Giambattista Amati
Fabio Crestani (Eds.)

LNCS 6931

Advances in Information Retrieval Theory

Third International Conference, ICTIR 2011
Bertinoro, Italy, September 2011
Proceedings

Springer

Lecture Notes in Computer Science 6931

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Giambattista Amati Fabio Crestani (Eds.)

Advances in Information Retrieval Theory

Third International Conference, ICTIR 2011
Bertinoro, Italy, September 12-14, 2011
Proceedings

Springer

Volume Editors

Giambattista Amati
Fondazione Ugo Bordoni
Viale del Policlinico 147, 00161 Rome, Italy
E-mail: gba@fub.it

Fabio Crestani
University of Lugano, Faculty of Informatics
6900 Lugano, Switzerland
E-mail: fabio.crestani@usi.ch

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-23317-3 e-ISBN 978-3-642-23318-0
DOI 10.1007/978-3-642-23318-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011934976

CR Subject Classification (1998): H.3, H.2, I.2.3, I.2.6, F.2.2, H.4, H.5.2-4, I.7

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

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

These proceedings contain the refereed papers and posters presented at the Third International Conference on the Theory of Information Retrieval (ICTIR 2011), held in Bertinoro, Italy, during September 12–14, 2011.

This biennial international conference provides an opportunity for the presentation of the latest work describing theoretical advances in the field of information retrieval (IR). The first ICTIR was held in Budapest in October 2007, organized by Keith van Rijsbergen, Sändor Dominich, Sändor Daränyi, and Ferenc Kiss. The second ICTIR was held in Cambridge, UK, in September 2009. It was organized by Leif Azzopardi, Gabriella Kazai, Stephen Robertson, Stefan Rüger, Milad Shokouhi, Dawei Song, and Emine Yilmaz.

ICTIR was brought about by the growing interest in the consecutive workshops run at ACM SIGIR each year from 2000 until 2005 on mathematical and formal methods in IR. This initiative was in large part down to the determination of Sándor Dominich and his passion for all things formal and mathematical. The foundation and the success of ICTIR is a direct result of his commitment and dedication to fostering research and development into the theoretical underpinnings of IR. Sadly, his untimely passing away in 2008 means that he is unable to witness how the theory of IR unfolds in the future. Nonetheless, his belief in the importance of theory and his spirit in advocating the development of formal methods in IR lives on through this conference series. We are glad to continue this initiative and bring ICTIR to Italy, a country with a long series of contributions to the theoretical advances of IR.

The papers accepted for publication and presentation at ICTIR 2011 were selected from a total of 65 submissions. Each submission was subject to a double-blind reviewing process by at least three Program Committee members and was ranked according to its scientific quality, originality, and contribution to the theory of IR. The Committee decided to accept 38 papers, of these 25 (38%) were accepted as full papers and 13 (20%) as short papers. Most of the submitted papers (66%) were about foundations of IR, the remaining submissions were about techniques or topics that refer to wider contexts. The Technical Program thus ranged from topics related to query expansion, co-occurrence analysis, user and interactive modelling, system performance prediction and comparison, probabilistic approaches for ranking and modelling IR to, ultimately, topics related to interdisciplinary approaches or applications.

The program also included two invited talks by ChengXiang Zhai and Keith van Rijsbergen. We would like to thank the invited speakers for their thought-provoking talks on axiomatic analysis and optimization and quantum information theory. We hope they inspired the young as well as the most senior researchers to continue in their steps.

We are grateful to the members of the Program Committee for their time and effort in providing timely and high-quality reviews and feedback to authors. We would also like to thank all the authors who submitted their work for consideration and all the participants and student volunteers for their contributions and help.

Finally, we would like to say special thanks to the following organizations and individuals who helped to make ICTIR 2011 a success:

- The Fondazione Ugo Bordoni for its support and help during the organization phases of the conference.
- The University of Lugano (USI) for providing conference website design and hosting. We especially thank Giacomo Inches and Cristian Bianchi for their work on designing the website.
- Almawave for the kind and generous sponsorship.
- Microsoft Research for their continuos sponsorship of the conference.
- Yahoo Research for sponsoring the Best Student Paper Award and the Student Travel Award.
- The editorial staff at Springer for their agreement and assistance in publishing the conference as part of the *Lecture Notes in Computer Science* (LNCS) series.
- EasyChair for the support during the reviewing stages of the submitted papers.

September 2011

Giambattista Amati
Fabio Crestani

Organization

ICTIR 2011 was jointly organized by the Fondazione Ugo Bordoni and the University of Lugano (Università della Svizzera Italiana). It was held at the the University Residential Centre of Bertinoro in Italy.

Conference and Program Chairs

Giambattista Amati	Fondazione Ugo Bordoni, Italy
Fabio Crestani	University of Lugano, Switzlerland

Sponsors

Fondazione Ugo Bordoni
University of Lugano
Almawave
Microsoft Research
Yahoo! Research

Program Committee

Maristella Agosti	Università degli Studi di Padova, Italy
Alvaro Barreiro	University of A Coruña, Spain
Marco Bianchi	Fondazione Ugo Bordoni, Italy
Roi Blanco	Yahoo! Research, Spain
Paolo Boldi	Università degli Studi di Milano, Italy
Gloria Bordogna	National Research Council, Italy
Giorgio Brajnik	Università degli Studi di Udine, Italy
Peter Bruza	Queensland University of Technology, Australia
Wray Buntine	NICTA and ANU, Australia
Fidel Cacheda	University of A Coruña, Spain
Fazli Can	Bilkent University, Turkey
Mark Carman	Monash University, Australia
Carpineto Claudio	Fondazione Ugo Bordoni, Italy
Stephane Clinchant	Xerox Research Centre Europe, France
Bruce Croft	University of Massachusetts Amherst, USA
Maarten De Rijke	University of Amsterdam, The Netherlands
Arjen De Vries	CWI, The Netherlands
Hui Fang	University of Delaware, USA
Norbert Fuhr	University of Duisburg-Essen, Germany
Giorgio Gambosi	Università di Roma Tor Vergata, Italy
Eric Gaussier	University Joseph Fourrier, France
Claudia Hauff	Delft University of Technology, The Netherlands
Ben He	Graduate University of Chinese Academy of Sciences, China
Eduard Hoenkamp	Queensland University of Technology, Australia
Jimmy Huang	York University, Canada
Qiang Huang	University of East Anglia, UK
Theo Huibers	University of Twente, The Netherlands
Giacomo Inches	University of Lugano, Switzerland
Peter Ingwersen	Royal School of Library and Information Science, Denmark
Kalervo Jarvelin	University of Tampere, Finland
Gareth Jones	Dublin City University, Ireland
Joemon Jose	University of Glasgow, UK
Donald Kraft	Louisiana State University, USA
Mounia Lalmas	Yahoo! Research, Spain
Wai Lam	The Chinese University of Hong Kong, China
Monica Landoni	University of Lugano, Switzerland

