which is non-dependent on any temporal data defined in SQL date data types but on numeric representation. The implementation is also database independent. The Hibernate framework [13] is mainly for object-relational mapping. The main reason is that many applications work with objects at application level while a relational model remains at database level. Hibernate presents the concept of *dialect* that is the abstraction for the current database. Changing the running database is as easy as changing the dialect (and some configuration options). Therefore, any relational database supported by the framework can be extended into a fuzzy database and also into a fuzzy valid-time database with the proposed implementation.

The structure of the paper will be the following: Section 2 introduces the basic concepts for temporal aspects. A temporal model should define such temporal elements. Dealing with temporal databases has several problems which are presented in the section too.

The model focuses on fuzzy valid time which is proposed in section 3. The section also explains how the fuzzy validity period is built and the underlying numerical domain. Section 4 explains the implementation within the Hibernate framework, including some examples of use and sample queries.

Finally, further research work and applications are discussed in section 5.

2 Temporal Databases

As described in the *Consensus glossary for temporal database concepts* [6] a **temporal database** management system (temporal DBMS or TDBMS) is a database that manages some aspects of time in its schema, not counting user-defined time (an uninterpreted attribute; supported in the standard SQL2 [18]). The shortest duration of time supported by the database is called **chronon**. There exists two ways to represent the time: as points either as intervals. [2] is a study between interval-based and point-based temporal data models. These representations are also extended in the fuzzy way by means of both fuzzyfication of the time point [5] or the time interval [11] and in a more general way in [23]. The comparison between two time intervals has been studied by Allen [1]. Time granularity is also associated with the representation of the time. A **granularity** is the result of partitioning on the set of chronons. The conversion among granularities is a common issue within temporal databases [14]. Indeed, granularity is considered the basis in [3], [4].

Three types of time which are handled specifically by a TDBMS:

- **Transaction time**: The time when the fact is stored in the database.
- **Valid time**: The time when the fact is true in the modelled reality.
- **Decision time** ([20]): The time when an event was decided to happen.

According to the time managed, the database model can be classified into **transaction time** [15], [12], **Valid time**, **bi-temporal** [24] (both valid and transaction time) or **tri-temporal** [20] (valid, transaction and decision time).

3 Fuzzy Valid-Time Model

Valid time represents the time when the tuple is true in the modeled reality. In classical temporal databases, valid time is usually represented by an interval. Two values are needed: **VST**: Valid Starting time and **VET**: Valid Ending time. If one or both time points are known with vagueness, then the model manages fuzzy time. This model is based on the proposal in [11]. The two ill-known values are summarized into one fuzzy number called Fuzzy Validity Period, **FVP**.

3.1 Fuzzy Validity Period

The fuzzy validity period is built from two values. These values are not precisely known. The FVP may be build in two ways:

1. **Preserving the imprecision**: This approach is based on preserving the imprecision both for the starting and ending point. The resulting fuzzy interval is less intuitive but more realistic. (See Fig. 1). The proposal is defined in [11]. Due to its properties, the first option is developed in the implementation.

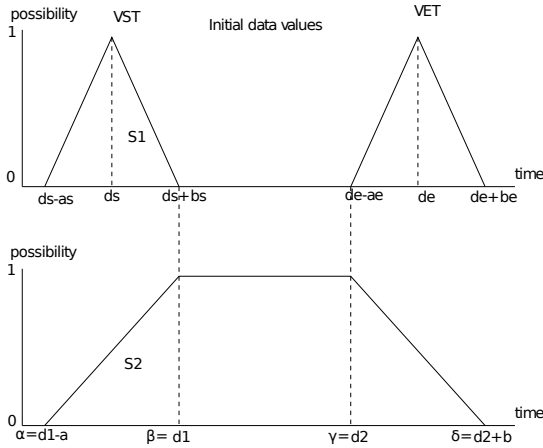


Fig. 1. Transformation to obtain the fuzzy validity period that preserves the imprecision

To preserve the imprecision, the two areas S_1 and S_2 must be equal (see fig. 1):

$$S_1 = S_2 \Rightarrow \frac{(d_s + b_s) - (d_s - a_s)}{2} = \frac{d_1 - (d_1 - a)}{2} \tag{1}$$

Where $d_1 = d_s + b_s$ and $d_2 = d_e + a_e$. A trapezoidal possibility distribution is usually represented by 4 values: $[\alpha, \beta, \gamma, \delta]$. Thus, $[d_1 - a, d_1, d_2, d_2 + b]$ is the FVP in fig. 1.

2. **Convex hull:** The validity period is built in an intuitive way by computing the convex hull of the union of both fuzzy values. For the given two ill known distributions VST and VET shown in fig. 1, the convex hull is the trapezoid $[d_s - a_s, d_s, d_e, d_e + b_e]$.

3.2 Underlying Domain: Julian Day Number

The underlying domain to represent the FVP is the Julian Day Number, **JDN**. It is a counter which value is incremented in one unit every day since 1 January 4713 B.C noon. The main reasons for choosing this representation are two: first of all, its representation as unique number, allows to simplify some computations.

The second reason is that current implementations of the *date* data type in SQL differ on the semantics of the temporal operators e.g. in MySQL databases, adding 1 to a date means adding a second, while in PostgreSQL means adding one day. Also the internal representation is different for each database.

The conversion formulas between the current Gregorian calendar to the corresponding JDN are explained in [7]. The computation of the membership degree for a date x (in JDN format) with respect to the FVP, T is done by the following formula:

$$\mu_T(x) = \begin{cases} 0, & \text{if } x \leq \alpha, x \geq \delta \\ 1, & \text{if } x \in [\beta, \gamma] \\ \frac{x-\alpha}{\beta-\alpha} & \text{if } x \in (\alpha, \beta] \\ \frac{\delta-x}{\delta-\gamma} & \text{if } x \in (\gamma, \delta] \end{cases} \quad (2)$$

3.3 Fuzzy Temporal Operators

The model defined works with fuzzy temporal validity period (FVP), which are intervals. Thus, some temporal operators are defined in the model in [11]. The operators are easily defined by FSQL sentences defined by in [8]. There are several proposals ([19], [23]) for the fuzzyfication of the Allen's operators [1].

Table 1 shows the implementation for the fuzzy temporal operators supported by the temporal FSQL which are implemented by means of the basic fuzzy operators (see table 2).

Also, two unary operators are provided (START and END) to get the approximate starting and ending point of an FVP. For a given FVP, $J = [\alpha, \beta, \gamma, \delta]$, the values for START and END are:

- START (J) = $[\alpha, \alpha + \beta/2, \alpha + \beta/2, \beta]$
- END (J) = $[\gamma, \gamma + \delta/2, \gamma + \delta/2, \delta]$

4 Implementation within Hibernate Framework

The Hibernate Framework [13] is an open source framework for object-relational mapping. The framework needs an object oriented language (Java or .NET) and a relational database. An object-oriented query language called HQL (Hibernate Query Language) is also provided.

The main advantage of this framework is that it is cross-platform and also non-database dependent. The first one is because it is written in Java and the second one is because of an abstraction called *dialect*. The dialect is an abstraction on the database layer, thus changing the running database is as easy as changing to the corresponding dialect.

The temporal model is built on top of a fuzzy layer developed by the researchers [21].

Two aspects are implemented in the present work:

1. **Representation** of fuzzy validity period FVP: Now, any relational database supported by the framework can deal with FVPs. These FVPs are represented as objects in the interface between Hibernate and the application. Hibernate translates the object-oriented representation of the FVPs to a relational representation. Finally, the *dialect* customizes the standard SQL representation into specific SQL representation for the running database.

Table 1. Implementation of fuzzy interval operators

Operator Name	Implementation
BEFORE (I,J)	$\begin{cases} 1, & \text{if } I_\beta \leq J_\alpha \\ \frac{I_\alpha - J_\beta}{(J_\alpha - J_\beta) - (I_\beta - I_\alpha)}, & \text{if } I_\beta > J_\alpha \text{ and } I_\alpha < J_\beta \\ 0, & \text{otherwise } (I_\alpha \geq J_\beta) \end{cases}$
OVERLAPS (I, J)	$\sup_t (\min (I (t), J (t)))$
EQUALS (I, J)	$\min (CONTAINS (I, J), CONTAINS (J, I)) = \min (\inf_t \{\max (1 - I (t), J (t))\}, \inf_t \{\max (1 - J (t), I (t))\})$
CONTAINS (I, J)	$\inf_t \{\max (1 - I (t), J (t))\}$

Table 2. Fuzzy comparators implemented in Fuzzy SQL

Possibility	Necessity	Meaning:
FEQ	NFEQ	(Possibly/Necessarily) Fuzzy equal
FGT	NFGT	(Possibly/Necessarily) Fuzzy greater than
FGEQ	NFGEQ	(Possibly/Necessarily) Fuzzy greater or equal
FLT	NFLT	(Possibly/Necessarily) Fuzzy less than
FLEQ	NFLEQ	(Possibly/Necessarily) Fuzzy less or equal
MGT	NMGT	(Possibly/Necessarily) Much greater than
MLT	NMLT	(Possibly/Necessarily) Much less than

Table 3. Implementation of fuzzy temporal operators

Operator Name	FSQL implementation
BEFORE (I, J)	I FGEQ J
OVERLAPS (I, J)	I FEQ J
EQUALS (I, J)	min(I NFEQ J, J NFEQ I)
CONTAINS (I, J)	I NFEQ J

2. **Flexible Querying:** The object-oriented query language, HQL is modified to query the new fuzzy data types by means of fuzzy operators (see table 2). Now, it is possible to build flexible temporal queries by using the implemented temporal operators shown in table 3.

This section is organized as follows: first of all, 4.1 is a brief explanation about the architecture of a typical Hibernate application. Then the representation of fuzzy types and specially, the representation of the FVP is discussed in 4.2. The modification of the query engine of Hibernate to support flexible temporal querying is explained in 4.3. Finally, the section finishes with an example of both definition and querying a fuzzy valid time database.

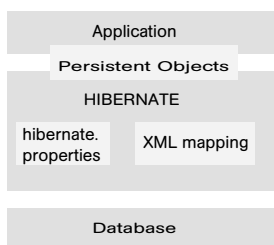


Fig. 2. Base Hibernate application

4.1 Architecture

The main architecture for a Hibernate application has three main layers (fig. 2):

- The **application layer:** Where the application performs CRUD (CReate, UUpdate, and DDelete) operations.
- The **Hibernate layer:** Where the object-relational mapping is done.
- The **database layer:** Where the running database is. The framework supports most of the major database systems like Oracle, MySQL, PostgreSQL among others.

The implementation is on top of the Hibernate layer. Figure 3 illustrates the minimal design of the model.

Therefore, the application works with objects. The database works with tables (entities and relationships). Thus, the Hibernate layer makes the object-relational mapping and the customization of the code sent to the database. The Hibernate FSQL allows the application to work (representation and querying) with objects with fuzzy attributes. Finally, the Hibernate FVP layer allows the application to work with objects with both fuzzy attributes and fuzzy validity periods. The fuzzy objects are translated to a relational representation in the database.

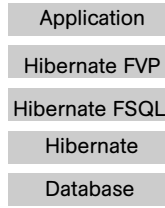


Fig. 3. Minimal hibernate architecture for fuzzy temporal representation

4.2 Representation

The object-oriented representation of the fuzzy data types is based on the Fuzzy Knowledge Representation Ontology (*FKRO*) [16]. The ontology is built on the top of the GEFRED [17] model and its interface FIRST [9, 10]. The reference implementation in the Oracle database is called FSQL [9].

Figure 4 shows the basic UML diagram for the representation. A fuzzy meta-domain is composed by 3 elements:

1. A set of fuzzy values. Two main fuzzy types are described below: fuzzy values with an underlying ordered domain and fuzzy values with a non-ordered underlying domain.
2. A set of constraints that restrict the set of possible values in the domain. The constraints may be either the fuzzy constants UNDEFINED, UNKNOWN, NULL or the fuzzy types like INTERVAL, CRISP, TRAPEZOIDAL, LABEL, defined in FSQL [9].
3. A set of labels. A label is a tag for a fuzzy value. Thus, 'Cheap' is a label associated with the fuzzy value represented by a possibility distribution.

has a set of values, a set of constraints and a set of labels that are associated to different values. There are defined two fuzzy meta-domains:

- **Fuzzy Domain Non-Ordered:** This meta-domain allows the definition of new fuzzy domains over an underlying non-ordered domain. E.g. The fuzzy domain of the hair color.
- **Fuzzy Domain Ordered:** This meta-domain allows the definition of new fuzzy domains over an underlying ordered domain. E.g. The fuzzy domain that models the prices of restaurants. Several labels may be defined like 'Cheap', 'Middle-priced', 'Expensive'.

Each fuzzy meta-domain contains a set of fuzzy values. These fuzzy values are of two types:

- **Non-ordered fuzzy type:** This fuzzy type models objects with an underlying non-ordered domain. E.g. The colour 'Red' for the hair color. The corresponding type in the GEFRED model is type 3.

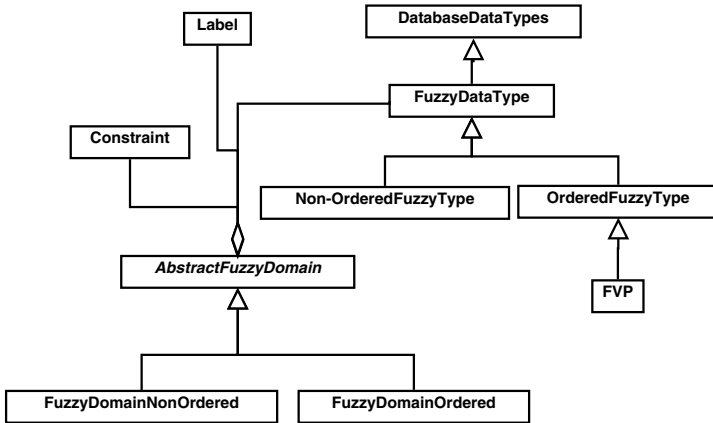


Fig. 4. UML diagram for fuzzy data types

- **Ordered fuzzy type:** This fuzzy type models objects with an underlying ordered domain. E.g. The label ‘Cheap’ represented as a triangular possibility distribution between 0 and 25 euros. The corresponding type in the GEFRED model is type 2.

The Fuzzy Validity Period FVP is represented in the framework as mentioned in section 3.1. The underlying ordered domain is the Julian Day Number (JDN). Thus, the representation of this data type on the framework is based on a fuzzy data type with an underlying ordered domain.

A fuzzy underlying domain is defined with some operations to convert between Java dates (usually in Gregorian calendar format) and the Julian Day Number with the formula explained in section 3.2 and the algorithm 7.

Table 4 is the relational representation of the FVP in a database. 5 columns are needed: The first one (Fuzzy Type, FT) stores the subtype for the object. Four values are allowed in order to represent a fuzzy validity period: the constants UNKNOWN, UNDEFINED and NULL and the trapezoidal possibility distribution. The following four columns (from F1 to F4) store the values for a given element.

Table 4. Relational representation for fuzzy validity period, a type 2 attribute. Note that N is the abbreviation for NULL constant.

Fuzzy Type	FT	F1	F2	F3	F4
UNKNOWN	0	N	N	N	N
UNDEFINED	1	N	N	N	N
NULL	2	N	N	N	N
TRAPEZ	7	α	β	γ	δ

4.3 Querying of Fuzzy Data Types

Hibernate provides a language based on SQL called HQL (Hibernate Query Language). The main advantage of this query language is the portability: The query in HQL is written at application level, then, it is translated into an abstract syntax tree (AST) which is finally customized into specific SQL sentences for the underlying database. Figure 5 shows the entire process.

The HQL language is modified to support the fuzzy operators that perform fuzzy comparisons (table 2) between fuzzy attributes. The operators are implemented in a declarative way: by using sentences in SQL (CASE - WHEN structure), instead of a procedural implementation. This is the key for the portability since Hibernate customizes the SQL implementation for any supported database.

The workflow for a sentence in HQL with a fuzzy operator is the following:

1. The HQL sentence is analyzed and translated into the AST representation.
2. The AST is analyzed. When a node is a fuzzy operator, then the node and the operands are replaced by the implementation in SQL.
3. Finally, the SQL sentence is customized by the dialect for the running database.

The fuzzy temporal operators implemented are the mentioned in table 3: BEFORE, OVERLAPS, EQUALS and CONTAINS. This operators are implemented as alias of the previous implemented fuzzy operators (in the case of the BEFORE operator) or by some simple combination of them (like in the case of the EQUALS operator).

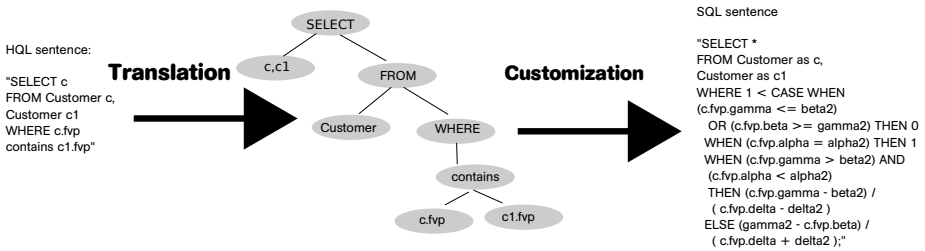


Fig. 5. Translation from HQL to customized SQL sentences. From the left to the right, the HQL query is translated into an AST. The dialect customizes the AST for the running database into specific SQL sentences.

Example 1. Consider a company which stores data about its employees. The data is stored in a fuzzy valid time database (see table 5). Each time the relation is updated, a new row with an updated version of the data is stored. The starting and the ending points of the validity period are not precisely known, and are represented by the FVP given in the $[\alpha, \beta, \gamma, \delta]$ format explained in section 3.1. For simplicity, the values for each element of the FVP are shown in their corresponding gregorian calendar but are stored in the JDN format mentioned in section 3.2.

Table 5. The relation employees with fuzzy validity periods (FVP)

ID	Name	Salary	BossID	Cat.	FVP
001	Josh	1200	002	C	[10/10/2009,27/10/2009,19/10/2010,27/10/2010]
001	Josh	1500	002	B	[19/10/2010,27/10/2010,-,-]
002	Robert	800	005	A	[14/05/2007,25/05/2007,17/01/2008,30/01/2008]
002	Robert	1200	005	A	[17/01/2008,30/01/2008,24/05/2009,30/05/2009]
002	Robert	1400	003	A	[24/05/2009,30/05/2009,28/10/2010,30/10/2010]
003	Alex	2100	-	A+	[02/05/2007,15/05/2007,-,-]
004	Tyna	1300	002	A+	[25/07/2009,30/7/2009,15/10/2009,25/10/2009]
005	Rose	2300	003	A+	[25/09/2010,30/09/2010,25/02/2011,30/02/2011]

Class definition. An entity class represents a table in the database. A field on an entity class represents one or several table columns. The corresponding class declaration for the table 5 is:

```
public class Employee implements Serializable {

private String ID; // primary key: ID
private String name; // name of the employee
private Double salary; // salary
private Employee boss; // boss
private Category category; // category
private FVP fvp; // fuzzy validity period
}
```

Querying. Consider the user wants to make the following query:

"Find all the employees with the boss 002 during the same period of time."

The translation of this query into HQL is the following:

```
SELECT e,f
FROM Employee e, Employee f
WHERE e.ID="002" AND e.ID<>f.ID AND e.fvp EQUALS f.fvp;
```

The framework translates the HQL query to SQL sentences as explained in section 4.3. Then, the sentences are sent to the database. The result set of the query returned by the database is mapped backwards to objects by Hibernate. Table 6 shows the result set for the query, and the compatibility for the result.

Table 6. The relation employees with fuzzy validity periods (FVP)

e.ID	e.name	f.ID	f.name	Comp.
001	Josh	004	Tyna	0,55

5 Conclusions and Further Research Work

The presented work is an extension built on the top of an implementation previously developed by the researchers. The temporal extension allows both representation and querying by means of fuzzy temporal operators. The main contribution of this work is therefore, the representation and querying for the fuzzy validity period in any relational database supported by the framework. The main drawback for the portability is that, outside the framework, the DBMS is not able to manage with the fuzzy valid time model. This is not such a big deal since over the last few years, the trend is to develop the business layer outside the framework.

Further research work includes applications where the time is not precisely known. Time is useful in geographic information systems. One of the main applications is observing the changes within the time for a geographical place. Also in historical databases, the time is usually not precisely known. Further work in applications with these two main lines will be done with the current implementation within the Hibernate framework. From a theoretical point of view, the researchers are studying new representations for fuzzy time, not only fuzzy valid time. This includes new representations by means of rough sets and triangular models [22].

Acknowledgements. The researchers are supported by the grant BES-2009-013805 within the research project TIN2008-02066: *Fuzzy Temporal Information treatment in relational DBMS*, and the project P07-TIC-03175: *Representation and Handling of Imperfect Objects in Data Integration Problems*.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 832–843 (1983)
2. Bohlen, M., Busatto, R., Jensen, C.: Point-versus interval-based temporal data models. In: *Proceedings of the 14th International Conference on Data Engineering*, pp. 192–200 (February 1998)
3. Van der Cruyssen, B., De Caluwe, R., De Tré, G.: A theoretical fuzzy time model based on granularities. In: *EUFIT 1997*, pp. 1127–1131 (September 1997)
4. De Tre, G., De Caluwe, R., Van Der Cruyssen, B., Van Gyseghem, N.: Towards temporal fuzzy and uncertain object-oriented database management systems. In: *1997 Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 1997*, pp. 68–72 (September 1997)
5. Dubois, D., Prade, H.: Processing fuzzy temporal knowledge. *IEEE Transactions on Systems, Man and Cybernetics* 19(4), 729–744 (1989)
6. Dyreson, C., et al.: A consensus glossary of temporal database concepts. *SIGMOD Rec.* 23, 52–64 (1994)
7. Fliegel, H.F., van Flandern, T.C.: Letters to the editor: a machine algorithm for processing calendar dates. *Commun. ACM* 11, 657 (1968)
8. Galindo, J., Medina, J.: Ftsql2: Fuzzy time in relational databases. In: *Proceedings EUSFLAT 2001*, pp. 47–50 (2001)

9. Galindo, J., Medina, J.M., Pons, O., Cubero, J.: A server for fuzzy SQL queries. In: Andreassen, T., Christiansen, H., Larsen, H.L. (eds.) FQAS 1998. LNCS (LNAI), vol. 1495, pp. 164–174. Springer, Heidelberg (1998)
10. Galindo, J., Urrutia, A., Piattini, M.: Fuzzy Databases: Modeling, Design and Implementation. Idea Group, Chocolate Avenue, Suite 200, Hershey PA 17033 (2006)
11. Garrido, C., Marin, N., Pons, O.: Fuzzy intervals to represent fuzzy valid time in a temporal relational database. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 17(suppl. 1), 173–192 (2009)
12. Jensen, C.S., Mark, L., Roussopoulos, N.: Incremental implementation model for relational databases with transaction time. *IEEE Trans. on Knowl. and Data Eng.* 3, 461–473 (1991)
13. King, G., Bauer, C., Andersen, M.R., Bernard, E., Ebersole, S., Ferentschik, H.: Hibernate Reference Documentation, 3.6.0.cr2 edn., <http://www.hibernate.org/docs>
14. Lin, H., Jensen, C.S., Ohlen, M.H.B., Busatto, R., Gregersen, H., Torp, K., Snodgrass, R.T., Datta, A., Ram, S.: Efficient conversion between temporal granularities (1997)
15. Rowe, L., Stonebreker, M.: The postgres papers (June 1987)
16. Martínez Cruz, C.: Sistema de gestión de bases de datos relacionales difusas multipropósito. Una ontología para la representación del conocimiento difuso. Ph.D. thesis, Universidad de Granada (2008)
17. Medina, J., Pons, O., Cubero, J.: Gefred. a generalized model of fuzzy relational databases. *Information Sciences* 76(1-2), 87–109 (1994)
18. Melton, J., Simon, A.R.: Understanding the new SQL: a complete guide. Morgan Kaufmann Publishers Inc., San Francisco (1993)
19. Nagypál, G., Motik, B.: A fuzzy model for representing uncertain, subjective, and vague temporal knowledge in ontologies. In: Chung, S., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 906–923. Springer, Heidelberg (2003)
20. Nascimento, M.A., Eich, M.H.: Decision time in temporal databases. In: Proceedings of the Second International Workshop on Temporal Representation and Reasoning, pp. 157–162 (1995)
21. Pons, J.E., Pons, O., Blanco Medina, I.: An open source framework for fuzzy representation and querying in fuzzy databases. In: Nunes, M.B., Pedro Isaas, P. P. (eds.) Proceedings of the IADIS International Conference Informations Systems (March 2011)
22. ng, Y., Asmussen, K., Delafontaine, M., De Tré, G., Stichelbaut, B., De Maeyer, P., Van de Weghe, N.: Visualising rough time intervals in a two-dimensional space. In: Proceedings 2009 IFSA World Congress/EUSFLAT Conference (July 2001)
23. Schockaert, S., De Cock, M., Kerre, E.: Fuzzifying allen’s temporal interval relations. *IEEE Transactions on Fuzzy Systems* 16(2), 517–533 (2008)
24. Snodgrass, R.: The temporal query language tquel. In: Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, PODS 1984, pp. 204–213. ACM, New York (1984)

On the Use of a Fuzzy Object-Relational Database for Retrieval of X-rays on the Basis of Spine Curvature Pattern Similarities

Sergio Jaime-Castillo¹, Juan M. Medina¹,
Carlos D. Barranco², and Antonio Garrido¹

¹ Department of Computer Science and Artificial Intelligence, University of Granada
C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

{sjaime,medina,a.garrido}@decsai.ugr.es

² Division of Computer Science, School of Engineering, Pablo de Olavide University
Ctra. Utrera km. 1, 41013 Seville, Spain

cbarranco@upo.es

Abstract. In medical practice radiologists use X-rays to diagnose and treat scoliosis, which is a medical condition that affects the spine. Doctors usually compare patients' X-rays to other images with known diagnosis so that they can propose a similar treatment. Since digital medical images are usually stored in large databases, an automatic way to retrieve them could truly help radiologists. In this paper we show how a Fuzzy Object-Relational Database System can be used to provide flexible querying mechanisms to retrieve the similar images. We present the main system capabilities to represent and store curvature pattern descriptions and how queries on them are solved.

Keywords: Fuzzy Databases, Flexible querying, CBIR, Medical images.