Raymond Lau	City University of Hong Kong, China
Christina Lioma	University of Stuttgart, Germany
David Losada	University of Santiago de Compostela, Spain
Robert Luk	The Hong Kong Polytechnic University, China
Craig Macdonald	University of Glasgow, UK
Andrew Macfarlane	City University London, UK
Massimo Melucci	Università degli Studi di Padova, Italy
Donald Metzler	University of Southern California, Information Sciences Institute, USA
Marie-Francine Moens	Katholieke Universiteit Leuven, Belgium
Boughanem Mohand	IRIT University Paul Sabatier Toulouse, France
Alessandro Moschitti	Università degli Studi di Trento, Italy
Jian-Yun Nie	Université de Montreal, Canada
Paul Ogilvie	LinkedIn, USA
Iadh Ounis	University of Glasgow, UK
Gabriella Pasi	Università degli Studi di Milano Bicocca, Italy
Vassilis Plachouras	Presans, France
Vijay Raghavan	Univertity of Louisiana at Lafayette, USA
Andreas Rauber	Vienna University of Technology, Austria
Stephen Robertson	Microsoft Research Cambridge, UK
Thomas Roelleke	Queen Mary University of London, UK
Stefan Ruger	Knowledge Media Institute, UK
Ian Ruthven	University of Strathclyde, UK
Jacques Savoy	University of Neuchatel, Switzerland
Giovanni Semeraro	Università degli Studi di Bari, Italy
Fabrizio Silvestri	National Research Council, Italy
Amanda Spink	Loughborough University, UK
Paul Thomas	CSIRO, Australia
Anastasios Tombros	Queen Mary University of London, UK
Theo Van Der Weide	Radboud University Nijmegen, The Netherlands
Keith Van Rijsbergen	University of Glasgow, UK
Olga Vechtomova	University of Waterloo, Canada
Emine Yilmaz	Microsoft Research Cambridge, UK
Chengxiang Zhai	University of Illinois, Urbana-Champaign, USA
Dell Zhang	Birkbeck College University of London, UK
Peng Zhang	Robert Gordon University, UK
Jianhan Zhu	University College London, UK
Guido Zuccon	University of Glasgow, UK

Additional Reviewers

Bonzanini, Marco
Calegari, Silvia
Caputo, Annalina
Daoud, Mariam
De Vine, Lance
Koopman, Bevan
Kopliku, Kopliku
Martinez-Alvarez, Miguel
Naji, Nada
Ozcan, Rifat
Parapar, Javier
Santos, Rodrygo
Yahyaei, Sirvan
Zhao, Jessie

Table of Contents

Comparison of Information Retrieval Systems and Approximate Search

Probability Ranking Principle and Alternatives

Interdisciplinary Approaches

User and Relevance

Result Diversification and Query Disambiguation

Logical Operators and Descriptive Approaches

Posters

Axiomatic Analysis and Optimization of Information Retrieval Models

ChengXiang Zhai

University of Illinois at Urbana-Champaign
czhai@cs.uiuc.edu

Abstract. Development of optimal retrieval models is an important, yet challenging research problem in information retrieval. Although many effective retrieval models have been proposed, there is still no clear single winner, making it interesting to ask the question whether there exists a single optimal retrieval model that is better than all others. However, this question is theoretically ill defined unless we can formally characterize what properties must be satisfied by an optimal retrieval model. In this talk, I will present a number of formal constraints that an optimal retrieval model are expected to satisfy, and show that these constraints not only provide a formal framework for analytically assessing the optimality of a retrieval model, but also are necessary for diagnosing deficiencies of existing models and improving them. I will use several examples to show that such an axiomatic analysis is required in order to better understand and bridge the gap between theoretically motivated models and empirically effective retrieval functions. Finally, I will discuss some interesting challenges in developing a complete axiomatic analysis framework for seeking an ultimately optimal retrieval model.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, p. 1, 2011.

What Is Quantum Information Retrieval?

C.J. Keith van Rijsbergen

keith@dcs.gla.ac.uk

Abstract. I will introduce the theoretical foundations for quantum information retrieval derived from Quantum Theory.

There will be an explanation of how such a theoretical framework could be useful in IR, for example, by showing how logic, probability, and geometry, as exploited in IR, can be represented in a consistent way in the underlying Hilbert Space.

The talk will conclude with some examples of recent concrete applications of the framework in IR.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, p. 2, 2011.

User Perspectives on Query Difficulty

Christina Lioma[1], Birger Larsen[2], and Hinrich Schutze[1]

[1] Informatics, Stuttgart University, Stuttgart, Germany
[2] Royal School of Library and Information Science, Copenhagen, Denmark
liomaca@ims.uni-stuttgart.de, blar@iva.dk, hs999@ifnlp.org

Abstract. The difficulty of a user query can affect the performance of Information Retrieval (IR) systems. What makes a query difficult and how one may predict this is an active research area, focusing mainly on factors relating to the retrieval algorithm, to the properties of the retrieval data, or to statistical and linguistic features of the queries that may render them difficult. This work addresses query difficulty from a different angle, namely the users' own perspectives on query difficulty. Two research questions are asked: (1) Are users aware that the query they submit to an IR system may be difficult for the system to address? (2) Are users aware of specific features in their query (e.g., domain-specificity, vagueness) that may render their query difficult for an IR system to address? A study of 420 queries from a Web search engine query log that are pre-categorised as `easy, medium, hard` by TREC based on system performance, reveals an interesting finding: users do not seem to reliably assess which query might be difficult; however, their assessments of which query features might render queries difficult are notably more accurate. Following this, a formal approach is presented for synthesising the user-assessed causes of query difficulty through opinion fusion into an overall assessment of query difficulty. The resulting assessments of query difficulty are found to agree notably more to the TREC categories than the direct user assessments.

Keywords: query difficulty, crowdsourcing, subjective logic.

1 Introduction

Information Retrieval (IR) systems aim to retrieve relevant information from a usually large and heterogeneous data repository such as the Web, in response to a user query. Whereas most IR systems aim to employ globally optimal algorithms that can reliably retrieve documents for most queries, there exist some particularly hard queries for which IR systems tend to underperform. Identifying this type of hard queries is important because it allows IR systems to address them in improved ways, for instance by suggesting automatically alternative or additional search terms to the users so that they can reformulate their queries, by expanding the retrieval collection of documents to better answer poorly covered queries, or by training models that can predict further difficult queries [2].

Identifying query difficulty has received a lot of attention (overviewed in Section 2), mainly focusing on factors relating to the system or algorithms used for

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 3–14, 2011.

retrieval, to the properties of the data to be retrieved, or to statistical and/or linguistic features of the queries that make them difficult. This work addresses query difficulty from a different angle, namely the user's own perspectives on query difficulty. Specifically, the research questions investigated are:

1. Are users aware that the query they submit to an IR system may be difficult for the system to address?
2. Are users aware of specific features in their query (e.g., domain-specificity, vagueness) that may render their query difficult for an IR system to address?

The motivation for studying user perspectives on query difficulty partly stems from the fact that increasingly more users regularly use Web IR systems for professional, personal, administrative and further reasons, hence they acquire experience in using search engines. This study investigates whether this search experience can allow users to estimate system-based query difficulty. In addition, the way in which users perceive query difficulty is an interesting question, especially if the users' perspections are found to divert from the system-based understanding of query difficulty, because it can be used constructively in several areas: for instance, when designing user-system interaction functionalities, such as selective user feedback, or when interpreting logged user search sessions and using them to create or train models that involve the user in the search process.

Motivated by the above, this work presents a study using 420 queries from the 2009 TREC Million query track [4], which have already been classified as `easy, medium, hard` by the track's organisers, based on the participating systems performance. A total of 370 anonymous experienced Web search users were recruited through crowdsourcing and asked for their perspectives on the difficulty of these 420 queries. Specifically, users were asked to assess how difficult each query may be for a search engine, without inspecting retrieval results, but simply according to their personal experience and subjective assessment. Furthermore, users were asked to assess, again based on their personal experience and without inspecting retrieval results, whether any of the following causes may render the query difficult for a search engine: the query being too vague, too short, too ambiguous, domain-specific, too specific, or containg typographic errors. Two findings emerge. Firstly, the user-based assessments of query difficulty disagree strongly with the TREC categorisation. Considering the TREC categories as ground truth indicates that users tend to largely underestimate the difficulty of a query for a search engine. Secondly, the user assessments of the causes that may render a query difficult for a search engine are notably more accurate than their overall assessments of query difficulty. In other words, even though users do not seem to reliably assess which query might be difficult, they can assess more reliably which query features might render the query difficult. Following this observation, a formal approach is presented for synthesising the user-assessed causes of query difficulty into an overall assessment of query difficulty. Using probabilistic logic from the subjective logic framework, the individual user-assessed causes of query difficulty are represented as formal beliefs of query difficulty, which are then fused to produce an expectation that

the query is overall difficult. The resulting assessments of query difficulty are found to agree notably more to the TREC categories than the user assessments.

This work contributes an alternative insight into how users perceive query difficulty, which has not been studied before to the best of our knowledge. A formal combination of user perspectives about the causes of query difficulty is presented and juxtaposed to system-based assessments of query difficulty.

The remainder of this paper is organised as follows: Section 2 overviews related work on query difficulty. Section 3 presents the adopted methodology for crowdsourcing user perspectives on query difficulty, and their comparison against TREC categories of query difficulty. Section 4 formalises the user perspectives to induce a probabilistic expectation of query difficulty, which is evaluated against the TREC categories of query difficulty. Section 5 summarises this work and suggests future research directions.

2 Related Work

The study of query difficulty is an active research area in IR, with several applications, such as improving the system's interaction with their users through recommending better terms for query refinement when faced with hard queries [10], providing users with an estimation on the expected quality of results retrieved for their queries, so that they can optionally rephrase difficult queries or resubmit them to alternative search resources, or selectively employing alternative retrieval strategies for particularly hard queries which might be too computationally costly if applied to all queries [2].

Studies of query difficulty can be generally separated into pre-retrieval and post-retrieval approaches (useful overviews are provided in [2,6]). Pre-retrieval approaches focus on features of the query that may render it difficult prior to retrieval, for instance naive features such as query length [14], or indexed statistics of the query terms (e.g., occurrence distribution over the documents in the collection [7], or query term co-occurrence statistics [16]). Further query features include linguistic aspects that may point to difficult queries (e.g. morpheme count per term, count of conjunctions/proper nouns/acronyms/numeral values/unknown words per query, syntactic depth, or polysemy value [11,12]).

Post-retrieval approaches focus on the observed retrieval performance to measure the coherence or clarity of the retrieved documents and their separability from the whole collection of documents [5], or the robustness of the set of retrieved documents under different types of pertrubations [15], or the retrieval status value distribution of the retrieved documents. Furthermore, there exist approaches that combine both pre-retrieval and post-retrieval aspects, for instance the model of Carmel et al., which posits that query difficulty strongly depends on the distances between the textual expression of the query, the set of documents relevant to the query, and the entire collection of documents [3].

Overall, the consensus seems to be that pre-retrieval approaches to query difficulty are inferior to post-retrieval approaches (particularly so when using linguistic features [12]). A reason for this may be that most queries are very

short and hence very poor in features that could potentially discriminate reliably between hard and easy queries. However, pre-retrieval approaches are not as computationally costly as post-retrieval methods, because they do not require dynamic computation at search time.

This work can be seen as a pre-retrieval approach. Its departure from other pre-retrieval approaches is that it does not aim to propose a new improved feature for identifying query difficulty; instead, the aim is to study whether and to what extent users perceive query difficulty. Hence, this work does not use automatic processing to derive features of query difficulty; instead, a large sample of users are asked directly for their opinions regarding whether a query is difficult and which causes might render it difficult. The resulting user perspectives can be potentially useful, both on a theoretical level, for instance to better understand the user's cognitive process during information seeking, and also on a practical level, for instance to improve user-system interaction design functionalities.

3 Crowdsourcing User Perspectives

The query set used in this work consists of the 420 queries categorised as `easy, medium, hard` by the 2009 TREC Million Query track [4] organisers, according to the average precision performance of the participating approaches. The distribution of query difficulty in this TREC categorisation is: 29.8% `easy`, 32.1% `medium`, 38.1% `hard` (see Figure 1(a) for the raw counts). These queries have been drawn from a large Web search engine log, without any manual refinement or error correction apart from case collapsing, as described in [1]. For the purposes of this study, user perspectives on the difficulty of these queries were obtained using the Amazon Mechanical Turk (AMT[1]) crowdsourcing platform. AMT is increasingly used to capture and study user preferences or insights into various facets of IR, such as evaluation measures [13]. In this study, 370 experienced Web search engine users were engaged through AMT to:

1. assess the difficulty of a query for a Web search engine, without inspecting retrieval results, but solely according to their personal experience and subjective assessments;
2. assess whether the difficulty of a query may be due to the causes shown in Figure 1(b) or to any other cause that they specify.

The assessments of query difficulty were given in the scale: `easy, medium, hard`, so that they could be directly comparable to the TREC categories. The user assessments of the individual causes that may render a query difficult were binary: `yes,no`. Each query was assessed by 5 users (who had at least $\geq 95\%$ AMT approval rate), resulting in a total of 2100 assessments. The final decision on each query was the most popular among its 5 assessments; in case of draw, another user assessed the query again. Regarding the user statistics, the average user was 31.7 years old and searched the Web 24.2 days per month on average. 51.5% of the users were native English speakers.

[1] https://www.mturk.com

Even though the users were asked to assess query difficulty without inspecting retrieval results, there is no guarantee that they did not do so. A pointer to this direction may be the time they spent on each assessment, which was overall quite low (69.5 seconds on average), leaving little time for inspecting retrieval results.

Finally, an explicit assumption of this study is that query difficulty can be perceived by a user for a query that is not his or her own. For 80.10% of the assessed queries, the participating users explicitly stated that they understood the queries they assessed. Even though understanding a query is not synonymous to cognitively formulating an information need and expressing it as a query, this study uses the former to approximate the latter.

Figure 1(a) shows the categories of query difficulty according to TREC (system-based) versus AMT (user-based). It emerges that users assessed as `easy` more than double the queries categorised as `easy` according to TREC. Furthermore, users assessed as `hard` almost one quarter of the queries categorised as `hard` by TREC. The % of agreement between AMT and TREC is overall low (approx. 34%) and particularly low for hard queries (5%). If the TREC categories are accepted as ground truth, Figure 1(a) seems to indicate that users cannot reliably assess query difficulty, and specifically that they tend to grossly underestimate query difficulty.