1 Introduction

Images are a fundamental tool in health care for diagnosis, clinical studies, research and learning. The diagnostic task generates a large amount of images that must be archived for future evaluations. Traditional Picture Archiving and Communication Systems (PACS) solve the problem of storing digital images but do not provide mechanisms to retrieve them on the basis of their content. Content-Based Image Retrieval (CBIR) [14] is the application of computer vision techniques to the problem of digital images search in large databases. The first image retrieval approaches were based on captions and textual descriptors collected by humans. Nowadays, image retrieval systems based on visual information outperform the textual-based ones by using features such as color, texture or shape, which are automatically extracted from images. Also, there are several attempts to incorporate more semantic knowledge to CBIR, to reduce the “semantic gap” between what queries represent (low-level features) and what the user thinks [15]. In this regard, a very important point to take into account is

the imprecision in the feature descriptions, as well as the storage and retrieval of that imprecise data. To deal with this vagueness, some interesting approaches introduce the use of fuzzy logic in the feature extraction, as well as in the retrieval process [25,8,10,12,9,22,24]. Some of these fuzzy approaches also allow to perform queries on the basis of linguistic terms, enriching the semantics of the user's query. The results obtained by CBIR techniques are more significant when applied to a specific domain or application area, as knowledge on the domain and on the image characteristics helps the process of extracting the relevant features for this specific area of application. Health care is an application area that may benefit from the CBIR techniques [19]. If we focus the CBIR techniques on the analysis of a certain pathology we can get high level features by processing certain types of images. For diagnostic purposes, radiologists are usually interested in retrieving spine images that present a similar curvature pattern to a given one, which implies the need of providing mechanisms to perform flexible queries based on the parameters describing the spine curvature.

Fuzzy Database Management Systems (FDBMS) can be useful in CBIR for retrieval purposes. Since these systems are able to represent fuzzy data and implement comparators on them, they can be used to provide flexible content based retrieval. There is a wide variety of proposals for fuzzy data handling in databases [16,17,6] but in general these models and/or implementations do not have enough modeling power and performance for image indexing applications. However, there are some proposals based on Fuzzy Object-Oriented Database models [2], like [20,8,13], and on logic databases like [23], that provide some capabilities to manage representation and retrieval of images based on general feature descriptors. In [5,1] we introduce a Fuzzy Object-Relational Database System (FORDBMS) model that evolves classical fuzzy databases models to incorporate object-oriented features for a powerful representation and handling of data, fuzzy or not. This paper shows how this "general purpose" FORDBMS is suitable for easy image representation and retrieval, using the fuzzy descriptors obtained by means of computer vision algorithms.

The remainder of the paper is organized as follows: section 2 deals with the feature extraction process; section 3 presents the most important features of the FORDBMS for storing and retrieving fuzzy data; section 4 shows how flexible queries are solved and the mechanisms provided to tune its behavior; finally, we present some conclusions and future lines of work.

2 Image Characterization

Scoliosis is a medical condition that causes vertebral rotation and crushing as well as spine lateral curvature. Depending on the severity and progression of the curvature, treatment may be necessary. Possible treatments include observation, braces or, in extreme cases, surgery. Doctors usually compare the curvature pattern present in a patient's X-ray to those of other patients already treated. If they are alike, similar treatments will be proposed. A curvature pattern is nothing else but a sequence of curves present in the patient's spine. We use

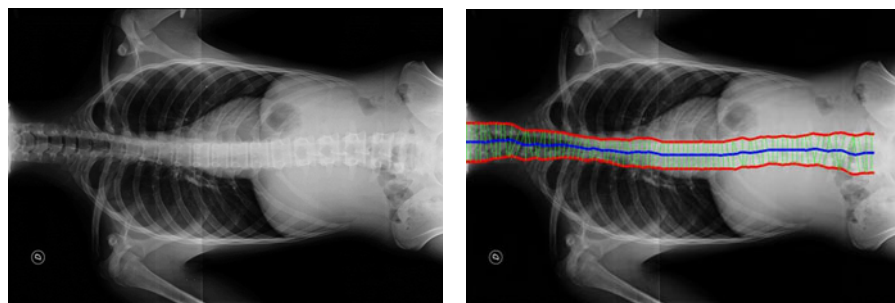
the position of the spine axis (the line drawn along the center of the spine, in blue in Fig. 1(b)) in the image to represent the curvature patterns. Note that we only need the points that define each curve, that is to say, the starting point, the apical point and the ending point. Those coordinates are obtained by automatically segmenting the spine from the X-ray.

Image segmentation is a complex process whereby we try to partition a digital image into various sets of pixels that, hopefully, correspond to objects present in the image. Many techniques have been developed to perform automatic segmentation of images. However, they are usually restricted to certain domains and are not general purpose techniques. Some of the techniques successfully applied to medical images include Active Shape Models (ASM) [4] and Snakes [11].

In [7], a novel method based on Snakes is proposed to segment the spine. This algorithm provides us with the position of the spine axis in the image alongside its derivatives. Traditional Snakes use an integral function that represents the energy of a contour in an image: the lower that energy is, the closer it is to the searched feature. The mathematical framework to do this is complex and requires a great deal of computations. Instead, dynamic programming (which is far quicker) is used in the paper as minimization technique. Additionally, this model uses a hierarchical multi-resolution approach and Catmull-Rom interpolation to obtain better results. Another version of the algorithm that does not use interpolation at all is proposed, although its performance regarding accuracy is slightly worse. It is, however, much faster than the one using interpolation since it does not need to compute any derivatives. The major difference between both algorithms is that the version using interpolation produces a smoother curve as spine axis, which is a desirable quality in our approach. It is important to note that the algorithm works in a completely automatic manner without any preprocessing stage. Nevertheless, slightly better results are obtained when cropping is applied to the images to remove parts of the image that do not give us any useful information, namely the skull and the legs. Besides being useless for our purposes, they can badly influence the segmentation algorithm producing undesired edges wherein Snakes can get trapped.

As a result of applying that algorithm on a dataset of 59 images we obtain, for each image, a set of points that represents the central axis of the spine. We also obtain the value of the first and second derivatives of the spine line. The results of the algorithm are shown in Fig. 1(b), where the spine axis is represented by the blue line and the red ones represent the spine boundaries.

Taking into account the information provided by the segmentation algorithm, we use the derivative values to look for inflection points and maxima in the spine axis. Inflection points are taken as starting and ending points for the curves and maxima and minima are taken as apical vertebrae. Relative position of maxima and inflection points allows us to determine curves' orientation. To be precise, a curve consist of a sequence of an inflection point, a maximum or a minimum and another inflection point. We only store their horizontal coordinates, but we use the vertical ones to check if the curve is oriented to the right (the vertical coordinate of the apical point is greater than the one of the inflection points) or

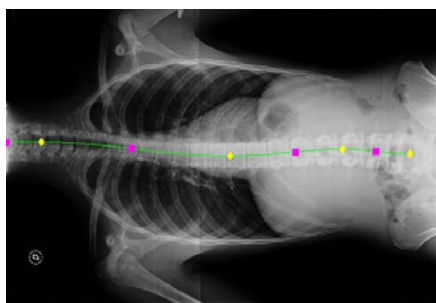


(a) Original image

(b) Segmented image



(c) First smoothing stage: axis subsampled every 50 points and smoothed



(d) Final spine axis: axis subsampled every 100 points and smoothed. Magenta squares are inflection points and yellow rhombi are maxima and minima ones

Fig. 1. Results of the image characterization algorithm

to the left (the vertical coordinate of the apical point is lower than the one of the inflection points). We also calculate the angle between the normal lines to the spine axis at the inflection points by means of the derivatives provided by the segmentation algorithm. This angle is known as Cobb angle [3] and is widely used in treatment of scoliosis. Although it is stored in the database as numerical data, it is not currently used to retrieve the images.

Yet before we can obtain such information, a data preprocessing stage is needed. That preprocessing is performed to obtain a curve as smooth as possible, since that will remove many spurious sub-curves that may appear in the original data (due to local maxima or minima as seen in Fig. 1(b)).

The previous step is done by using smoothing splines, as proposed by Reinsch in [21]. Our approach is divided in two steps: in the first one we subsample the data every 50 points and obtain the smoothing spline by means of the aforementioned algorithm (the results are shown in Fig. 1(c)); an additional step is

performed to ensure maximum smoothing in which the resulting data from the previous step is subsampled every 100 points this time. Again, we obtain the smoothing spline (the results are shown in Fig. 1(d)) and the resulting data is taken as the starting point for our retrieval approach.

3 The Fuzzy Object-Relational Management System

In [5,1] we introduce the strategy of implementation of our FORDBMS model, that is based on the extension of a market leader DBMS (Oracle®) by using its advanced object-relational features. This strategy let us take full advantage of the host DBMS features (high performance, scalability, etc.) adding the capability of representing and handling fuzzy data provided by our extension.

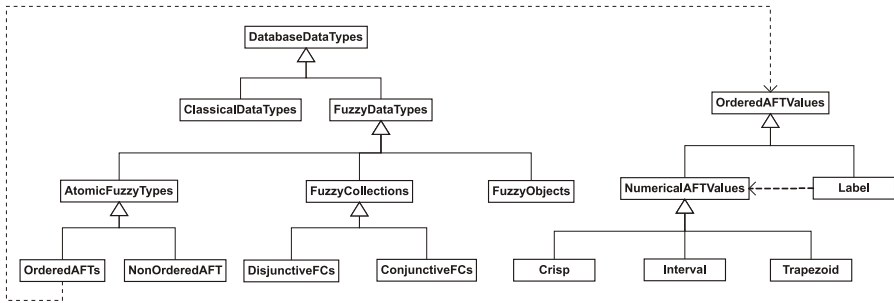


Fig. 2. Datatype hierarchy for the FORDBMS

Our FORDBMS is able to handle and represent a wide variety of fuzzy datatypes, which allow to easily model any sort of fuzzy data. These types of fuzzy data, that are shown in Fig. 2, are the following:

- Atomic fuzzy types (AFT), represented as possibility distributions over ordered (OAFT) or non ordered (NOAFT) domains.
- Fuzzy collections (FC), represented as fuzzy sets of objects, fuzzy or not, with conjunctive (CFC) or disjunctive (DFC) semantics.
- Fuzzy objects (FO), whose attribute types could be crisp or fuzzy, and where each attribute is associated with a degree to weigh its importance in object comparison.

All fuzzy types define a Fuzzy Equal operator (FEQ) that computes the degree of fuzzy equality for each pair of instances. Each fuzzy datatype has its own implementation of this operator in accordance with its nature. Moreover, the FORDBMS provides parameters to adjust the fuzzy equality computation to the semantics of the data handled. For OAFT the system uses the possibility measure to implement FEQ and implements other fuzzy comparators such as FGT (Fuzzy Greater Than), FGEQ (Fuzzy Greater or Equal), FLT (Fuzzy Less

Than) and FLEQ (Fuzzy Less or Equal), using this measure. Also, OAFI implement another version of those operators by using the necessity measure (NFEQ, NFGT, NFGEQ, NFLT and NFLEQ). A detailed description of these operators functioning can be found in [18].

4 Retrieving Medical Images from Databases Using the Extracted Curve Parameters

We have chosen an example based on the representation of the characteristics of curved spines taken from AP X-rays to illustrate how flexible queries can retrieve interesting information for the radiologists. Next, we show how to model the data structure needed in this example on our FORDBMS, how to parameterize the comparison behavior and how flexible queries are solved by the FORDBMS. We will analyze the retrieved images to show that our FORDBMS provides semantically rich results that visually match the curvature pattern sought.

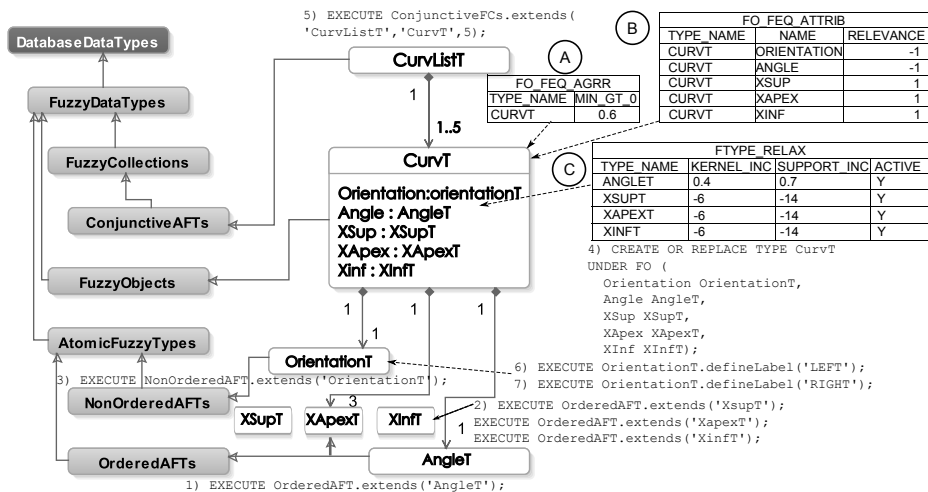


Fig. 3. DDL statements to represent the spine curves and catalog tables to parameterize comparison behavior

Classes in Fig. 3 with white background integrate the database datatype structure to represent spine parameters. Besides, this figure shows the statements to create these datatypes and the catalog tables that store the parameters that define the comparators behavior in query processes. As this figure illustrates, we model the curve description of the spine as a fuzzy conjunctive collection subtype called *CurvList*, which is created with statement 5). This subtype can include up to five fuzzy object subtype instances (instances of the *CurvT* subtype, created by the statement 4)). *CurvT* datatype represents a spine curve description. This

datatype has five attributes: *Orientation*, *Angle*, *XSup*, *XApex* and *XInf* that store, for each curve, the orientation of the curve (right or left, labels created by sentences 6) and 7), respectively), the angle value and the coordinates of the superior, apical and inferior points on the x-axis. The attribute *Orientation* stores instances of the *OrientationT* subtype of NOAFT, (this subtype is created by means of statement 3)). This attribute allows to perform queries with an unknown value for the orientation of the curve. The attribute *Angle* is of type *AngleT*, a subtype of OAFT, which is created by sentence 1). Thus, we can store on this attribute numerical values, crisp or fuzzy (using a trapezoidal membership function). The attributes *XSup*, *XApex* and *XInf* store instances of the *XSupT*, *XApexT* and *XInfT* OAFT subtypes, which are created by sentence 2). When complex fuzzy datatypes representing real objects are compared, we need to capture the real semantics of these objects. In our case, this implies relaxing some comparisons because of the imprecise nature of the spine segmentation. If we want to retrieve images containing a curve of a given angle, e.g. 30 degrees, the radiologist may also be interested in retrieving images with curves whose angles are between 25 and 35 degrees. The same applies to the limit of the considered curve. Therefore, in a whole curve comparison it is necessary to relax angle values and curve limits. With the help of Fig. 3 we will describe the necessary definitions to obtain this behavior.

The types *XSupT* and *XInfT* represent the x-coordinates for the superior and inferior extremes of the curve, and the *XApexT* represents the x-coordinate of the apex of the curve. All these coordinates are normalized between 0 and 100, where 0 is the position of the superior extreme of the spine and 100 the position of its inferior extreme.

To relax angle values in flexible comparisons we use the next three statements: `XSupT.setRelax(-6, -14, 'Y')`; `XInfT.setRelax(-6, -14, 'Y')`; and `XApexT.setRelax(-6, -14, 'Y')`; . We increment the kernel with 6 units and the support with 14 units for each instance of these subtypes in FEQ comparisons. The sentence `AngleT.setRelax(0.4, 0.7, 'Y')`; relax the kernel by 40% and the support by 70% for each angle value in FEQ comparisons. The relaxation degrees are specified using a percentage for angle values as the flexibility of the comparison should be proportional (the higher the angle, the wider the interval). However, the flexibility of the comparison should be constant for all vertebrae comparison, independently of their position, thus an integer value is used. To distinguish cases, positive or negative values are used respectively. The catalog table labeled as C) in Fig. 3 shows these parameters values.

The values shown in catalog table labeled as A) in Fig. 3 indicate to the FORDBMS that, for a FEQ comparison on two instances of *CurveT*, it must return 0 when at least 60% of attribute comparisons (3 in this case) return 0. This is to discard too different curves as the comparison is performed.

The values set in the catalog table labeled B) in Fig. 3 indicate that, if a comparison on the attribute *Angle* or the attribute *Orientation* returns 0, then the whole curve comparison must return 0 (the value -1 establishes that they are discriminant). This is to discard curves left oriented when we are searching for

curves right oriented and vice versa, as well as curves with a very different angle value. Finally, the catalog table *FO_FEQ_ATTRIB* shows that all attributes are equally relevant in object comparisons, since the weight of the aggregation is the absolute value of attribute *relevance*.

Once we have defined the data structure for the spine description and the behavior for the comparisons, we can create a table that stores X-rays images along with their fuzzy descriptions as follows:

```
create table APXRay (image# number, xray bfile, SpineCurv CurvListT);
```

We have inserted fifty-nine images, with their curve parameters automatically extracted as showed in Section 2, for query evaluation. For example, the following SQL sentence inserts the image shown on the left in Fig. 4, along with its curve parameters:

```
Insert into apxray values (56,BFILENAME('APXRays','56.gif'),
  CurvListT( 1,CurvT(OrientationT('RIGHT'), AngleT(Crisp(19.5)),
    XSupT(Crisp(26.9)), XApexT(Crisp(33.1)), XinfT(Crisp(44.3))),
    1,CurvT(OrientationT('LEFT'), AngleT(Crisp(39.5)),
    XSupT(Crisp(44.3)), XApexT(Crisp(52.3)), XinfT(Crisp(63.7))),
    1,CurvT(OrientationT('RIGHT'), AngleT(Crisp(38)),
    XSupT(Crisp(63.7)), XApexT(Crisp(78.6)), XinfT(Crisp(87.6))));
```

4.1 Computation of the Flexible Query

The database structure for image description created on the FORDBMS allows to perform two kinds of flexible queries: searching images that include a certain curve or set of curves, and searching images whose curvature pattern is similar to the one present in a sample image. The computation of the first kind of queries involves the use of the following comparators: *FEQ* on instances of the subtype *CurvT* of *FO*, and *FInclusion* on instances of the subtype *CurvListT* of *CFC*. The computation of the second kind of queries also involves the use of the comparator *FEQ* on instances of the subtype *CurvListT* of *CFC*. With the help of Fig. 4, we are going to show how these comparisons are computed, taking into account the flexibility parameters set into the catalog:

- *FEQ* computation on instances of *CurvT*. To compute the fuzzy equality of two curves, the FORDBMS evaluates the fuzzy equality between the values of each attribute and then, computes a weighted average from the five degrees obtained. It is previously checked weather all the discriminant attributes yield comparison values greater than 0 and if enough attributes do so. If that is not the case, no additional computation is needed. For example, for the images shown in Fig. 4, the computations to evaluate *FEQ*(s1,c1) are:

$$\begin{aligned}
 FEQ(s1, c1) &= (FEQ('RIGHT', 'RIGHT') \cdot |-1| + FEQ(19.5, 25.5) \cdot |-1| \\
 &\quad + FEQ(26.9, 18.3) \cdot |1| + FEQ(33.1, 26.1) \cdot |1| \\
 &\quad + FEQ(44.3, 41.7) \cdot |1|) / (1 + 1 + 1 + 1 + 1) \\
 &= (1 + 1 + 0.68 + 0.88 + 1) / 5 = 0.91 .
 \end{aligned}$$

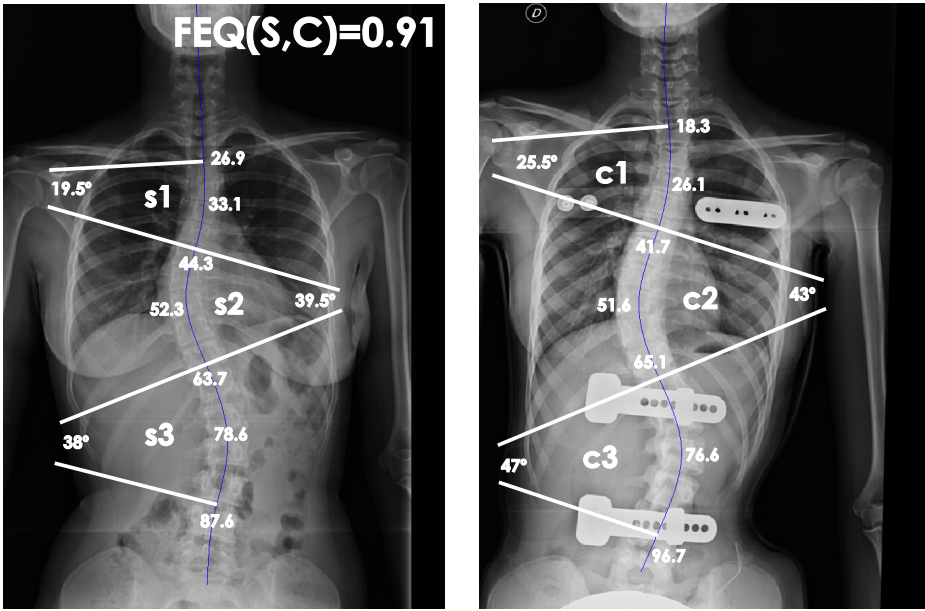


Fig. 4. Curve parameters used to compute fuzzy equality between the spine geometry of the images

In the previous comparison we have used the parameters set in the catalog table labeled as C) in Fig. 3 to relax the comparison of *Angle*, *XSup*, *XApex* and *Xinf* values. The values set into tables labeled as A) and B) are used to compute the whole comparison.

- *FInclusion* computation on instances of *CurvListT*. To evaluate the fuzzy inclusion of the set of curves of the left image into the set of curves of the right image, $FInclusion(LeftImage, RightImage)$, the FORDBMS computes the fuzzy equality for the curve *s1* on *LeftImage* with respect to each curve in *RightImage*, (*c1*, *c2*, *c3*), and takes the maximum; then, it evaluates the same for the curve *s2* and for the curve *s3*; finally, it takes the minimum of these three values as the *FInclusion* degree. In the example shown in Fig. 5 the computations are:

$$\begin{aligned}
 FInclusion(LeftImage, RightImage) &= \min(\max(FEQ(s1, c1), \\
 &FEQ(s1, c2), FEQ(s1, c3)), \max(FEQ(s2, c1), FEQ(s2, c2), \\
 &FEQ(s2, c3)), \max(FEQ(s3, c1), FEQ(s3, c2), FEQ(s3, c3))) \\
 &= \min(\max(0, 0.91, 0, 0), \max(0, 1, 0), \max(0, 0, 0.924)) = 0.91 .
 \end{aligned}$$

- *FEQ* computation on instances of *CurvListT*. To compute this comparison the system performs these two additional comparisons: $FInclusion(LeftImage, RightImage)$ and $FInclusion(RightImage, LeftImage)$, and returns the minimum. In this example:

$$\begin{aligned}
 &FEQ(LeftImage, RightImage) \\
 &= \min(FInclusion(LeftImage, RightImage), \\
 &FInclusion(RightImage, LeftImage)) = \min(0.91, 0.91) = 0.91 .
 \end{aligned}$$

4.2 Examples of Queries

As stated in Section 2, doctors are interested, for treatment purposes, in finding patients with similar spine curvature patterns. Fig. 5 shows several queries that search for X-rays of patients that present similar curvature patterns to that of the the query image labeled as Q_i .

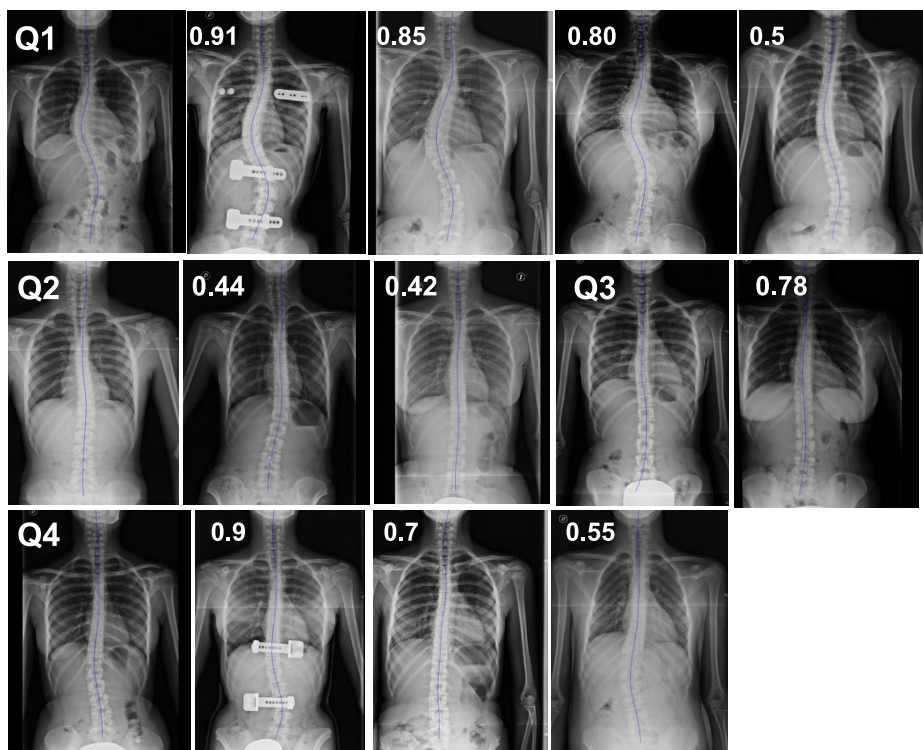


Fig. 5. Images retrieved (right) for each reference image (Q_i)

As can be seen, a higher compliance degree for an image denotes better visual matching with the query image and conversely: the lower the similarity between the curvature patterns is, the lower the compliance degree computed for them is.

5 Concluding Remarks

We have presented an application of fuzzy databases to medical images retrieval, which is an operation commonly needed by doctors when diagnosing patients. Since they are interested in retrieving similar images more than exactly equal images, we have devised a mechanism to allow flexible querying of such images. The images are processed using automatic segmentation techniques that give us information about the spine curvature pattern of each patient. This information is then stored in a FORDBMS so that flexible queries can be performed by means of the parameterizable operators provided by the system.

Although the focus of this paper has been in X-rays of patients suffering scoliosis, another type of images and pathologies can be easily considered. The information that describes the images could be provided by automatic algorithms, as in this case, or could be provided by medical experts as well. The capabilities for fuzzy data representation and fuzzy comparators implemented in the FORDBMS can then be used to design a scheme for the retrieval process.

Acknowledgment. This work has been supported by the “Consejería de Innovación Ciencia y Empresa de Andalucía” (Spain) under research projects P06-TIC-01570 and P07-TIC-02611, and the Spanish Ministry of Science and Innovation (MICINN) under grants TIN2009-08296 and TIN2007-68084-CO2-01.

References

1. Barranco, C.D., Campaña, J.R., Medina, J.M.: Towards a fuzzy object-relational database model. In: Galindo, J. (ed.) *Handbook of Research on Fuzzy Information Processing in Databases*, pp. 435–461. IGI Global (2008)
2. Caluwe, R.D.: *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*, *Advances in Fuzzy Systems-Applications and Theory*, vol. 13. World Scientific, Singapore (1997)
3. Cobb, J.: Outline for the study of scoliosis. *Am. Acad. Orthop. Surg. Inst. Course. Lect.* 5, 261–275 (1948)
4. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models: their training and application. *Comput. Vis. Image Underst.* 61, 38–59 (1995)
5. Cubero, J.C., Marín, N., Medina, J.M., Pons, O., Vila, M.A.: Fuzzy object management in an object-relational framework. In: *Proc. 10th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2004*, pp. 1767–1774 (2004)
6. Galindo, J. (ed.): *Handbook of Research on Fuzzy Information Processing in Databases*. Information Science Reference, Hershey (2008)
7. Garrido, A., Martínez-Baena, J., Medina, J.M., Jaime-Castillo, S.: A hierarchical deformable model based on dynamic programming to segment the spine (2010) (manuscript submitted)
8. Gokcen, I., Yazıcı, A., Buckles, B.P.: Fuzzy content-based retrieval in image databases. In: Yakhno, T. (ed.) *ADVIS 2000. LNCS*, vol. 1909, pp. 226–237. Springer, Heidelberg (2000)
9. Han, J., Ma, K.K.: Fuzzy color histogram and its use in color image retrieval. *IEEE Transactions on Image Processing* 11(8), 944–952 (2002)

10. Kanglin, X., Xiaoling, W.: Application of fuzzy logic in content-based image retrieval. *Journal of Computer Science & Technology* 5(1), 19–24 (2005)
11. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
12. Krishnapuram, R., Medasani, S., Jung, S.H., Choi, Y.S., Balasubramaniam, R.: Content-based image retrieval based on a fuzzy approach. *IEEE Trans. on Knowl. and Data Eng.* 16, 1185–1199 (2004)
13. Kulkarni, S.: Natural language based fuzzy queries and fuzzy mapping of feature database for image retrieval. *Journal of Information Technology and Applications* 4, 11–20 (2010)
14. Lew, M.S., Sebe, N., Lifi, C.D., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications* 2(1), 1–19 (2006)
15. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40(1), 262–282 (2007)
16. Ma, Z., Li, Y.: A literature overview of fuzzy database models. *Journal of Information Science and Engineering* 24, 189–202 (2008)
17. Medina, J.M., Pons, O., Vila, M.A.: Gefred. a generalized model of fuzzy relational databases. *Information Sciences* 76(1), 87–109 (1994)
18. Medina, J.M., Barranco, C.D., Campaña, J.R., Jaime-Castillo, S.: Generalized fuzzy comparators for complex data in a fuzzy object-relational database management system. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) *IPMU 2010. CCIS*, vol. 81, pp. 126–136. Springer, Heidelberg (2010)
19. Muller, H., Michoux, N., Bandon, D., Geissbuler, A.: A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. *International Journal of Medical Informatics* 73(1), 1–23 (2004)
20. Nepal, S., Ramakrishna, M., Thom, J.: A fuzzy object query language (foql) for image databases. In: *Proceedings of the 6th International Conference on Database Systems for Advanced Applications*, pp. 117–124 (1999)
21. Reinsch, C.: Smoothing by spline functions. *Numerische Mathematik* 10(3), 177–183 (1967)
22. Sánchez, D., Chamorro-Martínez, J., Vila, M.A.: Modelling subjectivity in visual perception of orientation for image retrieval. *Information Processing and Management* 39, 251–266 (2003)
23. Shahabi, C., Chen, Y.S.: Soft query in image retrieval systems. In: *Proceeding of SPIE Internet Imaging, Electronic Imaging*, San Jose, CA (USA), vol. 3964, pp. 57–68 (2000)
24. Verma, B., Kulkarni, S.: Fuzzy logic based interpretation and fusion of color queries. *Fuzzy Sets and Systems* 147, 99–118 (2004)
25. Wu, J.K., Narasimhalu, A.D.: Fuzzy content-based retrieval in image databases. *Information Processing and Management* 34(5), 513–534 (1998)

A Fuzzy-Rule-Based Approach to the Handling of Inferred Fuzzy Predicates in Database Queries

Allel Hadjali and Olivier Pivert

Irisa – Enssat, University of Rennes 1
Technopole Anticipa 22305 Lannion Cedex France
{hadjali,pivert}@enssat.fr

Abstract. This paper deals with database preference queries involving fuzzy conditions which do not explicitly refer to an attribute from the database, but whose meaning is rather inferred from a set of fuzzy rules. The approach we propose, which is based on the fuzzy inference pattern called generalized modus ponens, significantly increases the expressivity of fuzzy query languages inasmuch as it allows for new types of predicates. An implementation strategy involving a coupling between a DBMS and an inference engine is outlined.

1 Introduction

In database research, the last decade has witnessed a growing interest in preference queries. Motivations for introducing preferences inside database queries are manifold [1]. First, it has appeared to be desirable to offer more expressive query languages that can be more faithful to what a user intends to say. Second, the introduction of preferences in queries provides a basis for rank-ordering the retrieved items, which is especially valuable in case of large sets of items satisfying a query. Third, a classical query may also have an empty set of answers, while a relaxed (and thus less restrictive) version of the query might be matched by items in the database.

Approaches to database preference queries may be classified into two categories according to their qualitative or quantitative nature [1]. In the latter, preferences are expressed quantitatively by a monotone scoring function, and the overall score is positively correlated with partial scores. Since the scoring function associates each tuple with a numerical score, tuple t_1 is preferred to tuple t_2 if the score of t_1 is higher than the score of t_2 . Representatives of this family of approaches are top- k queries [2] and fuzzy-set-based approaches (e.g., [3]). In the qualitative approach, preferences are defined through binary preference relations. Representatives of qualitative approaches are those relying on a dominance relationship, e.g. Pareto order, in particular *Preference SQL* [4] and Skyline queries [5] and the approach presented in [6].

In this paper, we focus on the fuzzy-set-based approach to preference queries, which is founded on the use of fuzzy set membership functions that describe the preference profiles of the user on each attribute domain involved in the query. The framework considered is that of the fuzzy query language called SQLf [3].

The objective is to extend SQLf so as to authorize the use, inside queries, of fuzzy predicates for which *there does not exist any underlying attribute* in the database [7]. In such a case, it is of course impossible to explicitly define the membership function attached to a fuzzy predicate P related to a missing attribute from the database. It is rather assumed that the satisfaction level of P depends on the satisfaction level of other predicates C_1, \dots, C_n , but it is in general not easy to express its membership function μ_P as a simple aggregation of the μ_{C_i} 's. A solution is to use fuzzy rules to compute the satisfaction degrees associated with the retrieved items w.r.t the fuzzy predicate P (somewhat in the spirit of [8] where fuzzy preferences are inferred from a fuzzy characterization of the user's context).

The remainder of the paper is structured as follows. Section 2 consists of a short reminder about fuzzy sets and fuzzy queries. Section 3 provides a critical review of the work proposed in [7] for modelling inferred fuzzy predicates. In Section 4, we present an alternative approach, based on fuzzy rules and the inference pattern known as generalized modus ponens. Implementation aspects are dealt with in Section 5, whereas Section 6 summarizes the contributions and outlines some perspectives for future work.

2 Reminder about Fuzzy Sets and Fuzzy Queries

2.1 Basic Notions about Fuzzy Sets

Fuzzy set theory was introduced by Zadeh [9] for modeling classes or sets whose boundaries are not clear-cut. For such objects, the transition between full membership and full mismatch is gradual rather than crisp. Typical examples of such fuzzy classes are those described using adjectives of the natural language, such as *young*, *cheap*, *fast*, etc. Formally, a fuzzy set F on a referential U is characterized by a membership function $\mu_F : U \rightarrow [0, 1]$ where $\mu_F(u)$ denotes the grade of membership of u in F . In particular, $\mu_F(u) = 1$ reflects full membership of u in F , while $\mu_F(u) = 0$ expresses absolute non-membership. When $0 < \mu_F(u) < 1$, one speaks of partial membership.

Two crisp sets are of particular interest when defining a fuzzy set F :

- the core $C(F) = \{u \in U \mid \mu_F(u) = 1\}$, which gathers the *prototypes* of F ,
- the support $S(F) = \{u \in U \mid \mu_F(u) > 0\}$.

In practice, the membership function associated with F is often of a trapezoidal shape. Then, F is expressed by the quadruplet (a, b, c, d) where $C(F) = [b, c]$ and $S(F) = [a, d]$, see Figure 1.

Let F and G be two fuzzy sets on the universe U , we say that $F \subseteq G$ iff $\mu_F(u) \leq \mu_G(u)$, $\forall u \in U$. The complement of F , denoted by F^c , is defined by $\mu_{F^c}(u) = 1 - \mu_F(u)$. Furthermore, $F \cap G$ (resp. $F \cup G$) is defined the following way: $\mu_{F \cap G} = \min(\mu_F(u), \mu_G(u))$ (resp. $\mu_{F \cup G} = \max(\mu_F(u), \mu_G(u))$).

As usual, the logical counterparts of the theoretical set operators \cap , \cup and complementation operator correspond respectively to the conjunction \wedge , disjunction \vee and negation \neg . See [10] for more details.

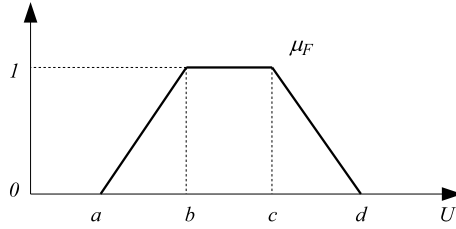


Fig. 1. Trapezoidal membership function (t.m.f.)

2.2 About SQLf

The language called SQLf described in [3] extends SQL so as to support fuzzy queries. The general principle consists in introducing gradual predicates wherever it makes sense. The three clauses *select*, *from* and *where* of the base block of SQL are kept in SQLf and the “from” clause remains unchanged. The principal differences affect mainly two aspects :

- the calibration of the result since it is made with discriminated elements, which can be achieved through a number of desired answers (k), a minimal level of satisfaction (α), or both, and
- the nature of the authorized conditions as mentioned previously.

Therefore, the base block is expressed as:

select [**distinct**] [$k \mid \alpha \mid k, \alpha$] attributes **from** relations **where** fuzzy-cond

where “fuzzy-cond” may involve both Boolean and fuzzy predicates. This expression is interpreted as:

- the fuzzy selection of the Cartesian product of the relations appearing in the *from* clause,
- a projection over the attributes of the *select* clause (duplicates are kept by default, and if *distinct* is specified the maximal degree is attached to the representative in the result),
- the calibration of the result (top k elements and/or those whose score is over the threshold α).

The operations from the relational algebra — on which SQLf is based — are extended to fuzzy relations by considering fuzzy relations as fuzzy sets on the one hand and by introducing gradual predicates in the appropriate operations (selections and joins especially) on the other hand. The definitions of these extended relational operators can be found in [11]. As an illustration, we give the definitions of the fuzzy selection and join operators hereafter, where r and s denote two fuzzy relations defined respectively on the sets of domains X and Y .

- $\mu_{select(r, cond)}(t) = \top(\mu_r(t), \mu_{cond}(t))$ where $cond$ is a fuzzy predicate and \top is a triangular norm (most usually, min is used),

- $\mu_{join(r, s, A, B, \theta)}(tu) = \top(\mu_r(t), \mu_s(u), \mu_\theta(t.A, u.B))$ where A (resp. B) is a subset of X (resp. Y), A and B are defined over the same domains, θ is a binary relational operator (possibly fuzzy), $t.A$ (resp. $u.B$) stands for the value of t over A (resp. u over B).

3 Koyuncu’s Approach

The situation considered is that when a user wants to express a fuzzy selection condition in his/her query but i) there does not exist any associated attribute in the database whose domain could be used as the referential underlying a fuzzy membership function, ii) it is not possible to express in a simple way the fuzzy condition as an aggregation of elementary fuzzy requirements on different attributes. We first present the approach presented in [7] and point out some of its shortcomings.

3.1 Principle of the Approach

An example given in [7] considers a relation *Match* describing soccer matches, with schema (*#id, goalPositions, goals, fouls, penalties, disqualifications, year*), and queries such as: “find the matches played in 2010 with a high harshness level”. In this query the fuzzy condition “high harshness level” does not refer to any specific attribute from the relation. The author proposes to give it a semantics by means of rules such as:

if (*fouls is several*) **or** (*fouls is many*) (with threshold 0.6)
and (*penalties is average*) (with threshold 0.7)
and (*disqualifications is average*) (with threshold 0.5)
then *harshness is high* (with $\mu = Y$).

In this rule (let us denote it by R_1), Y denotes the membership degree attached to the conclusion, and it is computed using i) the degrees attached to the predicates in the left-hand side of the rule, ii) the so-called “matching degree of the rule conclusion”, and iii) an implication function (Gödel’s implication defined as

$$p \rightarrow_{G\ddot{o}} q = \begin{cases} 1 & \text{if } q \geq p \\ q & \text{otherwise.} \end{cases}$$

is used by the author). Conjunction and disjunction in the left-hand side are interpreted by *min* and *max* respectively.

The “matching degree of the rule conclusion” expresses the extent to which the fuzzy term present in the right-hand side of the rule (*high* in the rule above) corresponds to the fuzzy term involved in the user query (for instance, the user might aim to retrieve matches with a *medium* level of harshness, in which case, the matching degree would assess the similarity between *high* and *medium*).

Example 1. Let us consider the query:

select #id from Match where year = 2010 and harshness_level is medium

and the following tuple from relation *Match*:

$$\langle 1, 19, 8, 23, 5, 3, 2010 \rangle.$$

Let us assume that

$$\begin{aligned} \mu_{several}(23) &= 0, & \mu_{many}(23) &= 1, & \mu_{average}(5) &= 1, \\ \mu_{average}(3) &= 0.5, & \mu_{sim}(high, medium) &= 0.5. \end{aligned}$$

Since $\max(0, 1) \geq 0.6$, $1 \geq 0.7$, and $0.5 \geq 0.5$, rule R_1 can be fired. The final truth degree obtained for the predicate “*harshness is medium*” is equal to:

$$\min(\max(0, 1), 1, 0.5) \rightarrow_{G\ddot{o}} \mu_{sim}(high, medium) = 0.5 \rightarrow_{G\ddot{o}} 0.5 = 1. \diamond$$

3.2 Critical Discussion

The technique advocated by Koyuncu calls for several comments:

- First and foremost, this approach does not actually infer fuzzy predicates. The truth degree it produces is associated with a gradual *rule* (it is an *implication degree*), not with the fuzzy *term* present in the conclusion of the rule. This way of doing does not correspond to a well-founded inference scheme such as, for instance, the generalized modus ponens [12].
- The use of a similarity relation between linguistic labels for assessing the rule is debatable: on which semantic basis are the similarity degrees defined and by whom?
- The use of *local thresholds* in the left-hand sides of the rules is somewhat contradictory with the “fuzzy set philosophy” which is rather oriented toward expressing trade-offs between different gradual conditions.
- It is not clear what happens when several rules involving the same attribute in their conclusions can be fired at the same time.

4 A Fuzzy-Inference-Based Approach

In [13], we proposed an approach based on a simple inference method borrowed from the fuzzy control field. The main limitation of this method lies in the fact that it considers a particular interpretation of a fuzzy rule: that based on a conjunction-like operator (for instance, the minimum) instead of a logical implication operator. This is not suitable when chaining rules in order to deduce new pieces of information. Here, we propose an alternative approach aimed at defining the semantics of an inferred fuzzy predicate by means of a set of fuzzy rules, and the inference principle known as *generalized modus ponens*. Consider again the relation *Match*; an example of the type of queries considered is “find the matches played in 2010 which were *highly harsh* and had *many* goals scored”, where *highly harsh* denotes an inferred fuzzy predicate.

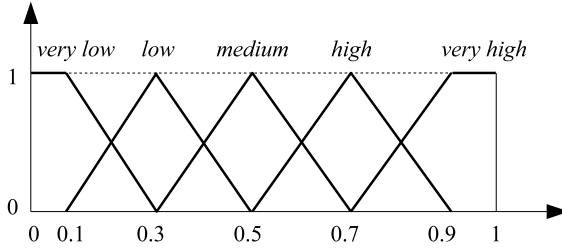


Fig. 2. Fuzzy partition

4.1 Overview of the Approach

Let \mathcal{D} be a relational database and Q a fuzzy query that involves a fuzzy predicate \mathcal{P} that is not related to any attribute specified in \mathcal{D} . Let \mathcal{A} be the “virtual attribute” concerned by the fuzzy predicate \mathcal{P} (for instance *harshness_level*). It is assumed that the domain of $t.\mathcal{A}$ is the unit interval. The principle of the approach we propose is summarized hereafter:

- One uses a fuzzy partition over the unit interval, associated with a list of linguistic labels (*very low*, *low*, *medium*, and so on), see an example of such a partition in Figure 2. These labels will be used in the conclusion parts of the fuzzy rules and constitute the basis of the evaluation of the satisfaction degree (in $[0, 1]$) of the inferred fuzzy predicates present in the query.
- One considers a fuzzy rule base \mathcal{B} defined by an expert of the domain. Rules are of the form:

R_1 : **if** (*fouls is several*) **or** (*fouls is many*)
and (*penalties is average*) **and** (*disqualifications is average*)
then *harshness_level is high*.

R_2 : **if** (*fouls is low*) **or** (*fouls is very low*)
and (*penalties is very low*) **and** (*disqualifications is very low*)
then *harshness_level is very low*.

- One considers a subset \mathcal{S} of tuples of \mathcal{D} . For instance, (i) in the case where Q is of a conjunctive type, \mathcal{S} is the set of answers to the query Q^* obtained from Q by deleting the fuzzy predicate \mathcal{P} ; (ii) in the case where Q is of a disjunctive type, \mathcal{S} is \mathcal{D} .
- For each tuple $t \in \mathcal{S}$, one computes its satisfaction degree w.r.t. the fuzzy requirement expressed by \mathcal{P} . This can be achieved by a two-step procedure:
 - First, one infers the fuzzy characterization of $t.\mathcal{A}$ from the rule base \mathcal{B} . To this end, an appropriate fuzzy inference pattern, which will be discussed in the next section, is used.
 - The degree of satisfaction $\mu_{\mathcal{P}}(t)$ is then equal to the *compatibility* between the fuzzy description $t.\mathcal{A}$ and the fuzzy predicate \mathcal{P} . One can use the following formula to compute the *compatibility degree* [14]:

$$\mu_{\mathcal{P}}(t) = \sup_{u \in [0, 1]} \min(\mu_{t.\mathcal{A}}(u), \mu_{\mathcal{P}}(u)). \tag{1}$$

4.2 Fuzzy Inference Scheme

As mentioned above, the goal is to infer a fuzzy predicate from the fuzzy rule base \mathcal{B} . To achieve this, we make use of the *generalized modus ponens (gmp)* as an inference pattern. In its simplest form, it reads:

from the rule: **if C is E then A is F**
 and the fact: C is E'

the conclusion A is F' can be inferred, where F' and E' , E and F are fuzzy sets. For $v \in \text{dom}(A)$, $\mu_{F'}(v)$ is computed by means of the combination/projection principle [14]:

$$\mu_{F'}(v) = \sup_{u \in \text{dom}(C)} \top(\mu_{E'}(u), \mu_E(u) \rightarrow \mu_F(v))$$

where \top stands for a triangular norm and \rightarrow a fuzzy implication. Assuming that the operator \rightarrow represents Gödel implication, i.e., $a \rightarrow b = 1$ if $a \leq b$ and b otherwise, and \top the *min* operator, we write $F' = [E' \circ (E \rightarrow F)]$ where \circ is the sup-min composition operator.

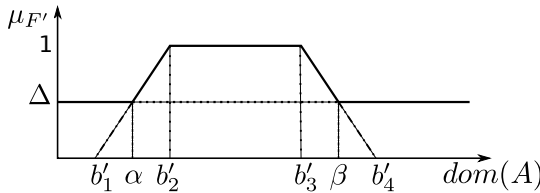


Fig. 3. t.m.f of the fuzzy set F'

In practice, if E , E' and F are represented by $(a_1, a_2, a_3, a_4, 0)$, $(a'_1, a'_2, a'_3, a'_4, 0)$ and $(b_1, b_2, b_3, b_4, 0)$ respectively, the t.m.f. $(b'_1, b'_2, b'_3, b'_4, \Delta)$ (where Δ expresses a global indetermination level which corresponds to the uncertainty degree attached to the conclusion when the inclusion relation $\text{support}(E') \subseteq \text{support}(E)$ does not hold) associated with F' is computed the following way (see [8] for more details):

$$\begin{aligned} \Delta &= \sup_{\{u \in \text{dom}(C) \mid \mu_E(u)=0\}} \mu_{E'}(u), \\ b'_1 &= b_1 \text{ and } b'_4 = b_4 \\ b'_2 &= b_2 - (1 - H)(b_2 - b_1), \\ b'_3 &= b_3 + (1 - H)(b_4 - b_3), \end{aligned}$$

with $H = \min(\mu_E(a'_2), \mu_E(a'_3))$. H is the smallest degree of an element belonging to the core of E' in E . As we can see in Figure 3, in the case where $\Delta > 0$, any value outside $[\alpha, \beta]$ is considered acceptable with a degree Δ . In particular, if

¹ The t.m.f. $(a, b, c, d, 0)$ is an augmented variant of (a, b, c, d) .

$\Delta = 1$ (i.e., $core(E') \not\subseteq support(E)$), $\mu_{F'}(v) = 1, \forall v \in dom(A)$. This means that we have a total indetermination about the inferred conclusion. As a consequence, the smaller Δ , the more certain the inferred conclusion is. For a precise input, i.e., $E' = \{e'\}$ which corresponds to the following t.m.f. $(e', e', e', e', 0)$, the t.m.f. of F' is such that: $\Delta = 1$ and $H = 0$ if $e' \notin E$, $\Delta = 0$ and $H = \mu_E(e')$ otherwise.

Obviously, one can choose other fuzzy implication operators. However, the major advantage of Gödel implication lies in the fact that it is the least sensitive to the mismatch between E and E' . Indeed, the global indetermination level is non-zero only in the case where the support of E' is not included in the support of E . Also, and due to the use of the *min* operator as a triangular norm here, Gödel implication is then more appropriate than other implications.

When the condition part of a fuzzy rule involves several elementary conditions, the *gmp* inference scheme writes:

Rule: **if** $(C_1 \text{ is } E_1) \wedge \dots \wedge (C_q \text{ is } E_q)$ **then** $A \text{ is } F$

Fact: $(C_1 \text{ is } E'_1) \wedge \dots \wedge (C_q \text{ is } E'_q)$,

where the t.m.f. associated with E_i (resp. E'_i) is $(a_{1i}, a_{2i}, a_{3i}, a_{4i})$ (resp. $(a'_{1i}, a'_{2i}, a'_{3i}, a'_{4i})$), for $i = 1, q$. In this case, the t.m.f. of the conclusion F' is computed in the same way as previously, except for Δ and H which are given by $\Delta = \max_{i=1,q} \Delta_i$ and $H = \min_{i=1,q} H_i$, with

$$\Delta_i = \sup\{\mu_{E_i}(u) \mid u \in dom(C_i), \mu_{E_i}(u)=0\}$$

and

$$H_i = \min(\mu_{E_i}(a'_{2i}), \mu_{E_i}(a'_{3i})).$$

4.3 Aggregating Step

It may be the case in practice that several fuzzy rules have their conclusion parts which pertain to a same attribute A . A same input can (approximately) match the condition parts of such rules. Let us consider two rules with a single premise to illustrate this case. We assume that each rule is coherent, as well as the the set of rules (see [14] for more details about this issue).

Case 1: R_1 : **if** $C_1 \text{ is } E_1$ **then** $A \text{ is } F_1$

R_2 : **if** $C_1 \text{ is } E_2$ **then** $A \text{ is } F_2$

For an input $\omega = C_1 \text{ is } E'$ and under the assumption $E' \cap E_i \neq \emptyset, i = 1, 2$, both R_1 and R_2 are triggered. To obtain a single overall characterization of the attribute A , two methods can be applied [14]. The first one, called FITA (First Infer Then Aggregate), consists in triggering the rules separately, then combining conjunctively the partial conclusions inferred. Let $(A \text{ is } F'_1)$ and $(A \text{ is } F'_2)$ be the conclusions deduced respectively from R_1 and R_2 . The overall conclusion on A is computed as follows (conjunctive aggregation is adopted due to the implication-based modeling of rules):

$$F' = \bigcap_{i=1,2} F'_i = \bigcap_{i=1,2} [E' \circ (E_i \rightarrow F_i)].$$

This method may result in a non-trapezoidal function, and then a trapezoidal approximation technique from the literature [8] must be used.

The second method, called FATI, first combines the rules, then infers. The semantics of F' is then computed as follows:

$$F' = E' \circ \left[\bigcap_{i=1,2} (E_i \rightarrow F_i) \right].$$

It has pointed out in [14] that:

$$E' \circ \left[\bigcap_{i=1,2} (E_i \rightarrow F_i) \right] \subseteq \bigcap_{i=1,2} [E' \circ (E_i \rightarrow F_i)].$$

This means that the FATI method leads to a conclusion which is more informative than the one obtained with FITA. However, building a t.m.f using FATI is not an easy task as explained in [8]. It is worth noticing that for a precise input, both methods yield the same result.

Case 2: R_1 : if C_1 is E_1 then A is F_1
 R_2 : if C_2 is E_2 then A is F_2

This second case can be seen as a variant of the first one where the premises of the fuzzy rules concern different attributes, but the conclusion is still over the same attribute. Thus, for an input $\{C_1 \text{ is } E'_1 \wedge C_2 \text{ is } E'_2\}$ such that $E'_1 \cap E_1 \neq \emptyset$ and $E'_2 \cap E_2 \neq \emptyset$, we get in the same situation as in the first case and, in a similar way, we aggregate the partial conclusions inferred.

4.4 Computation of the Final Degree

In the database query context considered in this paper, the ultimate goal is to compute satisfaction degrees of retrieved database tuples. This can be done as follows:

- one fires all of the rules which include the “virtual attribute” (for instance *harshness_Level*) in their conclusions,
- one aggregates the outputs of these rules (in the spirit of FITA method),
- one computes the degree of compatibility between the result and the fuzzy predicate from the query which concerns the virtual attribute.