Figure 1(b) shows the number (#) and % of queries for which the users identified the causes listed in column 1 as reasons for query difficulty. The three most common causes, sorted decreasingly by frequency, are the query being too vague, too short, and ambiguous. Despite identifying these causes of query difficulty in a query, users did not necessarily assess that query as difficult. This can be seen in Table 1 by comparing the distribution of the queries identified as too vague, too short and ambiguous in the TREC versus AMT categories: the number of vague/short/ambiguous queries increases steadily as one observes the `easy` versus `medium` versus `hard` queries categorised by TREC; however, this is not the case for the AMT assessments, where the number of vague/short/ambiguous queries is the smallest for the `hard` queries, compared to `medium` and `easy` queries. This observation also holds for the other causes of query difficulty. This may be due to the users' poor perception of the (well-known in IR) approximately inverse relation between term occurrence and term discriminativess [8]; users may be more likely to consider easy a term that they are very familiar with through frequent use, than a more discriminative term, and this may affect their estimation about the difficulty of the query containing the term.

The last three columns of Table 1 show the distribution of queries according to the causes of query difficulty only for the subset of queries where TREC and AMT agree. Query vagueness, short length and ambiguity are also the most common causes of difficulty for this subset of queries.

The above observations seem to point to the following paradox: assuming TREC categories as ground truth, user assessments of query difficulty are not accurate; however, user assessments of individual causes that may render queries difficult are not necessarily inaccurate. This begs the question: can the causes of

		AMT							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**84**	**20.0**	33	7.9	8	1.9	125	29.8
	medium	84	20.0	**38**	**9.1**	13	3.1	135	32.1
	hard	98	23.3	41	9.8	**21**	**5.0**	160	38.1
	∑	266	63.3	112	26.7	42	10.0	420	100

(a)

Cause	#	%
too vague	107	25.5
too short	84	20.0
ambiguous	69	16.4
domain-specific	45	10.7
has typos	27	6.4
too specific	23	5.5
none	65	15.5

(b)

Fig. 1. (a): Query difficulty according to AMT assessments (based on user perspectives) & TREC categories (based on system performance). Bold font indicates agreement. # indicates number of queries. (b): Reasons for query difficulty based on user perspectives.

Table 1. Causes of query difficulty for different query groups according to TREC (based on system performance), AMT (based on user perspectives), and the agreement between TREC and AMT

	TREC			AMT			TREC & AMT		
cause	easy	medium	hard	easy	medium	hard	easy	medium	hard
	#	#	#	#	#	#	#	#	#
too vague	20	31	56	39	45	23	5	12	13
too short	21	25	38	30	32	22	9	9	11
ambiguous	16	25	28	21	29	19	4	9	7
domain-specific	10	14	21	18	19	8	4	5	5
has typos	5	7	15	9	10	8	3	2	4
too specific	7	6	10	13	7	3	4	1	0

query difficulty identified by the users be accurately synthesised into an overall estimation of query difficulty? The next section addresses this question.

4 Query Difficulty Estimation as Opinion Fusion

This section presents (i) how the subjective perceptions of the users about causes of query difficulty can be formally represented as subjective beliefs (section 4.1); (ii) how the resulting formal beliefs can be fused to give an overall estimation of query difficulty (section 4.2); and (iii) how the resulting formally derived estimation of query difficulty compares to the TREC system-based categorisation of query difficulty (section 4.3).

4.1 Turning User Perspectives into Formal Opinions

Each assessment of the AMT users described in section 3 can be considered as a subjective belief of the user. Using the formalism of subjective logic [9], a frame of discernment can be defined over the proposition that the query is difficult, following [11]. Under this analogy, each of the causes of query difficulty listed in Figure 1(b) can be represented as a different observer holding an opinion about

the truth of the proposition that the query is difficult. Subjective logic considers an observer's opinion as decomposable into degrees of belief, uncertainty, and an *a priori* probability in the absence of committed belief mass. These components can be computed directly from the AMT user assessments, using the subjective logic bijective mapping between formal opinion components and observed evidence, defined for binary events [9]. Specifically, the observed evidence can be represented by the `yes, no` assessments of the AMT users described in section 3, denoted Y, N. The belief b and uncertainty u of an opinion can then be estimated as: $b = \frac{Y}{Y+N+2}$ and $u = \frac{2}{Y+N+2}$ (see [9] for a full derivation and explanation of these equations). Hence, the user-assessed causes of query difficulty can be mapped into formal subjective opinions about the query difficulty.

4.2 Fusing Opinions of Query Difficulty Using Bayesian Consensus

The next step consists in combining the resulting subjective opinions to estimate an overall expectation that the query is difficult. One way of combining these opinions is to assume that they have been formulated independently of each other, that their combination should be commutative, associative and unbiased, and that the uncertainty of at least one of the combined opinions is not zero (because if all opinions have zero uncertainty, they are dogmatic, hence there is no basis for their consensus). Indeed, in this work, the uncertainty of each opinion is uniform and never zero ($u = \frac{2}{7}$ because each query is always assessed by 5 assessors). Then, assumming that A and B represent two different causes of query difficulty, the Bayesian consensus of observers A and B is denoted $\omega^{A,B} = \omega^A \oplus \omega^b$, and its components can be estimated as follows [9]:

$$b^{A,B} = \frac{b^A u^B + b^B u^A}{\kappa} \tag{1}$$

$$u^{A,B} = \frac{u^A u^B}{\kappa} \tag{2}$$

$$a^{A,B} = \frac{a^B u^A + a^A u^B - (a^A + a^B) u^A u^B}{u^A + u^B - 2u^A u^B} \tag{3}$$

where b, d, a denote respectively belief, disbelief, and the a priori probability in the absence of assigned belief mass, and where $\kappa = u^A + u^B - u^A u^B$ ($\kappa \neq 0$). In this work, the *a priori* probability has been set to $a = 0.5$ following [11], so that it is split equally between the two possible states of the frame of discernment, namely that the query either is or is not difficult. The final expectation in the truth of the proposition that the query is difficult is given by:

$$E^{A,B} = b^{A,B} + a^{A,B} u^{A,B} \tag{4}$$

The estimation of query difficulty resulting from Equation 4 is a probability. In order to compare this estimation to the TREC categories of query difficulty, the subjective logic probability needs to be mapped to the `easy, medium, hard` classes of query difficulty. This is done by sorting increasingly all the estimations produced by Equation 4 for all the combinations of causes of query difficulty used

in this work, and then binning them into three equal-sized bins. The first, second and third bin respectively contain the lowest, medium, and highest estimations, which are mapped to the `easy, medium` and `hard` classes respectively.

For brevity, combinations of two observers only, which represent pairs of user-assessed causes of query difficulty, are presented in this work. The next section discusses their resulting assessments of query difficulty against the backdrop of the system-based TREC categories.

4.3 Bayesian Consensus Assessments versus TREC Categories

By representing each pair of the six causes of query difficulty listed in Figure 1(b) as observers A and B in Equation 4, 15 Bayesian consensus combinations of pairs of user-assessed causes of query difficulty emerge. Figures 2(a)-3(g) display the categories of query difficulty according to TREC (system-based) versus the assessments of the pairs of causes of query difficulty identified by the AMT users and combined by Bayesian consensus as discussed above. The first row displays the causes of query difficulty that are being combined. The last column is the same for all combinations because it shows the distribution of query difficulty according to TREC (i.e. the ground truth).