Example 2. Let us consider again the query from Example 1:

select #id from Match where year = 2010 and harshness_level is medium

and the following tuple from relation *Match*: $\langle 1, 19, 8, 23, 5, 3, 2010 \rangle$. Let us assume the following trapezoidal membership functions:

$$several = (5, 10, 15, 20), \text{ many} = (20, 23, 28, 30), \text{ average} = (2, 5, 8, 11).$$

Using the rule R_1 from Subsection 4.1 (i.e., that from Example 1 without the thresholds), one can check that the parameters Δ_i and H_i corresponding to each condition C_i of the rule ($i = 1..3$) are:

$$\begin{aligned}\Delta_1 &= 0, & H_1 &= 1, \\ \Delta_2 &= 0, & H_2 &= 0.33, \\ \Delta_3 &= 0, & H_3 &= 1.\end{aligned}$$

Then, the global values of these two parameters are $\Delta = \max_i \Delta_i = 0$ and $H = \min_i H_i = 0.33$. Since $high = (0.5, 0.7, 0.7, 0.9)$, one can check that the fuzzy characterization of the inferred conclusion is $(0.5, 0.57, 0.83, 0.9)$. The satisfaction degree of the above tuple w.r.t fuzzy predicate *medium* whose t.m.f. is $(0.3, 0.5, 0.5, 0.7)$, is equal to the compatibility between the two fuzzy sets represented by $(0.5, 0.57, 0.83, 0.9)$ and $(0.3, 0.5, 0.5, 0.7)$ respectively. Using Equation (11), the degree obtained equals 0.71. \diamond

Remark. There may be some “dependencies” between the “regular” predicates of the selection condition and the inferred ones (if any). Some contradictions may even appear, in which case the query will result in an empty set of answers. An example is: “find the matches with a high level of harshness with a very low number of fouls”. In such a situation, it should be possible to devise a way to provide the user with some explanations about the failure of his/her query, which would imply searching for the conflicts between the definition of the inferred predicate(s) and the rest of the query.

5 Implementation Aspects

This approach implies coupling a DBMS with a fuzzy inference engine, according to the architecture sketched in Figure 4.

First, the SQLf query is compiled into a procedural program (called the “evaluator” hereafter) which scans the relation(s) involved. Let us assume that the query involves a global satisfaction threshold α (which can be user-defined). For each tuple t , the evaluator:

- computes the degrees related to the “regular” fuzzy predicates (i.e., the non-inferred ones) involved in the selection condition,
- sends a request (along with the tuple t itself) to the inference engine if necessary (i.e., in the presence of inferred fuzzy predicates in the selection condition). For each inferred predicate φ_i of the form “ att_i is F_i ” (where att_i denotes a “virtual attribute”) the inference engine
 - selects the rules which have att_i in their conclusion,
 - computes $\mu_{\varphi_i}(t)$ according to the process described in Subsection 4.4,
 - sends it back to the query evaluator, which
- computes the final degree attached to tuple t ,
- discards tuple t if its degree is smaller than α ,

In the case of a conjunctive selection condition involving both “regular” fuzzy predicates and inferred ones, it is possible to use the so-called *derivation method* proposed in [15] so as to avoid an exhaustive scan of the relation(s) concerned (a derived Boolean selection condition is then derived from the part of the initial

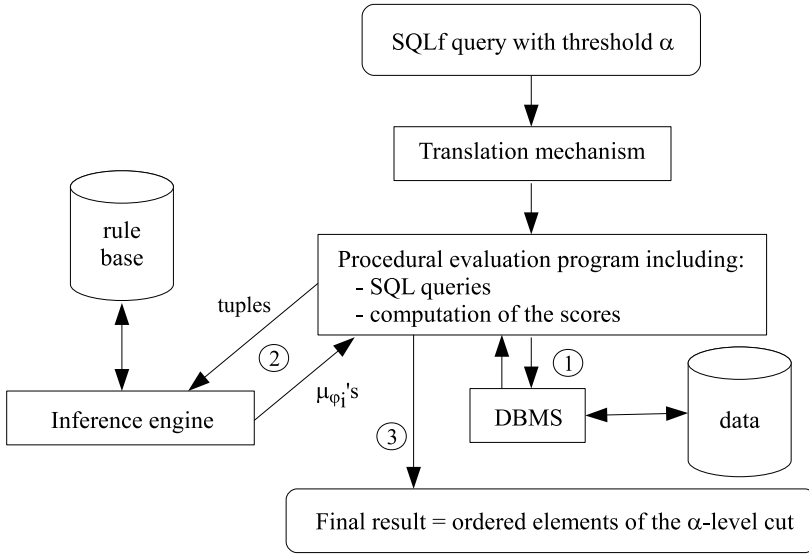


Fig. 4. Query processing strategy

fuzzy condition which does not include inferred predicates). Indeed, some connections between the fuzzy preference criteria considered and Boolean conditions make it possible to take advantage of the optimization mechanisms offered by classical DBMSs so as to efficiently process fuzzy queries.

6 Conclusion

In this paper, we have proposed an approach to the modelling and handling of *inferred fuzzy predicates*. These are predicates which may be used inside a database preference queries, which do not refer to any attribute from the database, and which cannot be easily defined in terms of a simple aggregation of other atomic predicates. After pointing out the flaws of a previous approach from the literature, we have defined an interpretation model based on a fuzzy rule base and the type of inference known as generalized modus ponens. The way such an inference module may be coupled with a DBMS capable of handling fuzzy queries has been described. An efficient processing technique based on the transformation of (non-inferred) fuzzy predicates into Boolean conditions makes it possible to expect a limited overhead in terms of performances.

As to perspectives for future work, one obviously concerns experimentation. The implementation of a prototype should make it possible to confirm the feasibility of the approach and the fact that reasonable performances may be expected from the evaluation strategy outlined in the present paper. The application of the approach to spatial and temporal databases constitutes also an interesting issue.

References

1. Hadjali, A., Kaci, S., Prade, H.: Database preferences queries – A possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 291–310. Springer, Heidelberg (2008)
2. Bruno, N., Chaudhuri, S., Gravano, L.: Top-k selection queries over relational databases: mapping strategies and performance evaluation. *ACM Trans. on Database Systems* 27, 153–187 (2002)
3. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. *IEEE Trans. on Fuzzy Systems* 3(1), 1–17 (1995)
4. Kießling, W., Köstler, G.: Preference SQL — Design, implementation, experiences. In: Proc. of VLDB 2002, pp. 990–1001 (2002)
5. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. of ICDE 2001, pp. 421–430 (2001)
6. Chomicki, J.: Preference formulas in relational queries. *ACM Transactions on Database Systems* 28(4), 427–466 (2003)
7. Koyuncu, M.: Fuzzy querying in intelligent information systems. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 536–547. Springer, Heidelberg (2009)
8. Hadjali, A., Mokhtari, A., Pivert, O.: A fuzzy-rule-based approach to contextual preference queries. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178, pp. 532–541. Springer, Heidelberg (2010)
9. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8(3), 338–353 (1965)
10. Dubois, D., Prade, H.: Fundamentals of fuzzy sets. *The Handbooks of Fuzzy Sets*, vol. 7. Kluwer Academic Publishers, Netherlands (2000)
11. Bosc, P., Buckles, B., Petry, F., Pivert, O.: Fuzzy databases. In: Bezdek, J., Dubois, D., Prade, H. (eds.) *Fuzzy Sets in Approximate Reasoning and Information Systems*. The Handbook of Fuzzy Sets Series, pp. 403–468. Kluwer Academic Publishers, Dordrecht (1999)
12. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning. *Fuzzy Sets and Systems* 40(1), 143–202 (1991)
13. Pivert, O., Hadjali, A., Smits, G.: On database queries involving inferred fuzzy predicates. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 592–601. Springer, Heidelberg (2011)
14. Bouchon-Meunier, B., Dubois, D., Godo, L., Prade, H.: Fuzzy sets and possibility theory in approximate and plausible reasoning. In: Dubois, D., Prade, H., Bezdek, J. (eds.) *Fuzzy Sets in Approximate Reasoning and Information Systems*, pp. 15–162. Kluwer Academic Publishers, The Netherlands (1999)
15. Bosc, P., Pivert, O.: SQLf query functionality on top of a regular relational database management system. In: Pons, O., Vila, M., Kacprzyk, J. (eds.) *Knowledge Management in Fuzzy Databases*, pp. 171–190. Physica-Verlag, Heidelberg (2000)

Flexible Content Extraction and Querying for Videos

Utku Demir¹, Murat Koyuncu², Adnan Yazici¹, Turgay Yilmaz¹, and Mustafa Sert³

¹ Department of Computer Engineering, Middle East Technical University
06531 Ankara, Turkey

² Department of Information Systems Engineering, Atilim University
06836 Ankara, Turkey

³ Department of Computer Engineering, Baskent University
Ankara, Turkey

mkoyuncu@atilim.edu.tr, {yazici,turgay}@ceng.metu.edu.tr,
msert@baskent.edu.tr

Abstract. In this study, a multimedia database system which includes a semantic content extractor, a high-dimensional index structure and an intelligent fuzzy object-oriented database component is proposed. The proposed system is realized by following a component-oriented approach. It supports different flexible query capabilities for the requirements of video users, which is the main focus of this paper. The query performance of the system (including automatic semantic content extraction) is tested and analyzed in terms of speed and accuracy.

Keywords: Multimedia database, fuzziness, content-based retrieval, multimedia querying.

1 Introduction

Large capacity and fast multimedia devices have become cheaper with the recent advances in technology and more and more people have begun to store and process large volumes of multimedia data by means of such devices. Conventional database approaches are not adequate for handling multimedia content in most cases due to the complex properties of multimedia objects such as multidimensional nature of data, non-standard and uncertain structures, and temporal and spatial aspects. Therefore, the need for specialized multimedia databases has been raised in order to access and retrieve desired portion of multimedia content easily. A multimedia database system should be able to extract existing semantic contents from the multimedia data, store and index semantic contents efficiently, and support some specialized capabilities for queries [1].

When multimedia data is in discussion, three different types of information gain importance: (1) *metadata* such as date, title and author, (2) knowledge that we infer as high level *semantic content*, (3) low level features such as fundamental frequency, harmonicity, dominant color, region shape and edge histogram as *audiovisual content* [2]. Although managing metadata and audiovisual content is easier, users are usually interested in meaning of the content (the high level *semantic content* in multimedia data), which is difficult to manage.

Objects, spatial/temporal relations and events are the basic components of such semantic content. The process of extracting these entities is a challenging research area and it is known as the *semantic gap* problem in multimedia domain. Using manual annotation techniques for extracting information from multimedia data is mostly boring, exhausting, slow and subjective to the user. Therefore, an automatic, preferably real-time annotation system is required. Since semantic content has no direct relation with raw multimedia data and is a collection of pixels and signals, it is very difficult to extract semantic content automatically from it [3]. To overcome this difficulty, some studies [4-9] offer different ways for extracting semantic information. However, more research is needed to achieve a mature technology in this domain.

Modeling of semantic entities is very important to be able to store multimedia data in an appropriate way to handle user queries efficiently [10-13]. Due to the complex and uncertain structure of multimedia data, using the object-oriented technology and handling uncertain information are considered as important requirements for developing effective multimedia databases. Long, nested and hierarchical transactions (resulting in slow queries) are inevitable with traditional approaches as multimedia objects contain huge amounts of information and exist in a compound form. Therefore, an efficient indexing mechanism is another important requirement for locating and retrieving multimedia objects accurately in an effective way.

To achieve a reasonable quality of service for multimedia database applications, a multimedia database management system, which considers all above functionalities, is necessary. Although there are various studies about multimedia databases, most of them ignore uncertainty and fuzziness existing in multimedia. Moreover, automatic object extraction, identification, and classification are usually neglected in such systems. In addition, most multimedia database related studies usually target a small research area in a limited domain.

The main motivation of this study is the need for an intelligent database system for accessing desired portions of video data efficiently and accurately. In this study, a fully qualified multimedia database system supporting annotation of objects, relations and events as well as providing all the basic functionalities of conventional database management systems such as indexing, querying and retrieving is presented. In this paper, especially, the flexible query capabilities of such a system are elaborated along with the performance results. Compared with the existing retrieval systems, the major contributions and advantages of the proposed system are as follows:

- In general, video related studies focus on a specific part of the content-based video information extraction and retrieval. However, this study gives a complete system including a semantic information extractor, a high-dimensional index structure and an intelligent database built around a coordinator structure.
- A novel automatic semantic content extraction model is developed and integrated into the system, which is probably the most challenging task for such systems.
- Uncertainty existing in video data or query is handled effectively, which is ignored by many other video database systems.
- The database is improved through a knowledge-based system to deduce domain information which is not stored in the database previously.
- Interoperability with the client and other systems is achieved through the web services, considerably a new technology for multimedia databases.

The rest of the paper is organized as follows; In Section 2, the proposed video database architecture is summarized by describing the main components and their functionalities. Section 3 summarizes the query types supported by the system. The query performance tests and their results are given in Section 4. The last chapter provides conclusions and gives future directions.

2 System Architecture

The overall view of the proposed system is given in Figure 1. A component-oriented approach is followed and a standard XML-based communication method is used for the interaction between the components of the architecture as well as between the clients and the server of the system. In the system architecture, a semantic information extractor, a high-dimensional index structure and an intelligent fuzzy object-oriented database components are coupled through a coordinator module. Information is provided by the semantic content extractor, then, stored in the intelligent fuzzy database component with uncertain properties and finally retrieved efficiently using the index structure specialized for multimedia data.

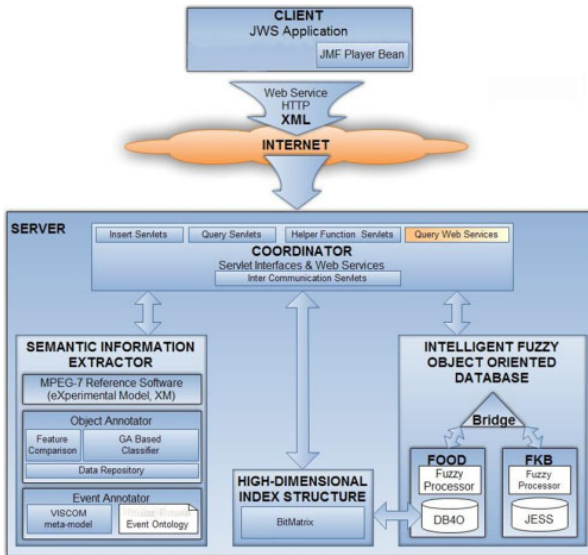


Fig. 1. System Architecture

The aim of the semantic content extractor module is to take a video as input and extract semantic content of the video in an appropriate format to be inserted into the intelligent fuzzy object-oriented database component. It is possible to use three alternatives for semantic content extraction: automatic extraction, manual extraction and hybrid extraction. In the automatic extraction, the contents of video (objects, relations and events) are extracted automatically without any human intervention and the extracted data is inserted into the database. In the manual extraction, the contents

of video are determined by a user and inserted into the database. Finally, in the hybrid approach, the automatic and manual extractions are used together as complementary to each other to benefit the advantages of two methods. Typically, automatic extraction is done first, and then the results are checked and corrected by a user. In addition, some object attributes which are not appropriate for automatic extraction are extracted by users manually.

An object-oriented database system is the best alternative to traditional relational approach for managing multimedia data, since multimedia data contains a great amount of complex information that exists in a compound and hybrid form. DB4Object is selected as the object database system in this study [14]. Contemporary systems require an interoperable use of database and knowledge base systems to add a level of intelligence to database systems [15]. The inference mechanism is utilized to deduce new semantic concepts about video content. JESS has been used as the knowledge-base of the system [16]. Fuzzy data processing is achieved by adding fuzzy processors to both the database and knowledge-base components. By integrating a fuzzy object-oriented database (FOOD) and a fuzzy knowledge-base (FKB), an appropriate database system is obtained for multimedia applications. A bridge is utilized for managing communication and interaction between FOOD and FKB. In addition to providing interoperability, it is used as an abstraction layer and entry point for user requests.

Since high-dimensionality is in consideration for multimedia data, most index structures become extremely inefficient because the number of nodes increases exponentially [17]. Moreover, most high-dimensional index algorithms face the problem of indexing large volumes of empty space. Therefore, storing and retrieving multimedia data efficiently require an index structure capable of handling such huge dimensions of multimedia objects as well as their low-level features. The intelligent fuzzy object-oriented database component of the system is supported by an access structure to increase efficiency of video retrieval as seen in Figure 1. BitMatrix access structure [18] is adapted to the video database in order to index the semantic objects of videos by using their low level features. We use four MPEG-7 descriptors which are color layout, dominant color, edge histogram, and region shape as low level features. These features are extracted using MPEG-7 eXperimentation Model (XM) Software [19] and then they are mapped to a bit string to be used in the BitMatrix access structure.

The coordinator is responsible for the interoperability between the main components to achieve multimedia data management. Besides, since the interaction with the client side is established via the coordinator, it can be considered as the interface of the system to the outer environment.

For the client side of the architecture, a light-weight, thin client application structure is suggested and developed using Java technologies. It is able to process audio-visual materials as well as textual results, since results of queries may contain audio and visual objects. Due to difficulties in constructing multimedia data queries, the client has user-friendly visual interfaces for building queries more easily. Interested readers may refer to [20-21] to get more information about the architecture and the functionality of components.

3 Query Capabilities

Semantic contents of videos are extracted and stored to satisfy different query requirements of users. This is the ultimate goal of a content-based video retrieval system. Therefore, it is very crucial to develop flexible query mechanisms. In this study, a client application which includes interfaces for query formulation is developed to provide a single access point to the system and to hide the underlying details from users. In order to provide platform independency, the main flow of the client application is developed using the Java language and Java Web Start (JavaWS) technology. Using JavaWS keeps users away from complex installation steps, enables version control procedures, and provides a rapid and easy web development environment for the application as a result.

There are three sub-modules for queries in the client application: *query builder*, *result fetcher* and *video player*. Queries are built using the *query builder*. *Result fetcher* is responsible for sending the constructed query to the server and receiving the results coming from the coordinator of the architecture. The status of the query, error messages and resulting semantic objects are displayed on this part. The *video player* part is for displaying visual materials and built using JMFPlayer bean.

Since multimedia data usually exists in huge sizes, transferring resulting objects from server to client may cause problems in slow connections. Therefore, an ability to select network speed for image quality is added to the client application. For debugging and testing purposes, a logger and a debugger are also implemented for the client application.

The main query types supported by the system can be divided into two main categories: text-based queries and image-based queries. In order to answer queries, some web services and Servlet interfaces are implemented in the coordinator structure of the server. Web services enable information sharing among different applications even if they are running on different platforms.

3.1 Text-Based Semantic Video Queries

As in traditional databases, the developed system is capable of executing text-based queries on previously extracted semantic contents. The objects, events and relations extracted and stored in the database are queried in this type. Beside crisp predicates, uncertain values and query conditions are acceptable in the system. In addition to stored data, the inference mechanism is activated when a query includes a rule which is defined in the knowledge-base of the system to infer new semantic information.

For constructing text-based queries in a visual manner, a user interface is dynamically created at runtime downloading an XML file from server, which includes domain information of the data stored in the database. Therefore, a user can form most part of a query selecting query components shown on the query interface. In addition, the user may fill the desired properties (query conditions) of searched contents using some crisp or uncertain values. For example, the age of a player can be entered as *very-young* or *less than 18* for a football video query.

When evaluating a query, all relevant database objects filtered by crisp and fuzzy query predicates are brought to the bridge from the fuzzy object-oriented database. If any rule needs to be fired, objects in the bridge are transferred to the working memory

of the knowledge-base. After firing needed rules, knowledge-base sends resulting objects back to the bridge and the objects meeting all predicates of the query are serialized into XML messages and returned to the client application through the coordinator. If query predicates contain uncertain information, fuzzy processors are activated in both database and knowledge-base of the system during query evaluation.

Text-based queries can be further categorized into different groups based on the semantic content queried. These query groups are summarized and some examples for them are given below:

Catalogue Queries

This type of query includes query conditions regarding the metadata of videos such as video name, date and resolution. This category uses the video attributes that can be obtained easily from the file system or video itself. A simple example can be given as follows:

- *Retrieve the video named as “the final match of the world-cup 2010”.*

Object Queries

This query type requests the video objects which constitute one of the semantic contents of the video. For example, players, referee, ball, field, goal area and goalpost are example objects in a football video. Since it takes too long time to extract this type of semantic contents automatically during a query or it is infeasible to achieve automatic extraction for some objects/object attributes, the automatic or manual extraction of semantic contents is terminated before a query process. The extracted semantic content is inserted into the database being interrelated with the relevant video segments. During an object query, the stored data within the database is retrieved as the answer to the query. If a user query requires playing of the video segment of the related object, then that video segment is also sent to the user. Some example video object queries are given below:

- *Retrieve the name of the referees in the final match of the world-cup 2010.*
- *Retrieve the name of the old defense players in the teams of the world-cup 2010 tournament.*
- *Retrieve the views of goalposts in the final match of the world-cup 2010.*

The first query includes a crisp predicate to be evaluated. In the second query, both crisp and fuzzy linguistic terms are used. The *old* word, defining the age of the player, is a fuzzy predicate, and the others are crisp predicates. Third one is an example query to retrieve fully automatically extracted objects. To answer the first and second queries, the names should be entered manually. However, the views of the goalposts can be obtained automatically from the video to answer the last query.

Event Queries

Events are the actions performed by objects in the videos. Similar to video objects, events can be extracted automatically to a certain extent or extracted manually and inserted into the database with a reference to the related video segments. Later, users may query the events as semantic contents of the video and reach related video portions. A sample text-based event query can be given as follows:

- *Retrieve the goal events in the final match of the world-cup 2010 tournament.*

Object and event queries can be combined to form more complex queries as follows:

- *Retrieve the goal events in the final match of the world-cup 2010 tournament scored by the player named Iniesta.*

Spatial Queries

The objects are typically stored with minimum bounding rectangle (MBR) which represents the coordinates of objects in the working space. Then, the spatial relations between object pairs can be extracted from the coordinates of those objects and queried by users. In this study, three types of basic spatial relations are defined, as positional relations (*above, below, left, right*), distance relations (*near, far*), and topological relations (*inside, partially inside, touch, disjoint*). More complex relations are also defined combining two simple relations such as “*near left*” or “*just (near) above*”. Spatial relations are defined as fuzzy relations, and the membership degree of each relation type is determined by its own membership function. An example fuzzy spatial query supported by our system which uses the positional relation *right* is as follows:

- *Find the player who appears on the right of Ozil (with a threshold of 0.5) in the foul event occurred at the 30th minute of the match.*

Temporal Queries

It is crucial to handle temporal relations existing in videos. Temporal relations are utilized in order to add temporality to the sequence of events. The temporal relations *before, overlaps, during, meets, starts, finishes* and *equal* are defined and used in the system. Moreover, these relations are extended as fuzzy relations such as “*long after*”, “*just before*” and “*nearly equal*” defining specific membership functions. An example temporal query can be given as follows:

- *Find all foul events happening just before the goal events (with a threshold of 0.8) in which Ozil is the scorer.*

Rule-based Semantic Queries

In the proposed architecture, a rule-based system is integrated into the database. Mostly rule-based systems are used for semantic information extraction. Here, semantic information refers to object attributes which are not stored in the database, but those extracted by deductive rules. In our system, a semantic query may involve conditions to be evaluated in the database and also conditions to be evaluated in the rule-based system. An example of a semantic query which requires rule firing can be given as follows:

- *Retrieve the matches with an enjoyment level of spectacular (with a threshold of 0.6) and played in the world-cup 2010 tournament.*

In this query, *tournament* is a condition to be evaluated in the database while the *enjoyment* level is a fuzzy rule defined in the rule-based system, as shown below:

*IF the number of goalPositions is many (with threshold=0.6)
AND the number of goals is large (with threshold=0.8)
AND the harshness level is high (with threshold=0.7)
THEN the enjoyment level of this match is spectacular (with $\mu=Y$).*

In this rule, *goalPositions*, *goals* and *harshness* are assumed to be defined as fuzzy types in the database. The above query requires the execution of this rule in the knowledge-base of the system. The objects satisfying the rule conditions are returned in the answer set.

3.2 Image-Based Video Queries (Query-By-Example)

In multimedia databases, image-based queries are sometimes more helpful than text-based searches. Having a picture of a girl or a castle, about which there is no textual information (such as name or location), the user may request other information or visual objects relevant to it. In image-based queries (query-by-example), an example object or the low level features of that object are given and the similar objects are requested from the system. In general, an example object is selected by the user using the query interface and the query is sent to the system. The system searches similar video objects recorded in the database and returns a list of the similar objects. The index structure which uses the low level features of the objects provides quick access to the related objects. Then, the user can reach the video segments of the related objects directly.

The execution of this type of queries starts with the selection of a region on the desired frame of the loaded video. This selection is the predicate of the query and the features of this region are used in calculation of its distance to other objects in the database. Section information about the selected region is sent to the system. Besides establishing communication between the server and the client for queries, the coordinator provides the required interaction with the semantic content extractor and the index structure. After receiving the query from the client, the coordinator module redirects this information to the semantic content extractor module for obtaining low-level features of the query example. The extracted low-level features are serialized and sent to the index structure for further processing. Using the serialized low-level features, the index structure finds the clusters where possible similar objects exist. For each object in these clusters, relative distances of features to the given object features are calculated and the most similar k objects are returned to the coordinator. The resulting objects are filtered by the possible object categories and ordered using the calculated similarity degrees. After being formatted, the results are sent to the client as an XML message and listed in the query result region of the interface. The related video shots are also downloaded and shown if the user wants to see them.

4 Test and Evaluation

Basically, two types of performance criteria are important for multimedia databases: speed and accuracy. The speed is important because multimedia data processing is a time-consuming action. Accuracy is important because, especially, semantic content extraction is a challenging issue and the proposed system should achieve an

acceptable accuracy. In this study, tests are executed to evaluate the performance of the implemented system in terms of speed and accuracy. They are performed on a randomly selected video data set containing 5 football and 2 basketball video sequences with nearly 10 thousand frames taken from a Turkish TV channel. While evaluating text query performance, the semantic entities extracted in the semantic information extraction process are used.

Time Measurements

Average execution times of different processes are given in Table 1. The execution time of each process is measured as the difference between the starting and completion times of the process. Since a thin-client architecture is employed, in order not to be affected by network issues, all tests are executed on the server. The speed evaluation of the system has been realized in three categories; semantic information extraction, text-based queries and image-based queries.

While evaluating the semantic information extraction capability, the detection of semantic contents and the insertion of the found entities into the object database are examined separately. Therefore, the average object and event extraction time and the insertion time of them into database are seen on consecutive rows in Table 1. Insertion times include also automatic indexing duration of the semantic contents.

Table 1. Average Execution Times for Information Extraction and Retrieval Tests

Test Type	Detailed Test Information	Time (ms)
Semantic information extraction	Object detection in a selected region	4925
	Object insertion	225
	Event detection using an object set	1386
	Event insertion	228
Text-based queries	Search for all objects of a specific type	125
	Using single crisp predicate	141
	Using multiple crisp predicates	141
	Using single fuzzy predicate	143
	Using multiple fuzzy predicates	152
	Using both crisp and fuzzy predicates	147
	No rows returned	120
	Single row returned	120
Image-based queries	Multiple rows returned	135
	Query-By-Example	4800

As seen in Table 1, the processes which include extraction of low-level features (object detection and query-by-example) take longer time than the other processes. This is because a great portion of time spent for these processes is consumed by the removal of image segments from the whole image and the extraction of low-level features of these segments by MPEG-7 Reference Software [19]. For text-based queries, there are small changes on the results depending on query details. Note that the time for downloading the results to the clients is excluded in these calculations.

Accuracy Tests

Another important criterion in evaluating the success of the implemented system is its accuracy. In the scope of this study, three tests have been performed to evaluate the success of the implemented system in terms of its accuracy. The difference among tests is the method of segmentation process. That is, the fixed size automatic segmentation, the n-cut automatic segmentation and the manual segmentation are integrated into the semantic content extractor and tested one-by-one to see their influences.

Although this section gives the accuracy test results of the semantic content extraction process, these results can be considered as the query performance of the system. It is because there will be no changes in the accuracy of this data during query processing.

The precision, recall and f-measure values which are commonly used for evaluation of accuracy of retrieval systems are calculated. Precision shows the ratio of the correctness of the query results and recall gives the ratio of the completeness of the result set. F-measure, which combines precision and recall, is the harmonic mean of precision and recall, and is calculated as $f=2(\textit{precision}*\textit{recall})/(\textit{precision}+\textit{recall})$.

In this paper, however, only the f-measure results of the 3 tests are given in Figure 2 because of the limited space. In the first test, the automatic semantic content extractor module is activated using the fixed size segmentation which divides each frame into 200 equal pieces. There are 864 objects to be extracted in the test video. The genetic algorithm-based classification module [20] classifies the given segments to a class with a membership degree which shows an uncertainty level for classification. Therefore, we measure the performance of the system by defining a threshold value and the figure shows the change of the system performance on different threshold levels.

Since the semantic content extractor module is forced to make a classification for each fixed size segment, the performance of the system is considerably low as seen in the figure. Even for high threshold values, the ratio of correctly detected objects to all retrieved objects is not satisfying and has resulted in low f-measure values.

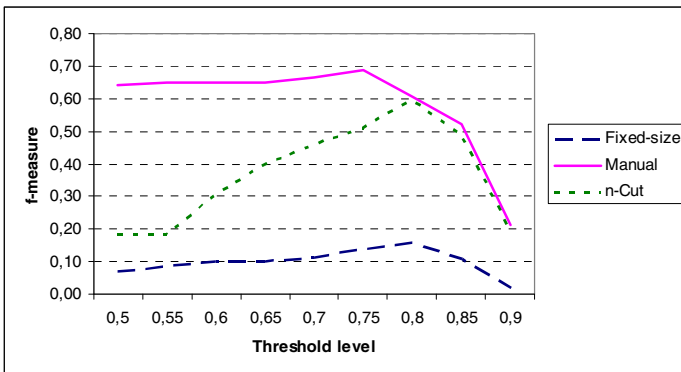


Fig. 2. Object Extraction performance with fixed size segmentation

The second test is done using the n-cut segmentation algorithm [22]. A remarkable improvement is obtained in the accuracy performance of the system as seen in Figure 2. The reason for the improvement is that the segmentation algorithm obtains more appropriate image segments for classification. With high threshold values, the precision

of the automatic object extraction process becomes higher and higher. However, it misses more and more objects in this case and the recall values go down. The threshold value 0.80 has been found as the optimum point for the f-measure value.

As for the third test, the automated segmentation is disabled and images are segmented manually, but object classification is executed automatically. As seen in the figure, the accuracy performance of the system is further improved in this case. From the f-measure values, it is understood that a better automatic segmentation algorithm may improve the performance of the system to a certain extent, but it will not be possible to reach a complete success. Therefore, the classification algorithm including its training phase also needs further improvements to obtain higher accuracy performance results.

5 Conclusions and Future Work

In this study, a video database system which supports uncertainty has been developed. The system is capable of extracting semantic information from multimedia data automatically to a certain extent. Since some semantic information about objects cannot be obtained without human interaction such as providing the name of a player who commits a foul event, the developed system also supports a semi automatic way or manual way of object and event insertion. An index structure specially designed for handling low-level features has also been utilized in this study to achieve efficient similarity searches.

The implemented video database system provides different query functionalities to satisfy different query requirements of users. The automatic content extraction functionality and query capabilities of the system have been tested to measure its speed and accuracy performances. It is evaluated that the performance test results gives positive feedbacks about feasibility of a video database system.

This is an on-going study and some improvements will follow. A future work aims to enable this system to handle audio and text parts of video data and support extraction of semantic information from them. In addition, the fusion of the extracted data from visual, audio and textual data to improve the accuracy performance of the system is another study to be done. Considering the implemented segmentation methods, a better algorithm will be another future project for a more efficient segmentation. Finally, the number of used MPEG-7 descriptors can be increased and the system can also be modified to benefit from all types of descriptors to improve classification performance of the system.

Acknowledgments. This work is supported in part by a research grant from TUBITAK EEEAG with grant number 109E014.

References

1. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-Based Multimedia Information Retrieval: State of the Art and Challenges. *ACM Trans. on Multimedia Computing, Communications, and Applications* 2, 1–19 (2006)
2. Hacid, M.S., Declair, C., Kouloumdjian, J.: A Database Approach for Modeling and Querying Video Data. *IEEE Trans. on Knowledge and Data Eng.* 12, 729–750 (2000)

3. Petkovic, M., Jonker, W.: An Overview of Data Models and Query Languages for Content-based Video Retrieval. In: *Int. Conf. on Advances in Infrastructure for E-Business, Science, and Education on the Internet, l'Aquila, Italy* (2000)
4. Marques, O., Furht, B.: MUSE: A Content-Based Image Search and Retrieval System Using Relevance Feedback. *Multimedia Tools and Applications* 17, 21–50 (2002)
5. Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: approaches and trends of the new age. In: *Proc. ACM International Workshop on Multimedia Information Retrieval*, pp. 253–262. ACM Press, New York (2005)
6. Fan, J., Elmagarmid, A.K., Member, S., Zhu, X., Aref, W.G., Wu, L.: ClassView: Hierarchical Video Shot Classification, Indexing, and Accessing. *IEEE Trans. on Multimedia* 6, 70–86 (2004)
7. Kompatsiaris, I., Avrithis, Y., Hobson, P., Strintzis, M.G.: Integrating Knowledge, Semantics and Content for User-Centered Intelligent Media Services: the aceMedia project. In: *Proc. of Workshop on Image Analysis for Multimedia Interactive Services* (2004)
8. Natarajan, P., Nevatia, R.: EDF: A framework for Semantic Annotation of Video. In: *10th IEEE Int. Conf. on Computer Vision Workshops*, p. 1876. IEEE Computer Society, Los Alamitos (2005)
9. Liu, Y., Zhang, D., Lu, G., Ma, W.: A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40, 262–282 (2007)
10. Ekin, A., Tekalp, A.M., Mehrotra, R.: Integrated Semantic-Syntactic Video Modeling for Search and Browsing. *IEEE Transactions on Multimedia* 6, 839–851 (2004)
11. Aygün, R.S., Yazici, A.: Modeling and Management of Fuzzy Information in Multimedia Database Applications. *Multimedia Tools and Applications* 24, 29–56 (2004)
12. Donderler, M.E., Saykol, E., Arslan, U., Ulusoy, O.: BilVideo: design and implementation of a video database management system. *Multimedia Tools and Applications* 27, 79–104 (2005)
13. Ozgur, B., Koyuncu, M., Yazici, A.: An intelligent fuzzy object-oriented database framework for video database applications. *Fuzzy Sets and Systems* 160, 2253–2274 (2009)
14. DB4Object, <http://www.db4o.com/>
15. Koyuncu, M., Yazici, A.: IFOOD: An Intelligent Fuzzy Object-Oriented Database Architecture. *IEEE Trans. on Knowledge and Data Engineering* 15, 1137–1154 (2003)
16. JESS, <http://www.jessrules.com>
17. Amsaleg, L., Gros, P.: Content-based Retrieval Using Local Descriptors: Problems and Issues from a Database Perspective. *Pattern Analysis and Applications* 4, 108–124 (2001)
18. Calistru, C., Ribeiro, C., David, G.: Multidimensional descriptor indexing: Exploring the bitMatrix. In: Sundaram, H., Naphade, M., Smith, J.R., Rui, Y. (eds.) *CIVR 2006. LNCS*, vol. 4071, pp. 401–410. Springer, Heidelberg (2006)
19. MPEG-7 Part-6 Reference SW: Experimentation Model (XM), ISO, <http://mpeg.chiariglione.org/technologies/mpeg-7/mp07-rsw/index.htm>
20. Yildirim, Y., Yilmaz, T., Yazici, A.: Ontology-Supported Object and Event Extraction with a Genetic Algorithms Approach for Object Classification. In: *Proc. of the 6th ACM Int. Conf. on Image and Video Retrieval (CIVR 2007)*, New York, USA, pp. 202–209 (2007)
21. Koyuncu, M., Yilmaz, T., Yildirim Y., Yazici, A.: A Framework for Fuzzy Video Content Extraction, Storage and Retrieval. In: *IEEE Int. Conf. on Fuzzy Systems* (2010), <http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=5583929>
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)

Bipolar SQLf: A Flexible Querying Language for Relational Databases

Nouredine Tamani¹, Ludovic Liétard², and Daniel Rocacher¹

¹ IRISA/ENSSAT/University Rennes 1,
6 rue de Kerampont, 22300 Lannion, France
{tamani,rocacher}@enssat.fr

² IRISA/IUT/University Rennes 1,
Rue Edouard Branly, BP 30219 22302 Lannion Cedex, France
ludovic.lietard@univ-rennes1.fr

Abstract. Flexible querying of information systems allows expressing complex preferences in user queries. Such preferences can be modeled by fuzzy bipolar conditions which are made of constraints c and wishes w and interpreted as "to satisfy c and if possible to satisfy w ". We define in this article the main elements of the Bipolar SQLf language, which is an SQL-like querying language based on a bipolar relational algebra [13]. This language is an extension of the SQLf language [21]. Basic statements (projection, selection, etc.) are firstly defined in terms of syntax, evaluation and calibration. Then, complex statements, such as bipolar queries based on nesting operators are studied in terms of expression, evaluation, query equivalence and backward compatibility with the SQLf language.

Keywords: Flexible Querying, Fuzzy Bipolar Conditions, Fuzzy Bipolar Algebra, SQLf Language, Bipolar SQLf Language.

1 Introduction

Flexible querying of databases allows users to express preferences in their queries. Within this framework, numerous tools have been proposed such as Preference SQL [5], SQLf language [21], Top- k queries [9], the winnow operator [8], etc. which are based on diverse mathematic foundations.

In the context of fuzzy querying, user preferences are expressed by fuzzy predicates (such as *high*, *fast*, *expensive*, etc.) which are defined by fuzzy sets [13]. The SQLf language is an extension of the SQL language to fuzzy conditions, which allows expressing queries addressed to relational databases. Such queries deliver a set of tuples attached with degrees used to rank them from the most to the least preferred. In this context, it is also possible to consider fuzzy bipolar conditions to model preferences.

A bipolar condition is a compound condition made of negative and positive conditions. Several interpretations have been introduced for the evaluation of queries involving such conditions (see [7,6,12,14,15]). In this paper, we rely on

the interpretation introduced by Dubois and Prade [7,6], in which a bipolar condition is made of a constraint c and a wish w . It is noted (c, w) and means "to satisfy c and if possible to satisfy w ". More precisely, for the expression of user preferences, we rely on fuzzy bipolar conditions in which the constraint and the wish are defined by fuzzy sets. Furthermore, we define a fuzzy bipolar query as a query that involves fuzzy bipolar conditions. More precisely, when querying a relation R with a fuzzy bipolar condition, each tuple t from R is then attached with a pair of grades $(\mu_c(t), \mu_w(t))$ that expresses the degree of its satisfaction to c and w and a so-called fuzzy bipolar relation is obtained.

In order to define a bipolar relational algebra, the algebraic operators (selection, projection, join, union, intersection) have been extended to fuzzy bipolar conditions [11,3]. These operators allow the expression of fuzzy bipolar queries.

We are aimed in this article to define the Bipolar SQLf language which is an SQL-like language based on a bipolar relational algebra. Since fuzzy bipolar conditions generalize fuzzy conditions, we consider the enrichment to fuzzy bipolar conditions of the SQLf language [2,1] which is devoted to flexible querying with fuzzy sets. At the first step basic Bipolar SQLf queries are defined in terms of expression, evaluation and calibration. Then, complex bipolar queries based on nesting ($in_=$, in_{\approx} , $exists$, θany) and partitioning operators are defined.

The remainder of this paper is as follows. In section 2, both fuzzy sets theory and the SQLf language are described. Section 3 introduces respectively fuzzy bipolar conditions, the bipolar relational algebra defined in [11,3] and the basis of the Bipolar SQLf language. In section 4, the extension of advanced Bipolar SQLf statements to fuzzy bipolar conditions is studied. Section 5 sums up our contribution and draws some lines for future works.

2 Flexible Querying within the SQLf Language

We introduce in this section the fuzzy sets theory and the SQLf language.

2.1 The Fuzzy Sets Theory

The fuzzy sets theory is introduced by Zadeh [13] to express the gradual membership of an element to a set. Formally, a fuzzy set F is defined on a referential U by a membership function $\mu_F : U \mapsto [0, 1]$ such that $\mu_F(x)$ denotes the membership grade of x in F . In particular, $\mu_F(x) = 1$ denotes the full membership of x in F , $\mu_F(x) = 0$ expresses the absolute non-membership and when $0 < \mu_F(x) < 1$, it reflects a partial membership (the closer to 1 $\mu_F(x)$, the more x belongs to F).

A fuzzy set generalizes an ordinary (crisp) set in which membership grades are in $\{0, 1\}$. If a fuzzy set is a discrete set then it is denoted $F = \{\mu_F(x_1)/x_1, \dots, \mu_F(x_n)/x_n\}$, otherwise, it is characterized by its membership function, generally a trapezoidal function.

The union \cup and the intersection \cap operators are defined with a couple of a t-norm and a t-conorm, such as (\min, \max) . Let F, G be two fuzzy sets, $\mu_{F \cup G}(x) = \max(\mu_F(x), \mu_G(x))$, $\mu_{F \cap G}(x) = \min(\mu_F(x), \mu_G(x))$, and the complement of F , noted F^c , is defined by $\mu_{F^c}(x) = 1 - \mu_F(x)$.

The logical counterparts of \cap , \cup and the complement are resp. \wedge , \vee and \neg . Other operators have also been defined such as fuzzy implications [4].

2.2 The SQLf Language

The SQLf language [21] is an extension of the SQL language to flexible querying based on fuzzy conditions. An SQLf query delivers a fuzzy relation r where the grade of membership of tuple t expresses its level of satisfaction.

The SQLf language is based on the fuzzy relational algebra, in which relational algebra operators have been extended to fuzzy predicates as follows: let r, s be two fuzzy relations such that the schema of r (resp. s) is X (resp. Y):

Fuzzy projection: $\mu_{\pi(r,V)}(v) = \max_w \mu_r(vw)$, where $V \subseteq X$, $v \in V$ and $w \in (X - V)$.

Fuzzy selection: $\mu_{\sigma(r,p)}(x) = \min(\mu_r(x), \mu_p(x))$, where p is a fuzzy predicate.

Fuzzy join: $\mu_{\bowtie}(r, s, \theta, A, B) = \min(\mu_r(x), \mu_s(y), \mu_{\theta}(x.A, y.B))$, where A and B are compatible sets of attributes such that A (resp. B) is a subset of X (resp. Y) and $x.A$ (resp. $y.B$) is the value of A in x (resp. B in y), and θ is either a crisp or a fuzzy binary operator ($\theta \in \{=, \approx, <, >, \text{much larger than}, \dots\}$).

The basic form of an SQLf query is a fuzzy restriction defined as follows:

Select [*distinct*] [$n|t|n,t$] *attributes* **From** *relations* **Where** *fuzzy_cond*;

This query returns a set of ranked tuples with their attached degree, where n specifies an n -top query and $t \in]0, 1]$ is a minimal threshold of satisfaction.

Example 1: Let *fast* be a fuzzy predicate defined on $\mathbb{R}_+ \rightarrow [0, 1]$: $\mu_{fast}(d) = 1$, if $d \in [0, 2]$; $\frac{d}{3} + \frac{5}{3}$, if $d \in [2, 5]$ and 0, otherwise.

The query "find the 2 fastest journeys from Brussels to Paris" can be expressed in SQLf by:

Select 2 #journey **From** Journey **Where**
source='Brussels' and destination='Paris' and fast (duration);

The fuzzy condition *fast* delivers the fuzzy relation *FastJourney*, where $\mu_{FastJourney}(t) = \mu_{Fast}(t.duration)$. Table 1 is an example of the fuzzy relation *fastJourney* and journey #12 and either journey #13 or #10 are delivered. ■

Table 1. Extension of the fuzzy relation *fastJourney*

#Journey	cost (\$)	duration (h)	...	$\mu_{FastJourney}$
12	70	2	...	1
13	50	3	...	0.66
10	50	4	...	0.66

The SQLf language allows the expression of more complex statements such as partitioning, nesting and division involving fuzzy relations. For example, the query "find journeys for which most of their steps are comfortable" corresponds to a partitioning based on fuzzy quantified propositions.

3 Flexible Querying and Bipolarity

In this section, we introduce fuzzy bipolar conditions, a bipolar relational algebra [11,3] and the basis of Bipolar SQLf language.

3.1 Fuzzy Bipolar Conditions

A bipolar condition is a compound condition which is made of two conditions defined on the same universe: i) a constraint c , which describes the set of acceptable elements, and ii) a wish w which defines the set of desired elements. Since it is incoherent to wish a rejected element, the property of coherence $w \subseteq c$ holds.

It is worth mentioning that the linguistic expression of a fuzzy condition may not follow the coherence property. As an example, a user may think of "a *japanese car and if possible a red car*", however, such a condition should be clearly be rewritten "a *japanese car, and if possible a japanese and red car*".

In addition, condition c is mandatory in the sense that an element which does not satisfy it is rejected. Condition w is optional in the sense that an element which does not satisfy it is not necessarily rejected.

If c and w are boolean conditions, the satisfaction with respect to (c, w) is a pair from $\{0, 1\}^2$. When querying a database with such a condition, tuples satisfying the constraint and the wish are returned in priority to the user. If such answers do not exist, tuples satisfying only the constraint are delivered.

If c and w are fuzzy conditions (defined on the universe U), the property of coherence becomes: $\forall u \in U, \mu_w(u) \leq \mu_c(u)$ and the satisfaction with respect to (c, w) is a pair of degrees from $[0, 1] \times [0, 1]$. Each element u from U is then attached with a pair of grades $(\mu_c(u), \mu_w(u))$ that expresses the degree of its satisfaction respectively to the constraint c and to the wish w .

In the context of bipolar relations, a tuple t is then denoted $(\mu_c, \mu_w)/t$. We assume that any tuple u such that $\mu_c(u) = 0$ does not belong to the fuzzy bipolar relation. In addition, tuples cannot be ranked from the most to the least preferred using an aggregation of μ_c and μ_w because constraints and wishes are not commensurable. However they can be ranked using the lexicographical order: t_1 is preferred to t_2 , denoted $t_1 > t_2$ or $(\mu_c(t_1), \mu_w(t_1)) > (\mu_c(t_2), \mu_w(t_2))$, iff $\mu_c(t_1) > \mu_c(t_2)$ or $(\mu_c(t_1) = \mu_c(t_2) \wedge \mu_w(t_1) > \mu_w(t_2))$. Since the constraint is mandatory, its satisfaction is firstly used to discriminate among answers. The satisfaction with respect to the wish being not mandatory, it can only be used to discriminate among answers having the same evaluation with respect to the constraint. A total order is then obtained on c and w (with $(1, 1)$ as the greatest element and $(0, 0)$ as the least element).

Based on the lexicographical order, the *lmin* and *lmax* operators [10,3] are used to define the conjunction (resp. intersection) and the disjunction (resp. union) of bipolar conditions (resp. relations).

They are defined on $([0, 1] \times [0, 1])^2 \rightarrow [0, 1] \times [0, 1]$ as follows:

$$((\mu, \eta), (\mu', \eta')) \mapsto \text{lmin}((\mu, \eta), (\mu', \eta')) = \begin{cases} (\mu, \eta) & \text{if } \mu < \mu' \vee (\mu = \mu' \wedge \eta < \eta'), \\ (\mu', \eta') & \text{otherwise.} \end{cases}$$

$$((\mu, \eta), (\mu', \eta')) \mapsto \text{lmax}((\mu, \eta), (\mu', \eta')) = \begin{cases} (\mu, \eta) & \text{if } \mu > \mu' \vee (\mu = \mu' \wedge \eta > \eta'), \\ (\mu', \eta') & \text{otherwise.} \end{cases}$$

The *lmin* (resp. *lmax*) operator is commutative, associative, idempotent and monotonic. The pair of grades (1, 1) is the neutral (resp. absorbing) element of the operator *lmin* (resp. *lmax*) and the pair (0, 0) is the absorbing (resp. neutral) element of the operator *lmin* (resp. *lmax*).

It can be noticed that a fuzzy predicate is a fuzzy bipolar predicate such that $\forall x, (\mu_c(x) = \mu_w(x))$, which means that a fuzzy condition is a fuzzy bipolar condition in which the wish is equal to the constraint. In other words, fuzzy bipolar conditions generalize fuzzy conditions and it has been proven [10,3] that *lmin* (resp. *lmax*) generalizes the t-norm *min* (resp. the t-conorm *max*).

3.2 Basis of the Bipolar Relational Algebra

We introduce the bipolar relational algebra proposed in [11,3]. It is based on the couple (*lmin*, *lmax*). Let *r* and *s* be two fuzzy bipolar relations defined respectively by the fuzzy bipolar conditions (c_1, w_1) and (c_2, w_2).

The intersection: $r \cap s$ is a fuzzy bipolar relation defined as follows:

$$r \cap s = \{(\mu_c, \mu_w)/t \mid (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t \in s \wedge (\mu_c, \mu_w) = \text{lmin}((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t), \mu_{w_2}(t)))\}.$$

The union: $r \cup s$ is a fuzzy bipolar relation defined as follows:

$$r \cup s = \{(\mu_c, \mu_w)/t \mid (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t \in s \wedge (\mu_c, \mu_w) = \text{lmax}((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t), \mu_{w_2}(t)))\}.$$

The cartesian product: $r \otimes s$ is a fuzzy bipolar relation defined as follows:

$$r \otimes s = \{(\mu_c, \mu_w)/t \oplus t' \mid (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_{c_2}, \mu_{w_2})/t' \in s \wedge (\mu_c, \mu_w) = \text{lmin}((\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{c_2}(t'), \mu_{w_2}(t'))\},$$

where \oplus is the operator of concatenation of tuples.

The projection π : the projection of distinct tuples of *r* on attributes a_1, \dots, a_k is a fuzzy bipolar relation of tuples $\langle a_1, \dots, a_k \rangle$ defined by:

$$\pi_{a_1, \dots, a_k}(r) = \{(\mu'_{c_1}, \mu'_{w_1})/\langle a_1, \dots, a_k \rangle \mid (\mu'_{c_1}, \mu'_{w_1}) = \text{lmax}_{t \in r \wedge t[a_1, \dots, a_k] = \langle a_1, \dots, a_k \rangle}((\mu_{c_1}(t), \mu_{w_1}(t)))\},$$

where $t[a_1, \dots, a_k]$ is the value of the tuple *t* on the attributes a_1, \dots, a_k .

The selection σ : the selection of tuples from *r*, based on the fuzzy bipolar condition (c', w') is defined as:

$$\sigma(r, (c', w')) = \{(\mu_c, \mu_w)/t \mid (\mu_{c_1}, \mu_{w_1})/t \in r \wedge (\mu_c, \mu_w) = \text{lmin}((\mu_{c'}(t), \mu_{w'}(t)), (\mu_{c_1}(t), \mu_{w_1}(t)))\}.$$

The join operator \bowtie : as in the SQL and SQLf languages, the join operator is defined by the bipolar selection operator applied over a bipolar cartesian product. This operator is studied in Section 4.

3.3 Bipolar SQLf Basic Statements

A Bipolar SQLf basic statement is a fuzzy bipolar selection applied over a bipolar projection operator. It has the following form:

Select [*distinct*] [*n*|*t*|(*t*₁, *t*₂)|*n*, *t*|*n*, (*t*₁, *t*₂)] *attributes* **From** *relations* [*as alias*] **Where** (*c*, *w*);

The parameters intended to calibration of the result are also extended to bipolarity. A bipolar top *k* query is obtained by positioning the optional integer *n*, which delivers the *n* best tuples attached with the *n* greatest pairs of degrees in the sense of the lexicographical order. The qualitative calibration can be specified by either a threshold *t* ∈ [0, 1], which delivers tuples *u* such that (μ_{*c*}(*u*), μ_{*w*}(*u*)) ≥ (*t*, 0), or a pair of thresholds (*t*₁, *t*₂) ∈ [0, 1]², such that *t*₂ ≤ *t*₁, which delivers tuples *u* such that (μ_{*c*}(*u*), μ_{*w*}(*u*)) ≥ (*t*₁, *t*₂).

Example 2: The query “find the 2 best journeys which are fast and if possible not expensive”, can be expressed in Bipolar SQLf as:

Select 2 #Journey **From** Journey **as** J **Where** (fast(*J.duration*), not expensive(*J.cost*));

Due to the coherence property of fuzzy bipolar conditions, the fuzzy bipolar condition “fast and if possible and not expensive” is interpreted as “fast and if possible (fast and not expensive)”.

The fuzzy predicate *expensive* can be defined as $\forall x \in \mathbb{R}_+ : \mu_{Expensive}(x) = \frac{x}{80}$, if *x* ∈ [0, 80]; 1, otherwise; where *x* expresses the cost of a journey. Its negation is defined as follows: $\forall x \in \mathbb{R}_+, \mu_{notExpensive}(x) = 1 - \mu_{Expensive}(x)$.