Averaging the number of queries assessed `easy` and `hard` for all 15 combinations shown in Figures 2(a)-3(g) reveals that 107.5 queries are now assessed as `easy` using the combinations of causes; this is a notable drop from the direct user assessments which classed 266 queries as `easy` (see Figure 1(a)), and much closer to the number of queries categorised as `easy` by TREC (namely 125). Hence, on average, the subjective logic combinations of user perspectives of query difficulty do not seem to overestimate the number of `easy` queries, like the users themselves did. Furthermore, the average number of queries assessed as `hard` for all 15 combinations is 51.3; this is an increase from the 41 queries that the users directly assessed as `hard`, however it is still much lower than the 160 queries categorised as `hard` by TREC. This indicates that identifying difficult queries is a much harder task than identifying easy queries, when using the combinations of user-assessed perspectives of query difficulty.

The individual combinations of causes of query difficulty are displayed in Figures 2(a)-3(g). Regarding the differences between the individual combinations of causes of query difficulty, Figure 3(h) summarises the number and proportion of queries correctly assessed as `hard` by each of these combinations, using the 160 queries categorised `hard` by TREC as a baseline (see Table 1(a)). The best combination seems to be the user's perception that a query is too short and too vague, which correctly identifies 34.37% of hard queries. Note that the users' direct assessments of query difficulty identified correctly only 13.1% of hard queries. Among the less reliable combinations of query difficulty causes are those involving the query having typographical errors, being too specific, and being domain-specific. These three causes are also the least frequent in the query set (see Figure 1(b)), being found respectively in only 6.4%, 5.5%, and 10.7% of all queries, which might affect the overall reliability of their combined assessment to a certain extent.

		Ambiguous ⊕ Domain						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**38**	**9.1**	69	16.4	18	4.3	125	29.8
	medium	41	9.8	**74**	**17.6**	20	4.8	135	32.1
	hard	39	9.3	92	21.9	**29**	**6.9**	160	38.1
	∑	118	28.1	235	56.0	67	16.0	420	100

(a)

		Ambiguous ⊕ Short						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**38**	**9.1**	69	16.4	18	4.3	125	29.8
	medium	36	8.8	**72**	**17.1**	27	6.4	135	32.1
	hard	26	6.2	95	22.6	**39**	**9.3**	160	38.1
	∑	100	2.4	236	56.2	84	20.0	420	100

(b)

		Ambiguous ⊕ Specific						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**32**	**7.6**	83	19.8	10	2.4	125	29.8
	medium	37	8.8	**83**	**19.8**	15	3.6	135	32.1
	hard	26	6.2	117	27.9	**17**	**4.0**	160	38.1
	∑	95	22.6	283	67.4	42	10.0	420	100

(c)

		Ambiguous ⊕ Typos						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**57**	**13.6**	59	14.0	9	2.1	125	29.8
	medium	59	14.0	**63**	**15.0**	13	3.1	135	32.1
	hard	49	11.7	92	21.9	**19**	**4.5**	160	38.1
	∑	165	39.3	214	51.0	41	9.8	420	100

(d)

		Ambiguous ⊕ Vague						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**36**	**8.6**	67	16.0	22	5.2	125	29.8
	medium	33	7.9	**69**	**16.4**	33	7.9	135	32.1
	hard	23	5.5	88	21.0	**49**	**11.7**	160	38.1
	∑	92	21.9	224	53.3	104	24.8	420	100

(e)

		Short ⊕ Specific						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**30**	**7.1**	82	19.5	13	3.1	125	29.8
	medium	29	6.9	**91**	**21.7**	15	3.6	135	32.1
	hard	32	7.6	100	24.4	**28**	**6.7**	160	38.1
	∑	91	21.7	273	65.0	56	13.3	420	100

(f)

		Short ⊕ Domain						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**40**	**9.5**	72	17.1	13	3.1	125	29.8
	medium	33	7.9	**85**	**20.2**	17	4.0	135	32.1
	hard	37	8.8	91	21.7	**32**	**7.6**	160	38.1
	∑	110	26.2	248	59.0	62	14.8	420	100

(g)

		Specific ⊕ Domain						∑	
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**36**	**8.6**	79	18.8	10	2.4	125	29.8
	medium	43	10.2	**83**	**19.8**	9	2.1	135	32.1
	hard	44	10.5	100	24.4	**16**	**3.8**	160	38.1
	∑	123	29.3	262	62.4	35	8.3	420	100

(h)

Fig. 2. Query difficulty according to TREC categories (based on system performance) versus query difficulty according to subjective logic predictions based on the following combinations of causes of query difficulty (identified by AMT users): (a): ambiguous & domain-specific; (b): ambiguous & too short; (c): ambiguous & too specific; (d): ambiguous & has typos; (e): ambiguous & too vague; (f): too short & too specific; (g): too short & domain-specific; (h): too specific & domain-specific.

		Typos ⊕ Domain							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**57**	**13.6**	64	15.2	4	0.9	125	29.8
	medium	61	14.5	**68**	**16.2**	6	1.4	135	32.1
	hard	69	16.4	77	18.3	**14**	**3.3**	160	38.1
	∑	187	44.5	209	49.8	24	5.7	420	100

(a)

		Vague ⊕ Domain							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**35**	**8.3**	72	17.1	18	4.3	125	29.8
	medium	33	7.9	**80**	**19.0**	22	5.2	135	32.1
	hard	23	5.5	96	23.0	**41**	**9.8**	160	38.1
	∑	91	21.7	248	59.0	81	19.3	420	100

(b)

		Vague ⊕ Specific							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**25**	**6.0**	84	20.0	16	3.8	125	29.8
	medium	23	5.5	**96**	**23.0**	16	3.8	135	32.1
	hard	14	3.3	120	28.6	**26**	**6.2**	160	38.1
	∑	62	14.8	300	71.4	58	13.8	420	100

(c)

		Vague ⊕ Typos							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**47**	**11.2**	69	16.4	9	2.1	125	29.8
	medium	51	12.1	**70**	**16.7**	14	3.3	135	32.1
	hard	28	6.7	107	25.5	**25**	**6.0**	160	38.1
	∑	126	30.0	246	58.6	48	11.4	420	100

(d)

		Short ⊕ Typos							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**63**	**15.0**	54	12.9	8	1.9	125	29.8
	medium	57	13.6	**63**	**15.0**	15	3.6	135	32.1
	hard	48	11.4	93	23.0	**19**	**4.5**	160	38.1
	∑	168	40.0	210	50.0	42	10.0	420	100

(e)

		Short ⊕ Vague							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**37**	**8.8**	69	16.4	19	4.5	125	29.8
	medium	27	6.4	**73**	**17.4**	35	8.3	135	32.1
	hard	21	5.0	84	20.0	**55**	**13.1**	160	38.1
	∑	85	20.2	226	53.8	109	26.0	420	100

(f)

		Specific ⊕ Typos							
		easy		medium		hard		∑	
		#	%	#	%	#	%	#	%
TREC	easy	**54**	**12.9**	67	16.0	4	0.9	125	29.8
	medium	69	16.4	**61**	**14.5**	5	1.2	135	32.1
	hard	63	15.0	87	20.7	**10**	**2.4**	160	38.1
	∑	186	44.3	215	51.2	19	11.5	420	100