Based on the definition of fuzzy predicates *fast* (example 1) and *not expensive*, the query is evaluated over the relation *Journey* and delivers the fuzzy bipolar relation *Journey*_(Fast, notExpensive) (see Table 2). The returned tuples are ranked using the lexicographical order : (1, 0.13)/12, (0.66, 0.38)/13. The tuple #12 is the best one with regard to the constraint (total satisfaction), and tuples #13 and #10 have the same satisfaction with respect to the constraint but #13 is better than #10 on the wish. ■

Table 2. Fuzzy bipolar relation *Journey*_(Fast, notExpensive)

#Journey	cost (\$)	duration (h)	μ _{Fast}	μ _{Fast} ∧ μ _{notExpensive}
12	70	2	1	0.13
13	50	3	0.66	0.38
10	50	4	0.66	0.33

4 Extension of Complex SQLf Statements to Bipolarity

In this section, we define the bipolar join operator. Then, the extension to bipolarity of nesting operators (*in*₌, *in*_≈, *exists*, *thany*) and aggregate functions are defined in the scope of their equivalence to the bipolar join operator. Since fuzzy bipolar conditions generalize fuzzy conditions, the bipolar definition of these operators is based on the extension to bipolarity of the SQLf statements.

4.1 Extension of the Join Operator to Bipolarity

The basic form of an SQLf join query is as follows:

Q_1 : *Select R.A, R'.B From R, R' Where $c_1(R)$ and $c_2(R')$ and $R.att_1 = R'.att_2$;*
 where c_1 and c_2 are two fuzzy conditions applied resp. on relations R and R' .

Tuples $u = (a, b)$ delivered from Q_1 are attached with degrees processed by the following formula (1):

$$\mu_{Q_1}(u) = \max_{t \in R \wedge t' \in R' \wedge t.A = a \wedge t'.B = b} \min(\mu_R(t), \mu_{c_1}(t), \mu_{R'}(t'), \mu_{c_2}(t'), \mu_{=(t.att_1, t'.att_2)}) \quad (1)$$

The formula (1) can be rewritten as follows:

$$\mu_{Q_1}(u) = \max_{t \in R \wedge t' \in R' \wedge t.A = a \wedge t'.B = b \wedge t.att_1 = t'.att_2} \min(\mu_R(t), \mu_{c_1}(t), \mu_{R'}(t'), \mu_{c_2}(t')) \quad (2)$$

In the context of bipolarity, the basic form of a join query is as follows:

Q_2 : *Select R.A, R'.B From R, R' Where*
 $(c_1(R), w_1(R))$ and $(c_2(R'), w_2(R'))$ and $R.att_1 = R'.att_2$;

The definition of the join operator based on the formula (1) is as follows:

$$(\mu_{cQ_2}(u), \mu_{wQ_2}(u)) = \max_{t \in R \wedge t' \in R' \wedge t.A = a \wedge t'.B = b} \min((\mu_{R_c}(t), \mu_{R_w}(t)), (\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')), (\mu_{=(t'.att_2, t.att_1), \mu_{=(t'.att_2, t.att_1)})) \quad (3)$$

Based on the formula (2), tuples $u = (a, b)$ delivered from Q_2 are attached with pairs of degrees processed by the following formula (4):

$$(\mu_{cQ_2}(u), \mu_{wQ_2}(u)) = \max_{t \in R \wedge t' \in R' \wedge t.A = a \wedge t'.B = b \wedge t.att_1 = t'.att_2} \min((\mu_{R_c}(t), \mu_{R_w}(t)), (\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')))) \quad (4)$$

It is easy to prove that formulas (4) and (3) are equivalent.

Remark: A bipolar θ -join operator, where θ is either a boolean or a fuzzy relational operator ($\theta \in \{<, >, \leq, \geq, =, \neq, \textit{around}, \textit{much greater than}, \dots\}$), can straightforwardly be defined from formula (3) by substituting

$(\mu_{=(t'.att_2, t.att_1), \mu_{=(t'.att_2, t.att_1)})$ by $(\mu_{\theta}(t'.att_2, t.att_1), \mu_{\theta}(t'.att_2, t.att_1))$.

4.2 Bipolar (θ_c, θ_w) -Join Operator

We define a new bipolar join operator denoted (θ_c, θ_w) -join made of two relational operators: θ_c and θ_w which are in $\{<, >, \leq, \geq, =, \neq, \textit{around}, \textit{greater than}, \dots\}$. This bipolar operator permits us to express queries such as "find salespersons

who get a turnover much greater **and if possible** very much greater than 10 times their own salary". The main form of such queries is:

Q'_2 : Select $R.A, R'.B$ From R, R' Where $(c_1(R), w_1(R))$ and $(c_2(R'), w_2(R'))$ and $(R.att_1 \theta_c R'.att_2, R.att_3 \theta_w R'.att_4)$;

Based on the formula (3), pairs of degrees associated to tuples delivered from Q'_2 are processed by the following formula (5):

$$(\mu_{cQ'_2}(u), \mu_{wQ'_2}(u)) = \underset{t \in R \wedge t' \in R' \wedge t.A = a \wedge t'.B = b}{lmax} \underset{lmin((\mu_{R_c}(t), \mu_{R_w}(t)), (\mu_{c_1}(t), \mu_{w_1}(t)), (\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')), (\mu_{\theta_c}(t'.att_2, t.att_1), \mu_{\theta_w}(t'.att_3, t.att_4)))}{lmin} \quad (5)$$

Example 3: In order to select the best young sellers, a manager based on the monthly balance sheets can express the following query "find young **and if possible** very young salespersons with turnovers of their low **and if possible** very low monthly balance sheets, are much greater **and if possible** very much greater than 5 times their salary". It can be written in Bipolar SQLf as:

Select #Seller From Seller as S, MonthBalance as MB Where $S.\#Seller = MB.\#Seller$ and (young (S.age), very young (S.age)) and (low (MB.turnover), very low (MB.turnover)) and (much greater (S.salary*5, MB.turnover), very much greater (S.salary*5, MB.turnover));

Due to the space limitation, we only describe the derived fuzzy bipolar relations: $Seller_{(Young, veryYoung)}$ (see Table 3) and $Balance_{(low, veryLow)}$ (see Table 4), in which we show the pairs of degrees of satisfaction to the bipolar join condition (much greater(S.salary*5, MB.turnover), very much greater(S.salary*5, MB.turnover)).

The predicate low is defined on $\mathbb{R}_+ \rightarrow [0, 1]$ as: $\mu_{low}(x) = 1$ if $x \in [0, 25000]$, $\mu_{low}(x) = \frac{-x}{5000} + 6$ if $x \in [25000, 30000]$, $\mu_{low}(x) = 0$ otherwise.

The predicate very low is defined as $\forall x \in \mathbb{R}_+ : \mu_{veryLow}(x) = (\mu_{low}(x))^2$.

We define the predicate much greater on $(\mathbb{R}_+^2 \rightarrow [0, 1])$: $\mu_{muchGreater}(x, y) = 1 - \frac{y}{x}$ if $x > y$; 0, otherwise; and the predicate very much greater is defined: $\forall (x, y) \in \mathbb{R}_+^2 : \mu_{vMGreater}(x, y) = (\mu_{mGreater}(x, y))^2$.

Table 3. The fuzzy bipolar relation $Seller_{(young, veryYoung)}$

#Seller	salary	age	μ_{young}	$\mu_{veryYoung}$
5	2000	24	1	1
1	2500	30	0.8	0.64
3	2800	38	0.2	0.04

The formula (5) is used to compute the pair of grades for the seller #1:

$$(\mu_c(\#1), \mu_w(\#1)) = lmax(lmin((0.8, 0.64), (0.5, 0.25), (0.55, 0.33)), lmin((0.8, 0.64), (0.7, 0.49), (0.53, 0.28))) = lmax((0.5, 0.25), (0.53, 0.28)) = (0.53, 0.28).$$

Table 4. The fuzzy bipolar relation $Balance_{(low,veryLow)}$

#Balance	#Seller	salary (\$)	turnover (\$)	μ_{low}	$\mu_{veryLow}$	$\mu_{mGreater}$	$\mu_{vMGreater}$
1	1	2500	27500	0.5	0.25	0.55	0.30
2	1	2500	26500	0.7	0.49	0.53	0.28
5	3	2800	28500	0.3	0.09	0.56	0.32
6	3	2800	28000	0.4	0.16	0.55	0.31
9	5	2000	29500	0.1	0.01	0.66	0.44
10	5	2000	25000	1	1	0.60	0.36

For sellers #3 and #5, the attached pairs of grades are respectively:

$$(\mu_c(\#3), \mu_w(\#3)) = (0.2, 0.04) \text{ and } (\mu_c(\#5), \mu_w(\#5)) = (0.6, 0.36).$$

Sellers are delivered as follows: $(0.6, 0.36)/\#5, (0.53, 0.28)/\#1, (0.2, 0.04)/\#3$. ■

4.3 Extension of $in_=$ and in_{\approx} Operators to Bipolarity

In the SQLf language, the $in_=$ (resp. in_{\approx}) operator expresses at what level a value of an attribute is equal (resp. is close) to a value from the fuzzy set delivered by the nested subquery. The main format of an in_{θ} query, where $\theta \in \{=, \approx\}$, in the SQLf language is:

Q_3 : *Select A From R Where c_1 and att_1 in_{θ} (Select att_2 From R' Where c_2);*

The query Q_3 is equivalent to the following join query [2]:

Q_4 : *Select R.A From R, R' Where $R.att_1 \theta R'.att_2$ and $c_1(R)$ and $c_2(R')$;*

Based on this equivalence, the evaluation of the condition v_1 such that:

$v_1 = att_1 in_{\theta}$ (Select att_2 From R' Where c_2) is as follows:

$$\mu_{v_1} = \max_{t' \in R'} \min(\mu_{R'}(t'), \mu_{c_2}(t'), \mu_{\theta}(t'.att_2, t.att_1)) \quad (6)$$

This equivalence holds in the case of bipolarity. The condition v_1 is written $v'_1 = att_1 in_{\theta}$ (Select att_2 From R' Where (c_2, w_2)).

The evaluation of the condition v'_1 is based on the extension of the formula (6) to bipolarity as follows:

$$(\mu_{c.v'_1}, \mu_{w.v'_1}) = \underset{t' \in R'}{lmax} \underset{t' \in R'}{lmin}((\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')), (\mu_{\theta}(t'.att_2, t.att_1), \mu_{\theta}(t'.att_2, t.att_1))) \quad (7)$$

4.4 Bipolar $in_{(\approx,=)}$ Operator

It is possible to define a bipolar in operator denoted $in_{(\approx,=)}$ which expresses conditions v_2 of the following form:

$v_2 = att_1 in_{(\approx,=)}$ (Select att_2 From R' Where (c_2, w_2)), which expresses at what level att_1 is close **and if possible** is equal to a value among those delivered from the bipolar SQLf subquery (Select att_2 From R' Where (c_2, w_2)). From the syntactic point of view the bipolar operator $in_{(\approx,=)}$ is expressed (*approx, equal*).

The evaluation of the condition v_2 is based on the following formula (8), which is an extension to bipolarity in the formula (6):

$$(\mu_{c_{-}v_2}, \mu_{w_{-}v_2}) = \underset{t' \in R'}{lmax} \underset{t' \in R'}{lmin} ((\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')), (\mu_{\approx}(t'.att_2, t.att_1), \mu_{=}(t'.att_2, t.att_1))) \quad (8)$$

It is worth noticing that a query defined with the bipolar $in_{(\approx, =)}$ operator is equivalent to a (θ_c, θ_w) -join query where θ_c corresponds to the \approx operator and θ_w corresponds to the $=$ operator.

Example 4: We consider the following query "find villas which are small, and if possible not far from the downtown and having a price similar, and if possible equal to the price of apartments which are spacious, and if possible located near to the downtown". It can be expressed in the Bipolar SQLf as follows:

Select #villa From Villa as V Where (small(V.surface), not far_town(V.address) and V.price (approx, equal) (Select #apart From Apartment as A Where (spacious (A.surface), near_town (A.address))); ■

4.5 Extension of the exists Operator to Bipolarity

In the scope of the SQLf language, the *exists* operator indicates a non emptiness of a fuzzy set. It is defined by the formula $\mu_{exists}(E) = \sup_{x \in support(E)} \mu_E(x)$ which expresses at what extent an element belongs to the returned fuzzy set.

The main form of an *exists* query in the SQLf language is:

Q_5 : Select A From R Where c_1 and exists (Select * From R' Where c_2 and $R.att_1 \theta R'.att_2$);

where θ is a relational operator which can be either boolean or fuzzy.

This query is equivalent to the following join query:

Q_6 : Select A From R, R' Where $c_1(R)$ and $c_2(R')$ and $R.att_1 \theta R'.att_2$;

The condition v_3 defined by the *exists* operator:

$v_3 = exists$ (Select * From R' Where c_2 and $R.att_1 \theta R'.att_2$) is evaluated by the following formula (9):

$$\mu_{v_3} = \underset{t' \in R'}{max} \underset{t' \in R'}{min} (\mu_{R'}(t'), \mu_{c_2}(t'), \mu_{\theta}(t.att_1, t'.att_2)) \quad (9)$$

This interpretation preserves the equivalence between the *exists* operator and the $in_{=}$ and in_{\approx} operators, when $\theta \in \{\approx, =\}$ and with the join operator.

In the context of bipolarity, these equivalences hold, and the condition v_3 is rewritten as $v'_3 = exists$ (Select * From R' Where (c_2, s_2) and $R.att_1 \theta R'.att_2$).

The condition v'_3 is evaluated by the following formula (10):

$$(\mu_{c_{-}v'_3}, \mu_{w_{-}v'_3}) = \underset{t' \in R'}{lmax} \underset{t' \in R'}{lmin} ((\mu_{R'_c}(t'), \mu_{R'_w}(t')), (\mu_{c_2}(t'), \mu_{w_2}(t')), (\mu_{\theta}(t.att_1, t'.att_2), \mu_{\theta}(t.att_1, t'.att_2))) \quad (10)$$

From the formula (10), we can define the *exist* operator as the retrieval of the greatest pair of grades which satisfies the imbricated query.

4.6 The Extension of the θany Operator to Bipolarity

In the SQLf language, a query involving θany can be rewritten as a query involving the $exists$ operator. This equivalence is also valid in the Bipolar SQLf language. As consequence, the two following queries are equivalent:

Q_7 : *Select A From R Where (c_1, w_1) and $att_1 \theta any$
(Select att_2 From R' Where (c_2, w_2));*

Q_8 : *Select A From R Where (c_1, w_1) and $exists$
(Select att_2 From R' Where (c_2, w_2) and $R.att_1 \theta R'.att_2$);*

The evaluation of a θany query relies then on the formula (10).

4.7 Extension of Aggregate Functions Based Query to Bipolarity

Aggregate functions such as sum , $count$, avg , min , max are used to perform arithmetic operations over a set of tuples. The following query expresses the main form of a SQLf query based on aggregate functions:

Q_9 : *Select A From R Where c Group By A Having
 $c_{f_1}(agg_1(att_1)) cnt \dots cnt c_{f_n}(agg_n(att_n))$;*

where c is a boolean condition, agg_1, \dots, agg_n are aggregate functions which are applied resp. on attributes att_1, \dots, att_n . The returned values are, then, used as parameters for fuzzy conditions c_{f_1}, \dots, c_{f_n} to determine their grades of satisfaction. Finally, the obtained grades are combined depending on connectors cnt .

The same principal of partitioning is used in the case of fuzzy bipolar conditions. The following query is the main form of such a partitioning:

Q'_9 : *Select A From R Where c Group By A Having
 $(c_1(agg_1(att_1)), w_1(agg_1(att_1))) cnt \dots cnt (c_n(agg_n(att_n)), w_n(agg_n(att_n)))$;*

where cnt can be either an *and* or an *or* operator. The combination of the pairs of grades returned by $(c_i, w_i), i = 1 \dots n$ is based on $lmin$ and/or $lmax$ operators.

5 Conclusion

This article has considered the definition of an SQL-like language to express preferences defined by fuzzy bipolar conditions. Such fuzzy bipolar conditions are extensions of fuzzy conditions. This language (namely Bipolar SQLf language) is an extension of the SQLf language to bipolarity. It is based on a relational bipolar algebra that defines basis operators and provides a well appropriate interpretation for each language statement. In this article, we have defined basic statements (projection, selection, join, etc.) and nesting operators such as $in=$, in_{\approx} , $exists$ and θany .

As future works, we aim at extending the language to fuzzy bipolar quantified propositions, to be able to express queries based on linguistic quantifiers such as "find stores that have **most of** their sellers are young **and if possible well paid**", and to study queries corresponding to divisions involving fuzzy

bipolar relations, such as *"find students who are well scored **and if possible** very well scored in all difficult **and if possible** in all very difficult courses"*. An implementation of a prototype for query evaluation is in progress, and in order to provide users with personalized services, this language is intended to be integrated into a platform of flexible querying of heterogeneous and distributed information systems developed in the field of multimodal transportation networks, in which complex queries could be expressed such as: *"find journeys from Lannion to Brussels which are fast and having early departures, **and if possible** not expensive and having steps which go through stations in which are located good restaurants which serve health foods **and if possible** not expensive"*.

Acknowledgments. We warmly thank the Brittany region, the department of Côtes-d'Armor and the National Agency for Research (AOC Ref. ANR-08-CORD-009) for financing this work.

References

1. Bosc, P., Liétard, L., Pivert, O., Rocacher, D.: Base de données - Gradualité et imprécision dans les bases de données Ensembles flous, requêtes flexibles et interrogation de données mal connues. Technosup, 1st edn. (2004)
2. Bosc, P., Pivert, O.: Sqlf: A relational database langage for fuzzy querying. IEEE Transactions on Fuzzy Systems 3(1), 1–17 (1995)
3. Bosc, P., Pivert, O., Liétard, L., Mokhtari, A.: Extending relational algebra to handle bipolarity. In: 25th ACM Symposium on Applied Computing, SAC 2010, pp. 1717–1721 (2010)
4. Bouchon-Meunier, B., Dubois, D., Godo, L., Prade, H.: Fuzzy set and possibility theory in approximate and plausible reasoning. In: Fuzzy Sets in Approximate Reasoning and Information Systems, ch.1, pp. 27–31. The Handbook of fuzzy sets, Kluwer Academic Publishers (1999)
5. Chomicki, J.: Querying with intrinsic preferences. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 34–51. Springer, Heidelberg (2002)
6. Dubois, D., Prade, H.: Bipolarité dans un processus d'interrogation flexible. In: Rencontres francophones sur la Logique Floue et ses Applications, LFA (2002)
7. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS (LNAI), vol. 2522, pp. 174–182. Springer, Heidelberg (2002)
8. Kießling, W.: Foundation of preferences in database systems. In: Proceedings of the 28th VLDB Conference, Hong Kong, China (2002)
9. Li, C., Chang, K.C.C., Ilyas, I.F., Song, S.: Ranksql: Query algebra and optimization for relational top-k queries. In: ACM (ed.) SIGMOD, Baltimore, Maryland, USA (2005)
10. Liétard, L., Rocacher, D.: On the definition of extended norms and co-norms to aggregate fuzzy bipolar conditions. In: IFSA/EUSFLAT, pp. 513–518 (2009)
11. Liétard, L., Rocacher, D., Bosc, P.: On the extension of sql to fuzzy bipolar conditions. In: The 28th North American Information Processing Society Annual Conference, NAFIPS 2009 (2009)

12. De Tré, G., Zadrozny, S., Matthé, T., Kacprzyk, J., Bronselaer, A.: Dealing with positive and negative query criteria in fuzzy database quering bipolar satisfaction degrees. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS(LNAI), vol. 5822, pp. 593–604. Springer, Heidelberg (2009)
13. Zadeh, L.: Fuzzy sets. *Information and Control* 8(3), 338–353 (1965)
14. Zadrozny, S., Kacprzyk, J.: Bipolar queries using various interpretations of logical connectives. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 181–190. Springer, Heidelberg (2007)
15. Zadrozny, S., Kacprzyk, J.: Bipolar queries: An approach and its various interpretations. In: *Proceedings of IFSA/EUSFLAT*, pp. 1288–1293 (2009)

On the Behavior of Indexes for Imprecise Numerical Data and Necessity Measured Queries under Skewed Data Sets

Carlos D. Barranco¹, Jesús R. Campaña², and Juan M. Medina²

¹ Division of Computer Science, School of Engineering, Pablo de Olavide University, Ctra. Utrera km. 1, 41013 Seville, Spain

cbarranco@upo.es

² Department of Computer Science and Artificial Intelligence, University of Granada, Daniel Saucedo Aranda s/n, 18071 Granada, Spain

{jesuscg,medina}@decsai.ugr.es

Abstract. This paper studies the influence of data distribution and clustering on the performance of currently available indexing methods, namely GT and HBPT, to solve necessity measured flexible queries on numerical imprecise data. The study of the above data scenarios lets to obtain valuable information about the expected performance of these indexes on real-world data and query sets, which are usually affected by different skew factors. Results reveal some sensibility of GT and no influence for the considered data scenarios on HBPT.

Keywords: Fuzzy databases indexing, necessity measured queries, skewed data and queries.

1 Introduction

Dealing with imprecise and uncertain (i.e. imperfect) data is, without any doubt, an important topic in computer science since decades. Fuzzy Sets Theory and Fuzzy Logic [19] are convenient tools in this regard. Besides, Fuzzy Databases [8] have proven their usefulness to manage large collections of imperfect information.

The success of the above techniques means their application in real-world environments where high performance is required. However, dealing with imperfect data means extra complexity in processing. In order to help Fuzzy Databases to achieve better performance, some indexing techniques have been proposed in the literature [5,15,6,17,10,16,12,3] (ordered chronologically). In some of these works, the performance of some of the above proposals has been tested. Particularly, restricting to the works on indexing techniques able to deal with imprecise numerical data queries using necessity measured flexible conditions, in [2] two indexing techniques, namely GT (indexing using a G-tree [11]) and HBPT (indexing using a B^+ -tree [7] combined with a Hilbert curve [12]), have been tested under different data and query scenarios focusing on the imprecise nature of

data and queries. During these tests, some evidences of sensitivity of the techniques to data distribution and clustering showed up, however these scenarios were beyond the scope of that work.

This paper focuses on the study of the above phenomenon by testing GT and HBPT indexing techniques on different scenarios of skewed data distribution and clustering. These scenarios are particularly relevant as real world data sets usually suffer from different skew degrees. Therefore, a study of these scenarios would reveal how these indexing methods would perform in real-world environments. For other non skewed scenarios, we refer the reader to previous work [2].

The paper is organized as follows. Section 2 briefly introduces some basic concepts about fuzzy databases and the indexing techniques discussed in this paper. The set of experiments designed to test data distribution and clustering influence on index performance is described in Sect. 3. Section 4 includes experiment results and discussion. Finally, Sect. 5 contains some concluding remarks and future work proposals.

2 Basic Concepts

Several interpretations of the *fuzzy database* concept can be found in the literature [8,9]. In this paper we focus on a possibilistic approach [14] where imprecise values, defined on a continuous and ordered domain, are represented as possibility distributions [18] and flexible conditions are modeled as fuzzy sets of desirable values. The matching of an imprecise value with a flexible condition is a degree obtained from a possibility or a necessity measure. In this paper we focus on indexing mechanisms designed to help solving queries using a necessity measure, particularly HBPT and GT. Other approaches designed to solve possibility measured queries [1] or to index imprecise values defined on an scalar domain [3] can be found in the literature.

A simple flexible condition is a triple $\langle A, C, T \rangle$, where A is the attribute on which the condition is applied, C is a fuzzy set of desirable values for A and T is a threshold level that sets a minimum for the matching (the necessity measure) between the value of A and C . This matching, for a data item r , is calculated as Eq. 1 shows, where $D(A)$ is the domain of the attribute A , $\pi_{A(r)}$ is the possibility function which describes the possibility distribution that models the imprecise value of the attribute A for the data item r and μ_C is the membership function of C .

$$N(C/r) = \min_{d \in D(A)} (\max((1 - \pi_{A(r)}(d)), \mu_C(d))) \quad (1)$$

In order to solve these flexible queries, HBPT and GT rely on an indexing principle proposed by Bosc et al. [4]. This indexing principle is defined as follows:

Definition 1 (Bosc et al. indexing principle for necessity measured flexible conditions). *Given a flexible condition $\langle A, C, T \rangle$, a data item r necessarily satisfies it if the expression in Eq. 2 is satisfied. In this equation, $S_T(C)$*

means for the α -cut, where $\alpha = T$, of the fuzzy set C and $S_1(A(r))$ for the core of $A(r)$ (a special α -cut where $\alpha = 1$). An α -cut of a fuzzy set $F \in \tilde{D}$, $S_\alpha(F)$, is defined as Eq. 3 shows.

$$N(C/r) \geq T \Rightarrow S_1(A(r)) \subseteq S_T(C) \tag{2}$$

$$S_\alpha(F) = \{d \in D : \mu_F(d) \geq \alpha\} \quad \forall \alpha \in (0, 1], F \in \tilde{D} \tag{3}$$

Practically, the techniques that rely on this indexing principle should provide a fast way to determine for each row whether the core of the value for the indexed attribute is a subset of the T -cut of the condition. It should be noticed that the above indexing principle is a necessary but not a sufficient condition, so the results returned by any indexing technique relying on it should be further checked to remove false positives.

When the data domain of the indexed data is ordered and continuous, as the case this paper focuses on, cores and T -cuts are intervals. Therefore, given a flexible condition $\langle A, C, T \rangle$, the indexing techniques for this kind of imprecise data should index the intervals $S_1(A(r))$, for each data item r , and allow to quickly determine which of them are contained inside the interval $S_T(C)$.

GT and HBPT index the intervals transforming the data space into a bidimensional space. Every interval $[a, b]$ is transformed into a bidimensional point (a, b) . Looking for the intervals contained inside the interval $[p, q]$ is equivalent to look for the data points inside the square region $(p, p) - (q, q)$.

The way GT and HBPT index bidimensional data points is different. GT makes use of a G-tree that splits the bidimensional data space into regions, which are mapped to disk blocks. Each time the disk block assigned to a region overflows, this region is split in two equally sized halves along one of the data space axis, which are chosen alternatively. Access to data blocks associated to the different regions is made by a directory (implemented by a B^+ -tree) using, as a key, fixed binary identifiers associated with the regions. To retrieve all data points within a given query $(p, p) - (q, q)$, it is necessary to determine which index regions (the regions in which the index divide the data space) intersect with the query region and, then, access their disk blocks using the directory. More details on GT can be found in [13,11].

HBPT indexes bidimensional points in a very different way. First, each bidimensional point is transformed into a one-dimensional counterpart by means of a Hilbert curve that induces a linear order for the points in the bidimensional space. Once the bidimensional space is transformed into a one-dimensional equivalent, the data points are indexed using an ordinary B^+ -tree. Using the index to retrieve all data points within the query region $(p, p) - (q, q)$ is an iterative process. It is necessary to perform as much one-dimensional queries on the B^+ -tree as the number of segments of the Hilbert curve that are within the query region. This requires to traverse the B^+ -tree many times, which means reading the same disk blocks (corresponding to non-leaf nodes). In order to avoid these repetitive disk block reads, the index includes a small disk block cache, whose size is the height of the B^+ -tree. A detailed description of HBPT can be found in [2].

3 Experiments

The aim of this paper is to evaluate the influence of different data clustering scenarios on the performance of the indexing techniques. In order to perform this evaluation, a group of experiments has been conducted. This section describes how we measure index performance in these experiments, the conditions in which the experiments have been run, and the detailed description of each experiment.

3.1 Performance Measurement

Index performance is measured as the saving of time, with respect to a sequential access, when a query is processed using the index. Obviously, this time saving is inversely related to the time required to apply the indexing mechanism, which can be divided into the necessary time to access index data and the time to process the index data.

The second time component, the index data processing time, is usually neglected in the index performance measurement context. As the data of a large index must be stored in secondary memory, the time needed to access this information is much greater (different magnitude orders) than the time to process it. Additionally, it is well-known that, for now, the computing power evolves faster than the secondary memory device bandwidth.

The first time component, the time required to access the data, is dependent on the amount of data accessed. However, DBMSs usually run on an operating system (OS), and its disk access mechanism must be taken into account. To increment disk performance, OSs usually include cache techniques to reduce data time access. This makes index data access time a bad candidate for an independent index performance measurement.

As secondary storage devices are block-based, the data block is their information transfer unit instead of bytes or bits. Without cache techniques the number of accessed data blocks and data access time would be proportional. This fact, makes the number of index data blocks accessed by the indexing technique an appropriate performance measurement. Any hardware or OS related techniques to reduce the access time are ignored as the amount of data (not the time taken to access it) is taken into account. Therefore, in the experiments, the performance of the indexing techniques is measured as the number of disk block accesses required to gather the necessary index data to process the queries.

3.2 Conditions of the Experiments

All the experiments described in this paper have been run following the same conditions, except for the parameters that must be changed for each experiment purposes.

Each experiment has been run on 30 randomly generated databases. The size of these databases has been fixed to 347,000 data items. This size has been chosen in order to minimize the influence of database size on the different experiments (a detailed explanation of how this value has been picked is included in the

description of the results of the experiment devoted to determine the database size influence on indexes performance).

Every data item in the database is a trapezoidal fuzzy number defined on a 8 byte double precision floating point domain in the range $[-1,000,000, 1,000,000]$. The core of each fuzzy number (which is the only relevant factor to process necessity based conditions) is randomly generated following a uniform probability distribution. This random process follows three steps. Firstly, a length for the core is randomly chosen in the range $[0, 2,000,000]$ following a uniform distribution. Secondly, a center for the core is randomly chosen in the range $[-1,000,000, 1,000,000]$ following a uniform distribution. Finally, we check if the generated core limits are within $[-1,000,000, 1,000,000]$, discarding it when they are not and returning to the first step to generate a new core.

In order to measure the index performance, we execute a set of 100 queries on each database. This means that each performance measure made for this paper is the mean value of the performance measured for 3,000 independent queries. The query conditions are randomly chosen. In fact, the 100 first elements of each database are used as the query conditions. This guarantees at least one result of each query, and a similar distribution of queries and data.

Regarding index set up, in order to avoid any index tuning, all indexes have been built from scratch by inserting, one by one, all database elements in the same order. HBPT could take advantage from index optimization building techniques for B⁺-tree (i.e. techniques to build a tree with maximum disk block usage to optimize query performance). However, none of these techniques have been applied in order to reduce the impact of this possibility, as GT can not be optimized (given its fixed division of the data space). Data pointers in the indexes are represented as 8 byte long integers. Finally disk block size is fixed to 4 kilobytes, as this is the default block/cluster size of most common file systems.

3.3 Experiments Description

In order to evaluate the impact of different clustering scenarios on the performance of the considered indexes, five experiments have been conducted.

The first one analyzes the impact of the data density of the different regions (which are mapped to data blocks) on which each indexing technique divides the data space by varying the database size. In this experiment we vary the database sizes from 10,000 up to 1,000,000, increasing the size at each iteration in 1,000 data items.

The second experiment tests the influence of the clustering of central points of the core of the indexed data elements. To do so, central points are randomly generated following a gaussian distribution whose mean value is fixed to the center of the data domain (the interval $[-1,000,000, 1,000,000]$) and its standard deviation is changed at each iteration, ranging from 10% of the domain length down to 1%. The rest of the experiment conditions are set to the standards fixed in Subject. 3.2. Figure 1 includes some sample database plots to illustrate the different clustering conditions of the experiments. Each square in the figure represents the data space, where the cores of data items are represented as

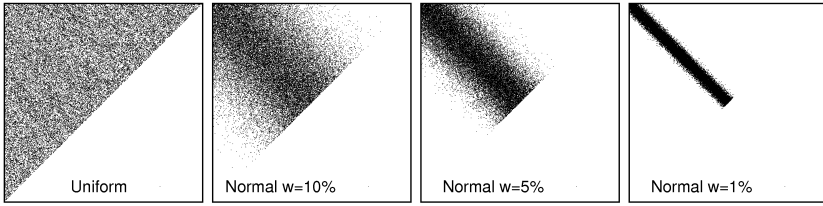


Fig. 1. Database plots for different data item core center clustering scenarios

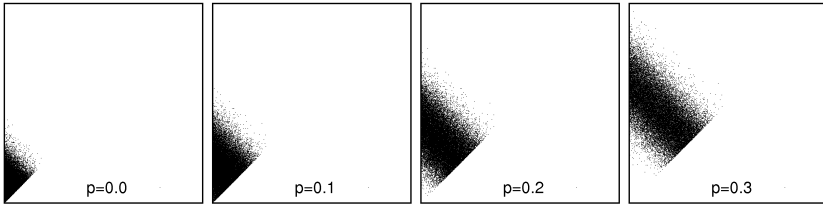


Fig. 2. Database plots for different core center distribution scenarios

bidimensional points. The most-left square represents, as a reference, a database randomly generated using the standard conditions described in Subject. 3.2. The others represent databases where the center of the cores of the data items are more or less clustered. The label of each square shows its data distribution (i.e. uniform or normal) and, when applicable, its standard deviation (parameter w) expressed as a percentage of the length of the parameter domain. It should be noted that, given that each point (a, b) represents an interval $[a, b]$ where $a \leq b$ always holds, the points in the square are always over the diagonal $x = y$.

The third experiment evaluates different scenarios of distribution of the centers of data item cores. Changes in distribution can also mean clustering. Given that the data space is not uniformly shaped (i.e. it is triangular), there is data clustering when the data items are located at the corners. In each run of the experiment, the core center of each data item has been randomly generated following a gaussian distribution whose mean value is fixed to the value $p * L$, where L is the length of the domain (in our case 2,000,000) and p is a value in $[0, 1]$. In our experiments p ranges from 0.0 to 1.0, incrementing the fixed value in 0.1 from each run to the next. The standard deviation has been fixed to 5% of the domain length in order to ensure significant changes in the distribution of data items (a wider deviation means that data items are less centered in a particular region in the data space). Figure 2 contains some sample data plots for different values of p to illustrate the different conditions considered in the experiment.

The fourth experiment is similar to the second. After an experiment on the influence of data item core center clustering, in this experiment we evaluate the impact of the clustering of the other factor defining data item cores: the length of the core. In order to do so, the length of the cores of the database items

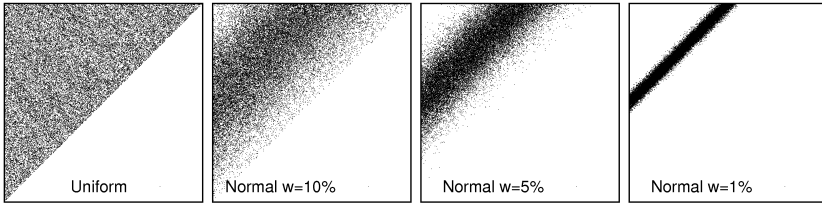


Fig. 3. Database plots for different core length clustering scenarios

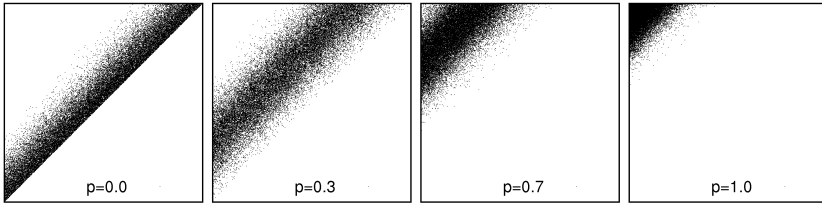


Fig. 4. Database plots for different core length distribution scenarios

are randomly generated following a gaussian distribution whose mean value is fixed to the half of the maximum length in the domain (in our case 1,000,000) and the standard deviation ranges from 10% of the domain length down to 1%. Figure 3 contains some sample data plots, including a database generated using standard conditions, to illustrate the different clustering scenarios evaluated in this experiment.

Finally, the fifth experiment is similar to the third one. It evaluates the influence of the variation of the mean core length of the data items in a database. To do so, the lengths are randomly generated following a gaussian distribution whose mean value is fixed for each run to the value $p * L$. The first run starts at $p = 0.0$ and the following runs increment the value of p in 0.1, until the value $p = 1.0$ is reached. Again, the standard deviation has been fixed to 5% of the domain length. Figure 4 contains some data plots to illustrate this scenario.

4 Results

As stated in the previous section, we are interested in testing the index performance under different data distribution scenarios. The following figures include graphical representations of the results of our experiments, where the index performance is represented in the Y axis as the number of blocks reads and the different scenarios of each analyzed factor are represented in the X axis.

4.1 Data Density

Figure 5(a) shows the results for our first experiment. It can be seen that GT method performance suffers fluctuations under different database sizes. These

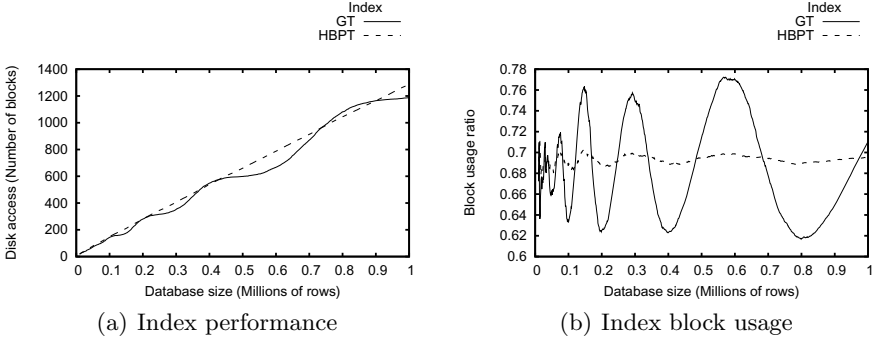


Fig. 5. Results under different data densities

fluctuations are wave shaped because of the uniform distribution of data, which results in splitting, more or less at the same time, of the regions in which the index divides the data space. These regions are mapped to data blocks, which results in fluctuations on index blocks usage (a difference between maximum and minimum peaks of around 20% of the mean block usage), as shown in Fig. 5(b). These waves become wider as the database size is bigger because the number of regions increases and, therefore, given a uniform distribution of data, it is necessary an increasing number of data items to make them overflow. Regarding HBPT, results show stability of its performance under the studied factor. The linear increment of both methods block reads is caused by the increment of results in queries.

Given the fluctuations in block usage of GT under different database sizes, the rest of the experiments are run using a database size where GT and HBPT show the combined nearest block usage ratio (sum of differences) to their mean values (0.684 for GT and 0.693 for HBPT) under this experiment. This neutralizes the performance fluctuations due to data density changes in the following experiments. However, it should be noted that the above mean values are clearly dependent of the database size ranges considered in the experiment. Therefore, the performance measures for GT in the rest of the experiments are useful only to compare both indexes stability, as actual performance strongly depends on database size. Particularly, it should be noted that results for GT in the following experiments can not be taken as absolute values, since this indexing method performance strongly fluctuates with database size variations.

4.2 Core Center Clustering

The influence of core center clustering of data items is illustrated in Fig. 6(a). GT shows performance changes on this factor, making the index performance fluctuate around HBPT results. These changes are due to variations in the index data block usage (a difference between maximum and minimum peaks of near 11% of the mean block usage), as Fig. 6(b) shows, because of data distribution.

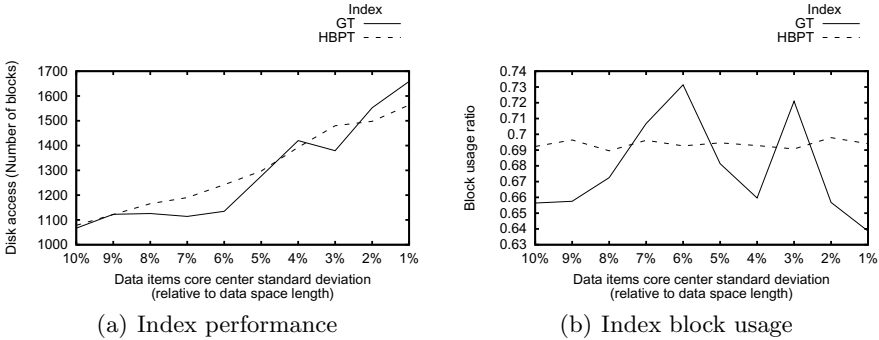


Fig. 6. Results under different core center clustering scenarios

For the conditions of the experiment (particularly for the chosen database size), the performance of GT is superior during the middle part of the graph. However, as stated in Subsect. 4.1, an absolute comparison of both indexing methods performance is not useful, since GT performance is strongly affected by database size.

Finally, the linear increment of index reads of both methods is caused by the increment of query results when data clustering is more significant.

4.3 Core Center Distribution

Figure 7(a) shows the results of our experiment to evaluate index performance under different scenarios where the mean value of the core center of the data items drifts along the domain length. Under these changes, GT suffers some fluctuations in performance. These fluctuations, once again, are due to changes on index block usage, as Fig. 7(a) shows, where valleys and peaks of block usage match with index performance fluctuations, with a difference near to 9% of the mean block usage between them. In contrast, HBPT performance behaves smoothly under these changes.

The bell-shape of the curves representing performance is caused by a decrement of query results when the data item core centers are located near the bounds of data domain. The nearer a data item core center is located of one of the bounds of the domain, the smaller the length of its core can be (notice the triangular shape of the data space) and, therefore, the smaller the query result set is when the data item is used as a query condition.

4.4 Core Length Clustering

The results of the experiment to check the influence of the clustering of data items core length are illustrated in Fig. 8(a). Results reveal that this factor has no significant influence on the indexes performance. The reduction of index data reads of both methods is due to a reduction of query results because of a smaller core length of data items. Moreover, under the particular conditions of this test

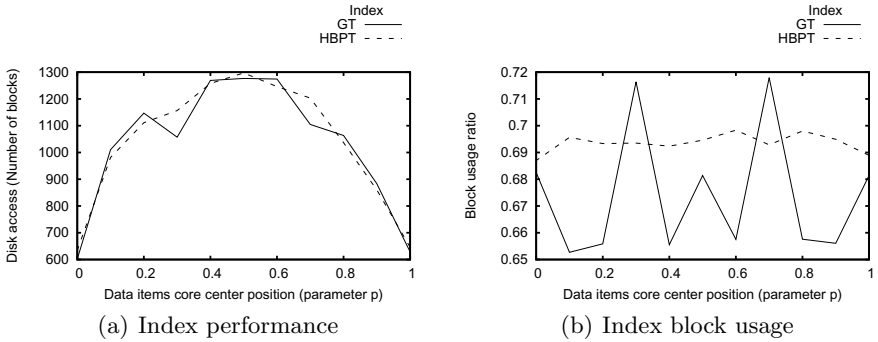


Fig. 7. Results under different core center distribution scenarios

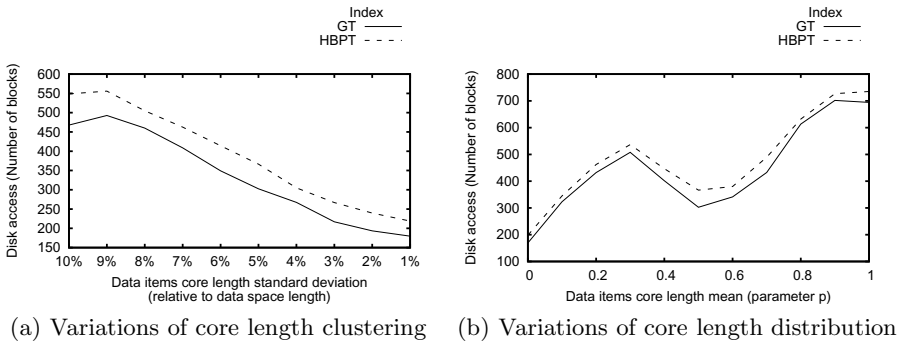


Fig. 8. Index performance under different core length scenarios

(the database size the uniform distribution of the core centers of data items) GT performance shows superior for the whole range.

4.5 Core Length Distribution

Results obtained under different scenarios with different core length mean values are depicted in Fig. 8(b). It can be noticed that none of the evaluated indexing method performance seems to be affected by this factor. The wave shape of the performance plots can be explained as a combination of factors. When the mean of the core length of the data items is near to the lower bound of the data domain (near $p = 0.0$), the number of query results is small because of the small core length, which represent strict conditions. As the mean core length of data items increases, the number of query results increases. However, when the mean is near $p = 0.25$, this trend is inverted. This change is a side effect of the reduction of data item clustering (at this level the number of possible core length values is greater than when the mean is near to the lower bound of the domain), which means less query results. Finally, when mean core length of data items approaches the higher end of the data domain, data items are, again, more

clustered. This, combined with less strict query conditions given the length of their cores, means more query results.

Like the previous experiment, results show that GT performance is over the performance measured for HBPT for the whole range. Again, the particular experiment conditions (database size and uniform distribution of the core center of data items) must be taken into account.

5 Concluding Remarks and Future Works

In this paper, the sensitivity on data distribution and clustering of two indexing techniques for solving necessity measured flexible conditions on numerical imprecise data, namely GT and HBPT, is studied.

During the experiments, some performance fluctuations have been noticed for GT. Particularly, GT performance has shown to be sensitive to data density, due to interaction between database size and disk block size. Additionally, GT performance has been affected by distribution and clustering of data item core centers. HBPT performance has not shown significant variations under the above scenarios. Regarding the experiments on different clustering and distribution of the core length of data items, none of the evaluated indexes have shown significant variations on performance due to these factors.

As results show, GT performance fluctuates around HBPT performance. Under particular data and query conditions GT performance has shown superior. These facts suggest that the usage of GT index can be appropriate for databases with a particular and stable data distribution and density (i.e. a fixed database size or core center distribution), particularly when these conditions mean high disk block usage. For other cases, where there is not such stability in database conditions, HBPT might be considered as it scales smoothly under data density and distribution changes.

Future work will focus on studying the indexes performance under data sets generated using multivariate distributions, which will help to simulate other interesting real-world data scenarios.

Acknowledgments. This work has been partially supported by the “Ministerio de Ciencia y Tecnología (MCYT)” (Spain) under grant TIN2007-68084-CO2-01, and the “Consejería de Innovación Ciencia y Empresa de Andalucía” (Spain) under research projects P06-TIC-01570 and P07-TIC-02611.

References

1. Barranco, C.D., Campaña, J.R., Medina, J.M.: A B^+ -tree based indexing technique for fuzzy numerical data. *Fuzzy Sets and Systems* 159(12), 1431–1449 (2008)
2. Barranco, C.D., Campaña, J.R., Medina, J.M.: Indexing fuzzy numerical data with a B^+ -tree for fast retrieval using necessity-measured flexible conditions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 17(supplementary issue 1), 1–23 (2009)

3. Barranco, C.D., Helmer, S.: An impact ordering approach for indexing fuzzy sets. *Fuzzy Sets and System* (in press, 2011)
4. Bosc, P., Galibourg, M.: Indexing principles for a fuzzy data base. *Information Systems* 14(6), 493–499 (1989)
5. Boss, B.: An index based on superimposed coding for a fuzzy object oriented database system. In: *Proceedings of the First International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference. The Industrial Fuzzy Control and Intelligent Systems Conference, and the NASA Joint Technolo., NAFIPS/IFIS/NASA 1994*, pp. 289–290 (1994)
6. Boss, B., Helmer, S.: Index structures for efficiently accessing fuzzy data including cost models and measurements. *Fuzzy Sets and Systems* 108(1), 11–37 (1999)
7. Comer, D.: Ubiquitous B-tree. *ACM Comput. Surv.* 11(2), 121–137 (1979)
8. Galindo, J., Urrutia, A., Piattini, M.: *Fuzzy Databases: Modeling, Design and Implementation*. Idea Group Publishing, Hershey (2006)
9. Galindo, J. (ed.): *Handbook of Research on Fuzzy Information Processing in Databases*. Information Science Reference - Imprint of: IGI Publishing, Hershey (2008)
10. Helmer, S.: Evaluating different approaches for indexing fuzzy sets. *Fuzzy Sets and Systems* 140(1), 167–182 (2003)
11. Kumar, A.: G-tree: a new data structure for organizing multidimensional data. *IEEE Transactions on Knowledge and Data Engineering* 6(2), 341–347 (1994)
12. Lawder, J., King, P.: Using space-filling curves for multi-dimensional indexing. In: Jeffery, K., Lings, B. (eds.) *BNCOD 2000*. LNCS, vol. 1832, pp. 20–35. Springer, Heidelberg (2000)
13. Liu, C., Ouksel, A., Sistla, P., Wu, J., Yu, C., Rishe, N.: Performance evaluation of g-tree and its application in fuzzy databases. In: *CIKM 1996: Proceedings of the Fifth International Conference on Information and Knowledge Management*, pp. 235–242. ACM Press, New York (1996)
14. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences* 34, 115–143 (1984)
15. Yazici, A., Cibiceli, D.: An index structure for fuzzy databases. In: *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1375–1381 (1996)
16. Yazici, A., Ince, C., Koyuncu, M.: An indexing technique for similarity-based fuzzy object-oriented data model. In: Christiansen, H., Hacid, M.S., Andreasen, T., Larsen, H. (eds.) *FQAS 2004*. LNCS (LNAI), vol. 3055, pp. 334–347. Springer, Heidelberg (2004)
17. Yazici, A., Cibiceli, D.: An access structure for similarity-based fuzzy databases. *Information Sciences* 115(1-4), 137–163 (1999)
18. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1(1), 3–28 (1978)
19. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)

Fuzzy Domains with Adaptable Semantics in an Object-Relational DBMS

José Tomás Cadenas^{1,2}, Nicolás Marín^{3,*}, and M. Amparo Vila³

¹ Department of Computation and I.T.,

Simón Bolívar University, 1080A, Caracas, Venezuela

² CAMYTD, FACYT, Carabobo University, Valencia, Venezuela

³ Intelligent Databases and Information Systems Research Group,

Department of Computer Science and A.I.,

University of Granada, 18071, Granada, Spain

jtcadenas@usb.ve, {nicm,vila}@decsai.ugr.es

<http://idbis.ugr.es>

Abstract. In this paper, we focus on the management of the user's context in the field of Fuzzy Database applications. Concretely, we propose an approach that uses fuzzy techniques to represent vague concepts with an adaptable semantics in order to meet user's preferences. We enhance previous work on the design and implementation of fuzzy databases, allowing a flexible use of different sets of linguistic labels for the same domain, which are transparently chosen according to user's context. This way, applications that operate on fuzzy databases get an additional piece of semantic power that makes an automatic customization in relation to user's preferences possible. This initial capability is proposed within a general architecture that aims at the management of the user context from a more general and interdisciplinary point of view. We complete our contribution with the description of a first proof of concept implementation of our proposal using the Oracle Object-Relational DBMS.

Keywords: Fuzzy Databases, Adaptable Semantics, User's Context, Object-Relational DBMS.

1 Introduction

Real world data are frequently imperfect (vague and/or imprecise) and this imperfection makes data management difficult in conventional database management systems (DBMS). Fuzzy Sets Theory of Zadeh [26] has proven to be a useful tool in order to manage this data imperfection as demonstrated in de Tré and Zadrozny [23] and Bosc et al. [6]. As a consequence, fuzzy database modeling and processing are important areas in database research.

In the literature, there are significant proposals using the relational data model, e.g. [12], [7], [14]. Additionally, there are also remarkable proposals for

* The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía, Consejería de Economía, Innovación y Ciencia) under project P07-TIC-03175 and Fundación Carolina.

fuzzy objects oriented models, e.g. [25], [4], [20], [21], [18], [2], [22]. In [17] the interested reader can find a recent overview of Fuzzy Conceptual Data Modeling; [11] can also be used for a compendium of research on Fuzzy Information Processing in Databases.

The extension of object-orientation in order to deal with imperfect data can also be considered in modern programming platforms [3], [1]. In recent years, fuzzy extensions have also been considered in Object-Relational DBMS like Oracle [19] and PostgreSQL [10].

Part of the success of almost all the mentioned approaches is due to the use of linguistic labels in order to represent imperfect values stored in the databases or to make the use of terms in queries that resemble the natural language used by humans possible (flexible querying). The link between the linguistic labels and the representation of imperfect values in the system is provided by the definition of the semantics according to well known results of Fuzzy Sets Theory.

Although the proposals cited above have proven to be suitable to address the representation and manipulation of the imperfection in the field of information systems, the semantics for fuzzy concepts is unique and shared by all the database users, and this fact is a clear drawback from a more human centered point of view; that is, the semantics defined to represent and manipulate imperfection is determined by the system designer. However, many users have to interact with the designed database; in this sense, it is clear that semantics largely depends on each different user or, moreover, on each user's context. Therefore, this semantics must appropriately change in order to suitably adapt to user's preferences.

For example, a user *A* inserts data related to a *big* object that can then be offered to another user *B* when asking for *medium* objects. Moreover, semantics can change even for the same person depending on his/her interest; i.e., John is 1.75 meters tall, a group of persons would say *John is medium height*, but the criteria may change if he is being selected to ride horses as a jockey or to play basketball; then, the same person would label John as *tall* in the first case and as *short* in the second situation. That is, linguistic labels are relative to the user context. In fact, it is assumed the absence of universal meaning in most of the descriptive terms used by humans for use in database systems.