(g)

Queries correctly assessed as hard		
assessment type	#	%
1. user direct assessment	21	13.1
2. formal combinations of causes:		
-too specific ⊕ has typos	10	6.25
-has typos ⊕ domain-specific	14	8.75
-too specific ⊕ domain-specific	16	10.00
-ambiguous ⊕ too specific	17	10.62
-ambiguous ⊕ has typos	19	11.87
-too short ⊕ has typos	19	11.87
-too vague ⊕ has typos	25	15.62
-too vague ⊕ too specific	26	16.25
-too short ⊕ too specific	28	17.50
-ambiguous ⊕ domain-specific	29	18.12
-too short ⊕ domain-specific	32	20.00
-ambiguous ⊕ too short	39	24.37
-too vague ⊕ domain-specific	41	25.62
-ambiguous ⊕ too vague	49	30.62
-too short ⊕ too vague	55	34.37

(h)

Fig. 3. Query difficulty according to TREC categories (based on system performance) versus query difficulty according to subjective logic predictions based on the following combinations of causes of query difficulty (identified by AMT users): (a): has typos & domain-specific; (b): too vague & domain-specific; (c): too vague & too specific; (d): too vague & has typos; (e): too short & has typos; (f): too short & too vague; (g): too specific & has typos. Table (h) displays the number and proportion of queries that have been assessed correctly as hard (using the 160 queries classed hard by TREC as ground truth), firstly by the users when asked directly, and secondly by formally combining the causes of query difficulty perceived by users.

5 Conclusion

This work investigated the users' perceptions of whether a query may be difficult for an IR system to process, and for which causes. 370 anonymised Web search users were recruited using the Amazon Mechanical Turk crowdsourcing platform, and asked to assess the difficulty of 420 Web search queries without inspecting the results retrieved for these queries, but solely according to their subjective opinions and personal experience with search engines. The queries were previously classed as `easy, medium, hard` by TREC as part of the 2009 Million Query track. Considering the TREC categories as ground truth revealed an interesting paradox: when asked to estimate the difficulty of a query, users gave overall inaccurate assessments, largely underestimating hard queries; however, when asked to assess the individual causes that render a query difficult, user assessments largely improved. One plausible reason for this may be the users' incomplete understanding of the (well-known in IR) inverse relation between term occurrence and discriminativeness. In order to investigate further the user-perceived causes of query difficulty, a formal approach was taken, whereby user perceptions were represented as subjective beliefs in the framework of subjective logic. These beliefs were then fused using the Bayesian consensus operator, to produce estimates of overall query difficulty. The resulting estimates were found to be notably better than the direct user assessments, improving the proportion of correctly assessed hard queries from 13.1% up to 34.37%.

The main contribution of this work is in casting light into the user perceptions of query difficulty, and in comparing them to a system-based understanding of query difficulty. Future work includes investigating users' perceptions of query difficulty in relation to their own information needs, to see whether their assessments are more closely related to a system-based understanding of query difficulty, and to find ways of practically applying the user perceptions of query difficulty to improve user-system interaction design for cases of difficult queries. One possible way of doing this is by applying the subjective logic formalism presented here to represent and fuse different aspects of subjective user perceptions.

Acknowledgements. This research was partially supported by the Tools for Integrated Search project funded by Denmark's Electronic Research Library (grant number 2007-003292), and by the Relevance of Information Searched in Context project funded by the Research Council of the Danish Ministry of Culture (grant number 2008-001573).

References

1. Allan, J., Carterette, B., Dachev, B., Aslam, J.A., Pavlu, V., Kanoulas, E.: Million query track 2007 overview. In: Voorhees, E.M., Buckland, L.P. (eds.) TREC. Special Publication 500-274, National Institute of Standards and Technology, NIST (2007)
2. Carmel, D., Yom-Tov, E.: Estimating the Query Difficulty for Information Retrieval. In: Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, San Francisco (2010)

3. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: SIGIR, pp. 390–397 (2006)
4. Carterette, B., Pavlu, V., Fangz, H., Kanoulas, E.: Overview of the trec 2009 million query track. In: Voorhees, E.M., Buckland, L.P. (eds.) TREC. Special Publication 500-277, National Institute of Standards and Technology, NIST (2009)
5. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: SIGIR, pp. 299–306 (2002)
6. Hauff, C.: Predicting the Effectiveness of Queries and Retrieval Systems. PhD thesis, University of Twente (2010)
7. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 43–54. Springer, Heidelberg (2004)
8. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–20 (1972)
9. Josang, A.: A logic for uncertain probabilities. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9(3), 279–311 (2001)
10. Kumaran, G., Allan, J.: Selective user interaction. In: Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, Ø.H., Falcão, A.O. (eds.) CIKM, pp. 923–926. ACM, New York (2007)
11. Lioma, C., Blanco, R., Palau, R.M., Moens, M.-F.: A Belief Model of Query Difficulty that Uses Subjective Logic. In: Azzopardi, L., Kazai, G., Robertson, S.E., Rüger, S.M., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 92–103. Springer, Heidelberg (2009)
12. Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty - a case study on previous TREC campaigns. In: SIGIR Workshop on Predicting Query Difficulty: Methods and Applications (2005)
13. Sanderson, M., Paramita, M.L., Clough, P., Kanoulas, E.: Do user preferences and evaluation measures line up? In: Crestani, F., Marchand-Maillet, S., Chen, H.-H., Efthimiadis, E.N., Savoy, J. (eds.) SIGIR, pp. 555–562. ACM, New York (2010)
14. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: SIGIR, pp. 512–519 (2005)
15. Zhou, Y., Croft, W.B.: Ranking robustness: a novel framework to predict query performance. In: CIKM, pp. 567–574 (2006)
16. Zhou, Y., Croft, W.B.: Query performance prediction in web search environments. In: SIGIR, pp. 543–550 (2007)

A Unified Framework for Post-Retrieval Query-Performance Prediction

Oren Kurland[1], Anna Shtok[1], David Carmel[2], and Shay Hummel[1]

[1] Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel
kurland@ie.technion.ac.il, annabel@tx.technion.ac.il, projphoto@gmail.com
[2] IBM Research, Haifa Lab, Haifa 31905, Israel
carmel@il.ibm.com

Abstract. The query-performance prediction task is estimating the effectiveness of a search performed in response to a query in lack of relevance judgments. Post-retrieval predictors analyze the *result list* of top-retrieved documents. While many of these previously proposed predictors are supposedly based on different principles, we show that they can actually be derived from a novel unified prediction framework that we propose. The framework is based on using a pseudo effective and/or ineffective ranking as reference comparisons to the ranking at hand, the quality of which we want to predict. Empirical exploration provides support to the underlying principles, and potential merits, of our framework.

Keywords: query-performance prediction, post-retrieval prediction framework.

1 Introduction

There has been much work throughout recent years on predicting *query performance* [4]. That is, estimating the effectiveness of a search performed in response to a query in lack of relevance judgments. Pre-retrieval query-performance predictors, for example, analyze the query and may use corpus-based statistics [11,4]. Post-retrieval predictors [6,4] also utilize information induced from the *result list* of the most highly ranked documents.

We present a (simple) novel unified post-retrieval prediction framework that can be used to derive many previously proposed post-retrieval predictors that are supposedly based on completely different principles. The framework is based on using a pseudo effective and/or ineffective ranking(s) as reference comparisons to the ranking at hand, the effectiveness of which we want to predict. The more similar the given ranking to the pseudo effective ranking and dissimilar to the pseudo ineffective ranking the higher its effectiveness is presumed to be. As it turns out, many previous post-retrieval predictors simply differ by the choice of the pseudo (in)effective ranking that serves for reference, and/or the inter-ranking similarity measure used.

Experiments performed using TREC datasets provide empirical support to the underlying principles, and potential merits, of our framework. For example,

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 15–26, 2011.

while current predictors use either a pseudo effective or a pseudo ineffective ranking, we demonstrate the potential merits of using both.

2 Related Work

Post-retrieval query-performance prediction methods are based on analyzing the result list of top-retrieved documents [4]. These methods can be classified into three categories [4]. Clarity-based approaches [6] estimate the focus of the result list with respect to the corpus. Robustness-based approaches [19,22,18,23,2] measure the stability of the result list under perturbations of the query, documents, and the retrieval method. Score-distribution-based approaches [8,23,15] utilize properties of the retrieval scores in the result list. We show that predictors representing these three categories can be derived from, and explained by, our proposed post-retrieval prediction framework.

A utility estimation framework (UEF) [16], which inspired the development of our framework, is based on estimating a relevance model and using it to induce a pseudo effective ranking. The induced ranking serves as a reference comparison in estimating the quality of a given ranking as in our framework. Yet, UEF, which we show to be a specific case of our framework, was used to derive predictors based on a specific way of inducing a pseudo effective ranking. We show that several previous predictors can be instantiated from our framework by using different approaches for inducing a pseudo effective ranking. More importantly, in contrast to our framework, UEF does not utilize a (pseudo) ineffective ranking as a reference comparison. Thus, quite a few predictors that we derive from our framework cannot be derived from UEF. Moreover, we demonstrate in Section 4 the merits of using both pseudo effective and ineffective rankings

A conceptual framework for modeling (predicting) topic difficulty [5] is based on similarities between the query, the result list, and the corpus. In contrast, our framework predicts the effectiveness of a ranking by measuring its similarity with (pseudo) effective and ineffective rankings. The corpus, which served to induce a non-relevance model in this framework [5], is utilized in our framework for inducing pseudo ineffective rankings that are used to derive several predictors.

3 Query-Performance Prediction Framework

Suppose a retrieval method $\mathcal{M}$ is employed in response to query q over a corpus of documents $\mathcal{D}$ so as to satisfy the information need expressed by q. The goal of query-performance prediction methods is to quantify the effectiveness of the resultant corpus ranking, denoted $\pi_{\mathcal{M}}(q;\mathcal{D})$, in lack of relevance judgments.

Now, let $\pi_{opt}(q;\mathcal{D})$ be the optimal corpus ranking with respect to the information need expressed by q as defined by the probability ranking principle [13]; that is, a ranking that corresponds to the "true" degrees (probabilities) of documents' relevance. Naturally, the more "similar" the given ranking $\pi_{\mathcal{M}}(q;\mathcal{D})$ is to the optimal ranking $\pi_{opt}(q;\mathcal{D})$, the more effective it is:

$$Q(\pi_{\mathcal{M}}(q;\mathcal{D})) \stackrel{def}{=} Sim(\pi_{\mathcal{M}}(q;\mathcal{D}), \pi_{opt}(q;\mathcal{D})) \ ; \qquad (1)$$

$Q(\pi_{\mathcal{M}}(q;\mathcal{D}))$ is the quality (effectiveness) of $\pi_{\mathcal{M}}(q;\mathcal{D})$ that we aim to predict; and, $Sim(\cdot,\cdot)$ is an inter-ranking similarity measure discussed below.

One way to derive a prediction method using Eq. 1 is to try to approximate the optimal ranking. This is the task addressed, for example, by probabilistic retrieval methods that estimate the probability of a document being relevant. Now, if we have a retrieval approach that is known, in general, to be quite effective, we could use it to induce a *pseudo effective* (PE) corpus ranking $\pi_{PE}(q;\mathcal{D})$. Then, the PE ranking can be used in Eq. 1, instead of the optimal ranking, as a reference comparison in estimating (predicting) $\mathcal{M}$'s ranking effectiveness:

$$\hat{Q}_{PE}(\pi_{\mathcal{M}}(q;\mathcal{D})) \stackrel{def}{=} Sim(\pi_{\mathcal{M}}(q;\mathcal{D}), \pi_{PE}(q;\mathcal{D})) \ . \tag{2}$$

Clearly, the quality of predictors derived from Eq. 2 depends on the actual effectiveness of $\pi_{PE}(q;\mathcal{D})$, and on the inter-ranking similarity measure used. To potentially improve the ranking-quality estimate in Eq. 2, we use the dissimilarity between $\mathcal{M}$'s ranking and a *pseudo ineffective* (PIE) ranking as a means of regularization:

$$\begin{aligned}&\hat{Q}_{PE;PIE}(\pi_{\mathcal{M}}(q;\mathcal{D})) \stackrel{def}{=} \\ &\quad \alpha(q)Sim(\pi_{\mathcal{M}}(q;\mathcal{D}), \pi_{PE}(q;\mathcal{D})) - \beta(q)Sim(\pi_{\mathcal{M}}(q;\mathcal{D}), \pi_{PIE}(q;\mathcal{D})) \ ;\end{aligned} \tag{3}$$

$\alpha(q)$ and $\beta(q)$ are (query-dependent) weights. This approach is conceptually reminiscent of Rocchio's retrieval method [14] that is based on using interpolation of prototypes of relevant and non-relevant documents for query refinement.

Retrieval effectiveness measures such as mean average precision (MAP) and precision@k attribute much more importance to documents at high ranks than to those at low ranks. Consequently, post-retrieval query-performance predictors [4] analyze the *result list* of the documents most highly ranked rather than the entire corpus ranking. Along the same lines, we approximate the quality of the given corpus ranking, $\pi_{\mathcal{M}}(q;\mathcal{D})$, by focusing on the highest ranks. Formally, let $L_x^{[k]}$ denote the result list of the k highest ranked documents in x's ranking. The ranking-quality estimate from Eq. 3 is approximated using an estimate for the quality of the result list $L_{\mathcal{M}}^{[k]}$, which is in turn estimated based on the similarity of $L_{\mathcal{M}}^{[k]}$ with the result lists of the PE and PIE rankings:

$$\hat{Q}_{PE;PIE}(\pi_{\mathcal{M}}(q;\mathcal{D})) \approx \alpha(q)Sim(L_{\mathcal{M}}^{[k]}, L_{PE}^{[k]}) - \beta(q)Sim(L_{\mathcal{M}}^{[k]}, L_{PIE}^{[k]}) \ . \tag{4}$$

Various inter-ranking (list) similarity measures ($Sim(\cdot,\cdot)$) can be used. For example, if both lists are (different) rankings of the same document set, then Kendall's-τ, which uses rank information, or Pearson's correlation coefficient computed based on retrieval scores in the lists, can be applied. Document content can also be used to induce inter-ranking (list) similarity as we discuss below.