In our opinion, if we want our fuzzy database systems advance from a human friendliness perspective, we will have to provide tools to represent vague concepts that have the capability of adapting its semantics to the user context and to do it in a transparent way.

Some proposals focused on this issue can be found in the literature. Though they are initial approaches to this challenging problem, some interesting results are offered. The extent of user queries with preferences deduced from fuzzy rules that describe the current user context is given by Hadjali et al. [13]; this work shows the importance of context and customization of applications in database environments. Additionally, Ughetto et al. [24] presents techniques of data summaries and customized systems, allowing users to choose their own vocabulary

when they are querying a database, using linguistic variables defined for each user's profile and getting the required level of granularity in results.

There are other relevant approaches like [28], [5], [15], [29]; but they are mainly centered on customizable fuzzy querying of crisp data in relational databases. Our proposal aims at extending the use of user context in Fuzzy Databases (not only in flexible querying) with the following advantages: the user can design and manage their applications using powerful tools (object oriented data model and ORDBMS), allowing to closely represent the real world, properly handle imperfect data, and efficiently implementing human-centered systems.

The proposal we introduce in this paper is focused on the customization of fuzzy database systems according to the needs of each user at each moment. Hence, two important contributions are the representation of fuzzy information regarding the user context; and, the provision of flexibility in database querying based on the user's preferences, including linguistic terms with adaptable semantics in human centered queries.

We focus on these issues and we provide a first progress in this direction: by means of techniques of representation and manipulation of imperfect objects based on the theory of Fuzzy Sets, we present an approach to build different fuzzy domains in a database whose semantics is automatically adapted to the user, eliminating the need to define the desired semantics at the moment of database querying.

This is an initial contribution within a more challenging goal: we want our fuzzy database management system to adaptively respond to the user *context*, with context understood in a broader sense. For this reason, our approach is integrated within an architecture for building a system with these characteristics that is also described in this paper.

The work is organized as follows: next section presents a general architecture for a fuzzy database system based on user context. Section three is focused on the representation of fuzzy domains with adaptable semantics in the framework of the proposed architecture. Section four is deserved to describe a proof of concept implementation of this capability within a conventional ORDBMS as Oracle; in this section the transparent use of our model through SQL:99 standard sentences is showed, but with adaptive semantics based on the user's preferences. Finally, section 5 concludes the paper and presents some guidelines for future work.

2 A General Architecture for a Fuzzy Database System Adaptive to the User's Context

Nowadays, people have to use Information and Communication Technologies (ICT) more effectively. To achieve this effectiveness these technologies must adapt to users as much as possible: ICT systems must be sensitive to the user's context.

Most sensitive systems are developed by Ubiquitous Computing at low level, closely tied to implicit inputs gathered by sensors, without regarding things such as desires, objectives, and the user's emotional state. These nuances related to the context are defined in a higher level of abstraction.

Moreover, uncertainty and ambiguity are present in various aspects of the user's context, such as the semantics of the used concepts, data captured by devices (sensors), or inferred through diverse methods. These facts produce a wide survey of challenges not only in the representation of information, but also in the context modeling using Fuzzy Databases and the development of context-aware systems.

To the best of our knowledge, most information systems are designed to be used mostly independently of user's context; the programmer and/or designer imposes a built-in criteria that is unable to take into account the multiple future contexts that users may have: who the user is, when and where the application works, the level of expertise of the user (beginner/expert), whether he/she is alone or in company, among others.

Usually, in all these situations, the software is designed to work in the same way; the programmers make certain assumptions that are suitable for a given pre-defined context, but may not result very appropriate in the user's current context. The problem of efficiency in retrieving information of interest for the user is even more crucial in current information systems, due to the increasing volume of information available.

A way to address this problem is developing context-aware systems that manage more knowledge about the user and his/her context. *Doing the right thing entails that it be right given the user's current context* [16].

The proposal described in this section is intended to be the base for the development of context-aware systems in environments with imperfect data managed using Fuzzy Databases.

Diverse forms of data imperfection involve the notion of graduality [2], moreover the semantics is diverse, i.e. depends on the user's context. Who can set universal limits (even fuzzy) to determine if a person is *tall*? This term may be influenced by the location of the user (Japan, Spain, or Venezuela), objectives (looking for a gymnast, a jockey, or a basketball player) or others factors that may be sensitive to the context and affect the semantics of the concepts.

In the broadest sense, a context-aware system must provide mechanisms for acquiring implicit inputs provided by the environment (either through environmental sensors or software agents inferring the context from people activity), and explicit inputs regarding the context (i.e. role and user profile). In Fig. 1 we present an architecture for a context-aware information system that is capable of manage explicit and implicit data regarding the user context. The architecture is founded on two main pillars: Ubiquitous Computing (UbiComp) in order to transparently deal with implicit information about user's context and Computational Theory of Perceptions(CTP) of Zadeh [27] in order to manage data imperfection and to make the system fit to user's requirements according to the context.

The elements of the figure are:

- **User:** it is the main beneficiary of this user-oriented architecture. In an ideal situation, the interaction of the user with the system must involve only data and commands related to information system functionalities that must

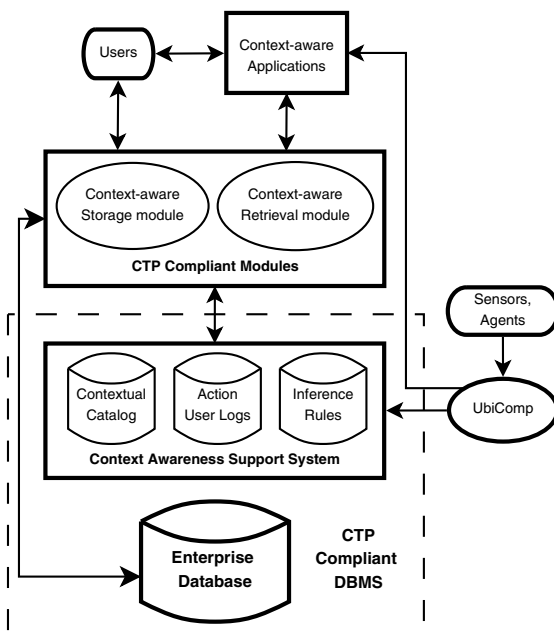


Fig. 1. Architecture for a Fuzzy Database system adaptive to user context

transparently adapt according to user’s context inferred thanks to UbiComp technologies.

- **Context-aware applications:** interaction of the user with the database is usually done through the use of applications. In our architecture, these applications must be built so that they can easily adapt to user needs according to his/her context. In some cases, these applications can also get context information about the user, when the implicit information obtained by the ubiComp system is not enough.
- **CTP Compliant Modules:** data flow from user to database must be done through storage (*CAS*) and retrieval (*CAR*) modules. These modules properly translate data from the user language to the system representation (and viceversa) according to the user’s context. CTP techniques have a main impact in this part of the system as a suitable tool for data imperfection representation and handling in natural language.
- **Context Awareness Support System:** this layer offers support for context-awareness founded on the use of context information obtained from the UbiComp system, the applications, the user, and a suitable logic to conveniently adapt system functionalities to the context. This part of the architecture is composed by:
 - **Contextual catalog:** metadata regarding the representation and management of user’s context. For example, different definitions for the linguistic terms used by each different user are stored in this catalog.

- **Action user logs:** data regarding user interaction with the system, specially useful in order to infer information about user's context.
- **Inference rules:** to express contextual data referred to user's preferences. For example, rules that determine the set of linguistic labels to be used according to the current user context.
- **Enterprise database:** conventional data storage on a DBMS. It stores and provides data from/to CTP compliant Modules.
- **CTP compliant DBMS:** It is used to implement the representation requirements of context-awareness support system and the enterprise database, providing data management functionalities and data imperfection handling.

In summary, users will be the most favored as they will interact with customized systems with adaptable semantics based on the context. The system will be able to capture perceptions of the real world as closely as possible, will be used to manage imperfect data, and will allow context-aware queries based on the users preferences with adaptable semantics; moreover, the system will have the capability of inferring and discovering contextual data; thus improving the quality of access to user's information through flexible and intelligent applications.

3 Fuzzy Domains with Adaptable Semantics

In this section, we explain how to implement a simplified version of the CTP Compliant Modules through stored procedures that hide to the user the management of an adaptable semantics in the concepts. To achieve this goal, we designed a fuzzy object oriented UML-based model of an example application (see Fig. 2). We implement a fuzzy domain according to the proposal of [9]. We extend the data types of Oracle ORDBMS and we enhance previous work to allow flexible queries, using the SQL:99 standard. The interested reader can find a detailed description about how to implement different fuzzy domains in [8].

We use an Object Relational Database Management System (ORDBMS) like Oracle due to its capability to represent complex objects, to extend the native data types, the variety of methods, and its interoperability with other programming languages besides SQL (Java, C, C ++, PL / SQL). It allows to make intuitive applications in a reasonable time, user-friendly, while preserving the inherent capabilities of a relational database (concurrent access, multiple levels of security, high performance, protecting the consistency and integrity of the data).

In this first proof of concept implementation, the content of the UML-based model is:

- **Domain:** Fuzzy Domains with adaptable semantics are given by a set of linguistic labels represented by trapezoidal membership functions for a linguistic variable such as height of a person. The system designer creates object types using the ORDBMS Oracle, accordingly to the proposed model. Then, object tables and/or object attributes (i.e. trapezoid) based on the object types are created.

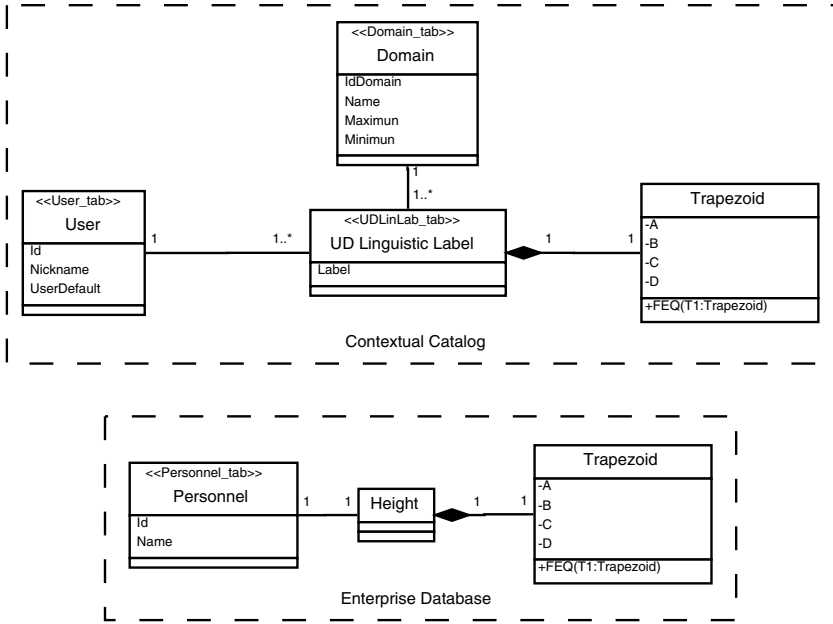


Fig. 2. Object-Oriented Fuzzy Data Model of a Context-Aware Application

- **User:** User related information. Among the users, one of them is selected as default user. The semantics provided by this user will be used in those situations where no information regarding the user is available.
- **User Domain (UD) Linguistic Label:** Both end users and database designer can define linguistic labels that are stored into the Contextual Catalog.
- **Trapezoid:** Each linguistic label is associated to a given user and a fuzzy domain, and it is described by a *Trapezoid* object. Trapezoid objects can be compared by means of a member function Fuzzy Equal (FEQ).
- **Enterprise Database:** In this naive example, it contains a Personnel Table with identifier (*Id*), *Name*, and *Height*. Each person height is described by a trapezoid of the domain Height.

Users will interact with the system through SQL standard language provided by the ORDBMS Oracle. We implement stored procedures and member functions for managing contextual data. Below, some representative sentences are showed in relation with this example.

The first step is to input data relative to this simplified version of the system. As we say, each user is able to create his/her own set of preferred linguistic labels. This information makes possible the implementation of adaptability through the *Context-Aware Storage* or *CAS* module. We present a DML statement with the corresponding *CAS* procedure (see Table II). This procedure is able to identify the user that is interacting with the database when he/she connects to Oracle database, e.g. *USER1*.

Table 1. Context Aware Storage (CAS)

CAS Insert	Description
INSERT INTO Personnel_tab VALUES(Id_seq.nextval,'Jose Tomas Cadenas', CA_Height('Low'));	Look for Low value on UDLinLab Table for current user. If defined, insert the trapezoid in Personnel_tab; if not, take the default designer's trapezoid.
INSERT INTO Personnel_tab VALUES(Id_seq.nextval,'Manuel Fernandez', CA_NEW_Height ('Average',150,175,185,200));	Create or replace the definition of Average for current user in UDLinLab table. Insert the trapezoid as value for Height in Personnel_tab

Another user may connect to the database, e.g. *USER2*, and run the following inserts.

```
INSERT INTO PERSONNEL_TAB VALUES(Id_seq.nextval, 'Luis Perez',
    CA_New_Height('Average', 165, 175, 180, 200));
INSERT INTO PERSONNEL_TAB VALUES (Id_seq.nextval,
    'Paco Martinez', CA_Height('High'));
```

Then, a new definition for label *Average* of domain *Height* according to the *USER2* is inserted into the Contextual Catalog, and the tuple regarding *Luis Perez* is inserted into personnel with the corresponding trapezoid.

The second insertion, related to *Paco Martinez* is inserted into Personnel table using the trapezoid defined by default for High, because this label has not been previously defined by *USER2* and no new semantics is provided.

Below, we show data stored into the Contextual Catalog (see Table 2) after a few number of insertions.

Table 2. Contextual Catalog Data

User	Domain	Label	Trapezoid
USER_DEF	Height	Low	(0,0,150,160)
USER_DEF	Height	Average	(150,160,170,180)
USER_DEF	Height	High	(170,180,300,300)
USER1	Height	Low	(0,0,150,160)
USER1	Height	Average	(150,175,185,200)
USER1	Height	High	(190,210,300,300)
USER2	Height	Low	(0,0,165,175)
USER2	Height	Average	(165,175,180,200)
USER2	Height	High	(180,200,300,300)

In summary, we can create logical data structures and a stored pre-defined logic to implement compliant CTP modules so that context management can be hidden to the user. The system designer can make appropriate adjustments in accordance with particular applications, e.g. he/she can incorporate diverse user's roles and profiles, generalizing the proposed idea according to figure 1.

4 Context-Aware Querying Using the ORDBMS Oracle

After populating the database, we show some queries in order to exemplify the implementation of the *Context-Aware Retrieval (CAR)* CTP compliant module.

At this moment, the database can be queried with semantics of linguistic labels linked to the current user. Below we show the DML statements that a user, e.g. *USER1*, has to execute in order to find those people that have *average* height according to *USER1* semantics.

```
connect USER1
SELECT Name_person, P.Height.FEQ('Average') FEQ
FROM Personnel_tab P
WHERE P.Height.FEQ('Average')>0.5;
```

NAME_PERSON	FEQ
-----	-----
Manuel Fernandez	1
Tomas Cadenas	.86
Macringer Montero	1
Maria Rodriguez	.56
Luis Perez	1
Paco Martinez	1

Another user, i.e. *USER2*, runs exactly the same statement from his session, obtaining a different output showed below.

```
connect USER2
NAME_PERSON
```

NAME_PERSON	FEQ
-----	-----
Manuel Fernandez	1
Tomas Cadenas	.83
Macringer Montero	1
Luis Perez	1
Paco Martinez	1

In summary, any user is able to make queries according to his/her preferences of height with an adaptable semantics.

The CTP compliant retrieval module also includes a Context Aware method (*LShow*) that allows to hide the trapezoid defined by an user and to retrieve the convex combinations of linguistic labels according to his/her own pre defined semantics.

Below we have two examples with different users

```
connect USER1
SELECT Name_person, P.Height.LShow() Height
FROM Personnel_tab P
```

NAME_PERSON	Height
Mariamni Cadenas	1/Low; .29/Average;
Tomas Cadenas	.5/Low; .86/Average;
Macringer Montero	1/Average; 1/High;
Jose Tomas Cadenas	1/Low; .29/Average;
Manuel Fernandez	.29/Low; 1/Average; .29/High;
Luis Rivera	.29/Average; 1/High;
Maria Rodriguez	.5/Low; .56/Average;
Luis Perez	.17/Low; 1/Average; .25/High;
Paco Martinez	1/Average; 1/High;

Another user, e.g. *USER2*, can run exactly the same statement from his session, obtaining a different output showed below.

```
connect USER2
```

NAME_PERSON	Height
Mariamni Cadenas	.5/Low; .17/Average;
Tomas Cadenas	.83/Low; .83/Average;
Macringer Montero	.17/Low; 1/Average; 1/High;
Jose Tomas Cadenas	.5/Low; .17/Average;
Manuel Fernandez	.56/Low; 1/Average; .57/High;
Luis Rivera	.25/Average; 1/High;
Maria Rodriguez	1/Low; .5/Average;
Luis Perez	.5/Low; 1/Average; .5/High;
Paco Martinez	.17/Low; 1/Average; 1/High;

5 Conclusions and Future Work

In this paper, we have proposed a fuzzy approach to represent vague concepts with a user-adaptable semantics in an object-relational database as a first step towards a context aware database management system. We have introduced a general architecture for a fuzzy database system based on user context and we have described how the representation of fuzzy domains with adaptable semantics can be faced in the framework of the proposed architecture.

As future work, we have to consider the development of the complete set of CTP compliant storage and retrieval modules of our architecture and the extension of this initial capability beyond a more complete context model, based not only on the user, but on other features that describes the execution context, e.g. provided by ubiquitous computing sensors (taking advantage of the action user logs and inference rules of the context awareness support system).

References

1. Berzal, F., Cubero, J.C., Marín, N., Vila, M.A., Kacprzyk, J., Zadrozny, S.: A general framework for computing with words in object-oriented programming. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 15(supplement-1), 111–131 (2007)
2. Berzal, F., Marín, N., Pons, O., Vila, M.A.: Managing fuzziness on conventional object-oriented platforms. *Int. J. Intell. Syst.* 22, 781–803 (2007)
3. Berzal, F., Marín, N., Pons, O., Vila, M.A.: Development of applications with fuzzy objects in modern programming platforms. *Int. J. Intell. Syst.* 20(11), 1117–1136 (2005)
4. Bordogna, G., Pasi, G., Lucarella, D.: A fuzzy object-oriented data model for managing vague and uncertain information. *International Journal of Intelligent Systems* 14(7), 623–651 (1999)
5. Bordogna, G., Psaila, G.: Customizable Flexible Querying in Classical Relational Databases. In: *Handbook of Research on Fuzzy Information Processing in Databases*, pp. 191–217. IGI Global (2008)
6. Bosc, P., Kraft, D.H., Petry, F.E.: Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets and Systems* 156(3), 418–426 (2005)
7. Bosc, P., Pivert, O.: SQLf query functionality on top of a regular relational DBMS. In: *Knowledge Management in Fuzzy Databases*, pp. 171–190. Springer-Verlag New York, Inc., secaucus (2000)
8. Cadenas, J.T., Marín, N., Vila, M.A.: Gestión de dominios difusos permitiendo semántica adaptable basada en el contexto del usuario utilizando un sistema de base de datos objeto relacional. Technical report, Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada (2010)
9. Cuevas, L.: Modelo Difuso de bases de datos objeto-relacional: propuesta de implementación en software libre. PhD thesis, Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada (2001)
10. Cuevas, L., Marín, N., Pons, O., Vila, M.A.: pg4db: A fuzzy object-relational system. *Fuzzy Sets Syst.* 159, 1500–1514 (2008)
11. Galindo, J. (ed.): *Handbook of Research on Fuzzy Information Processing in Databases*. IGI Global (2008)
12. Galindo, J., Medina, J., Pons, O., Cubero, J.: A server for fuzzy sql queries. In: *Andreasen, T., Christiansen, H., Larsen, H. (eds.) FQAS 1998. LNCS (LNAI), vol. 1495*, pp. 164–174. Springer, Heidelberg (1998)
13. Hadjali, A., Mokhtari, A., Pivert, O.: A fuzzy-rule-based approach to contextual preference queries. In: *Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178*, pp. 532–541. Springer, Heidelberg (2010)
14. Kacprzyk, J., Zadrozny, S.: Computing with words in intelligent database querying: standalone and internet-based applications. *Inf. Sci.* 134(1-4), 71–109 (2001)
15. Lai, L.F., Wu, C.C., Huang, L.T., Kuo, J.C.: A fuzzy query mechanism for human resource websites. In: *Deng, H., Wang, L., Wang, F., Lei, J. (eds.) AICI 2009. LNCS, vol. 5855*, pp. 579–589. Springer, Heidelberg (2009)
16. Lieberman, H., Selker, T.: Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal* 39(3&4), 617 (2000)
17. Ma, Z.M., Yan, L.: A literature overview of fuzzy conceptual data modeling. *Journal of Information Science and Engineering* 26(2), 427–441 (2010)
18. Marín, N., Medina, J.M., Pons, O., Sánchez, D., Vila, M.A.: Complex object comparison in a fuzzy context. *Information and Software Technology* 45(7), 431–444 (2003)

19. Medina, J.M., Barranco, C.D., Campaña, J.R., Jaime-Castillo, S.: Generalized fuzzy comparators for complex data in a fuzzy object-relational database management system. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. CCIS, vol. 81, pp. 126–136. Springer, Heidelberg (2010)
20. Marín, N., Pons, O., Vila Miranda, M.A.: Fuzzy types: A new concept of type for managing vague structures. *Int. J. Intell. Syst.* 15(11), 1061–1085 (2000)
21. Marín, N., Pons, O., Vila Miranda, M.A.: A strategy for adding fuzzy types to an object-oriented database system. *Int. J. Intell. Syst.* 16(7), 863–880 (2001)
22. Ozgur, N.B., Koyuncu, M., Yazici, A.: An intelligent fuzzy object-oriented database framework for video database applications. *Fuzzy Sets and Systems* 160(15), 2253–2274 (2009)
23. De Tré, G., Zadrozny, S.: The application of fuzzy logic and soft computing in information management. *Fuzzy Sets and Systems* 160(15), 2117–2119 (2009)
24. Ughetto, L., Voglozin, W.A., Mouaddib, N.: Database querying with personalized vocabulary using data summaries. *Fuzzy Sets and Systems* 159(15), 2030–2046 (2008)
25. Van Gyseghem, N., de Caluwe, R.: The UFO Model: dealing with Imperfect Information. In: *Fuzzy and Uncertain Object-Oriented Databases, Concepts and Models*, vol. 13, pp. 123–185. World Scientific Publishing, Singapore (1997)
26. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8(3), 338–353 (1965)
27. Zadeh, L.A.: A new direction in ai: Toward a computational theory of perceptions. *AI Magazine* 22(1), 73–84 (2001)
28. Zhang, W., Yu, C., Reagan, B., Nakajima, H.: Context-dependent interpretations of linguistic terms in fuzzy relational databases. In: *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 139–146 (March 1995)
29. Zheng, J., Sun, J.: Building graphical models from relational databases for context-aware querying. In: *International Conference on Information Engineering*, vol. 1, pp. 626–630 (2009)

Author Index

- Abbaci, Katia 318
Alzabdi, Mohammedsharaf 306
Andreasen, Troels 108
Atanassov, Krassimir 306
Aufaure, Marie-Aude 37
- Barranco, Carlos D. 436, 485
Belohlavek, Radim 400
Ben Yahia, Sadok 388
Billiet, Christophe 60
Blanco Medina, Ignacio 162, 424
Bordogna, Gloria 221
Bosc, Patrick 412
Bouzeghoub, Mokrane 318
Buche, Patrice 174
Bulskov, Henrik 108
- Cadenas, José Tomás 497
Campaña, Jesús R. 84, 485
Cardenosa, Jesús 119
Carlone, Domenico 234
Carman, Mark James 198
Charnomordic, Brigitte 174
Chountas, Panagiotis 306
Church, Joshua 364
Crestani, Fabio 198
- de la Rosa Esteva, Josep Lluís 186
Demir, Utku 460
Destercke, Sébastien 174
De Tré, Guy 60
- Eckhardt, Alan 258
Endres, Markus 246
- Felfernig, Alexander 13
Fischer Nilsson, Jørgen 96, 108
- Gallardo Pérez, Carolina 119
Garrido, Antonio 436
Grigori, Daniela 318
Grzegorzewski, Przemysław 342
- Hadımlı, Kerem 210
Hadjali, Allel 318, 412, 448
Horníčák, Erik 258
- Inches, Giacomo 198
Inoue, Katsumi 1
- Jaime-Castillo, Sergio 436
Jaworska, Tatiana 137
Jensen, Per Anker 108
Jose, Joemon 376
- Keskin, Sinan 72
Kießling, Werner 246
Koyuncu, Murat 460
Kuchmann-Beauger, Nicolas 37
Küçük, Dilek 128
- Lassen, Tine 108
Laurent, Anne 330
Lemos, Fernando 318
Liétard, Ludovic 318, 472
- Madsen, Bodil Nistrup 108
Marín, Nicolás 497
Marrara, Stefania 294
Masciari, Elio 270
Maslowski, Dany 25
Matthé, Tom 60
Medina, Juan M. 84, 436, 485
Motro, Amihai 364
Moulahi, Bilel 388
- Oğuztüzün, Halit 72
Ortiz-Arroyo, Daniel 234
Osička, Petr 400
- Panzeri, Emanuele 294
Pasi, Gabriella 294
Pivert, Olivier 412, 448
Poncelet, Pascal 330
Pons, Jose Enrique 60, 162, 424
Pons Capote, Olga 60, 162, 424
Psaila, Giuseppe 221
- Quintero, Malaquias 330

- Reinfrank, Florian 13
Rocacher, Daniel 318, 472

Schubert, Monika 13
Sert, Mustafa 460
Simonenko, Ekaterina 282
Špírková, Jana 354
Spyratos, Nicolas 282
Sugibuchi, Tsuyoshi 282

Tamani, Nouredine 472
Thomsen, Hanne Erdman 108
Trabelsi, Chiraz 388
Trias Mansilla, Albert 186
Turhan Yöndem, Meltem 210

Verstraete, Jörg 49
Vila, M. Amparo 84, 497
Vojtáš, Peter 258
Vychodil, Vilem 400

Whiting, Stewart 376
Wiese, Lena 1
Wijsen, Jef 25

Yazıcı, Adnan 72, 128, 149, 460
Yildirim, Yakup 149
Yilmaz, Turgay 149, 460

Zambach, Sine 108
Ziembińska, Paulina 342