3.1 Deriving Previously Proposed Predictors

We next show that several previously proposed post-retrieval predictors can be instantiated from the framework described above (Eq. 4). Specifically, either a

pseudo effective or ineffective result list is used as a reference comparison to the given result list ($L_{\mathcal{M}}^{[k]}$), and some inter-list similarity measure is used.

Using a Pseudo Ineffective (PIE) Result List

Clarity. The clarity predictor estimates the focus of the given result list, $L_{\mathcal{M}}^{[k]}$, with respect to the corpus by measuring the (KL) divergence between their induced language models [6]. The assumption is that the more distant the models are, the more focused the result list; therefore, the higher the quality of $\pi_{\mathcal{M}}(q;\mathcal{D})$.

Clarity can be explained as a specific instance of the prediction framework described above. Let $\alpha(q) = 0$ and $\beta(q) = 1$; i.e., only a pseudo ineffective (PIE) result list $L_{PIE}^{[k]}$ is used. The PIE list is composed of k instances of the corpus that represents a general (average) non-relevant document. (The documents in the corpus can be concatenated to yield one long document to be used.) Let $p(\cdot|L)$ denote a language model induced from the document list L; and, let $Sim(L_1, L_2) \stackrel{def}{=} -KL\big(p(\cdot|L_1)||p(\cdot|L_2)\big)$ be an inter-list similarity measure that is based on the KL divergence between the lists' language models. Indeed, the clarity of $L_{\mathcal{M}}^{[k]}$ is defined as $-Sim(L_{\mathcal{M}}^{[k]}, L_{PIE}^{[k]})$; $p(\cdot|L_{\mathcal{M}}^{[k]})$ is a relevance language model [12] induced from $L_{\mathcal{M}}^{[k]}$; and, $p(\cdot|L_{PIE}^{[k]})$ is the corpus language model, as the corpus is the only (pseudo) document that appears (k times) in $L_{PIE}^{[k]}$.

Weighted information gain (WIG). The WIG predictor is based on measuring the amount of information in the given result list $L_{\mathcal{M}}^{[k]}$ with respect to that in a result list that is created using the corpus as an average non-relevant document [23]. In practice, WIG is computed by the average divergence of retrieval scores of documents in $L_{\mathcal{M}}^{[k]}$ from that of the corpus. When retrieval scores reflect surface-level document-query similarities, the higher the divergence, the higher the query-similarity documents in the list exhibit with respect to that of the corpus; consequently, the more effective $L_{\mathcal{M}}^{[k]}$ is presumed to be.

As with clarity, to derive WIG from our framework we set $\alpha(q) = 0$, $\beta(q) = 1$, and $L_{PIE}^{[k]}$ to k copies of the corpus, which serves for a non-relevant document. The (average) $L1$ distance between retrieval scores serves for an inter-list similarity measure; that is, $Sim(L_{\mathcal{M}}^{[k]}, L_{PIE}^{[k]}) \stackrel{def}{=} \frac{1}{k}(\sum_{i=1\ldots k} Score(L_{\mathcal{M}}^{[k]}(i)) - Score(L_{PIE}^{[k]}(i))$, where $L(i)$ is the document at rank i of list L and $Score(L(i))$ is its retrieval score in the list. (Recall that for $i \in \{1,\ldots,k\}$ $L_{PIE}^{[k]}(i) \stackrel{def}{=} \mathcal{D}$.)[1]

Thus, the difference between WIG and clarity, as instantiated from our framework, is the measure used to compute the (dis)similarity between the given result list and a result list composed of k copies of the corpus that serves for a non-relevant document.[2]

[1] In implementation, the retrieval scores used by WIG are further normalized so as to ensure inter-query compatibility.

[2] See Zhou [21] for an alternative view of the connection between WIG and clarity.

NQC. The NQC predictor [15] measures the standard deviation of retrieval scores in the result list. It was shown that the mean retrieval score in the list corresponds to the retrieval score of a centroid-based representation of the documents in the list [15] for some retrieval methods for which retrieval scores represent document-query similarities. Furthermore, the list centroid was argued to manifest query drift, and hence, could be thought of as a pseudo non-relevant document that exhibits relatively high query similarity. Accordingly, high divergence of retrieval scores from that of the centroid, measured by the standard deviation, was argued, and empirically shown, to imply high quality of the result list.

Hence, if we (i) set $\alpha(q) = 0$ and $\beta(q) = 1$, (ii) use k instances of the centroid-based representation of $L_{\mathcal{M}}^{[k]}$ (denoted $Cent(L_{\mathcal{M}}^{[k]})$) to create a pseudo ineffective list ($L_{PIE}^{[k]}$), and (iii) use $Sim(L_{\mathcal{M}}^{[k]}, L_{PIE}^{[k]}) \stackrel{def}{=} -\sqrt{\frac{1}{k}\sum_{i=1...k}\left(Score(L_{\mathcal{M}}^{[k]}(i)) - Score(L_{PIE}^{[k]}(i))\right)^2}$ for an inter-list similarity measure (note that $L_{PIE}^{[k]}(i) \stackrel{def}{=} Cent(L_{\mathcal{M}}^{[k]})$), we derive NQC from our framework[3].

Recall from above that WIG uses the $L1$ distance between retrieval scores in $L_{\mathcal{M}}^{[k]}$ and those in a PIE list composed of k copies of the corpus, which serves as a *general non-relevant document*. In comparison, NQC uses the $L2$ distance between the retrieval scores in $L_{\mathcal{M}}^{[k]}$ and those in a PIE list composed of k copies of a *pseudo non-relevant document* that exhibits high surface-level query similarity (i.e., $Cent(L_{\mathcal{M}}^{[k]})$).

Query-independent vs. query-dependent ranking. Another approach for producing a pseudo ineffective result list, $L_{PIE}^{[k]}$, is based on re-ranking the given result list, $L_{\mathcal{M}}^{[k]}$, using non-query-dependent information; e.g., based on documents' PageRank [3]. The idea is that the higher the divergence between $L_{\mathcal{M}}^{[k]}$'s original ranking and its query-independent re-ranked version, the higher the quality of $L_{\mathcal{M}}^{[k]}$; Kendall's-tau, for example, can serve for an inter-ranking similarity measure [3]. Thus, this approach is another instance of our framework when using only a PIE list (i.e., the query-independent ranked version of $L_{\mathcal{M}}^{[k]}$) with $\beta(q) = 1$.

Using a Pseudo Effective (PE) Result List

Query feedback. In the query feedback (QF) predictor [23], a query model is induced from $L_{\mathcal{M}}^{[k]}$ and is used to rank the entire corpus. Then, the overlap (i.e., number of shared documents) between the n_{QF} highly ranked documents by this retrieval, and the n_{QF} highly ranked documents by the given ranking $\pi_{\mathcal{M}}(q; \mathcal{D})$ (n_{QF} is a free parameter), presumably indicates the effectiveness of the latter. That is, the higher the overlap, the less non-query-related noise there is in $L_{\mathcal{M}}^{[k]}$ from which the query model was induced; hence, $L_{\mathcal{M}}^{[k]}$, and the $\pi_{\mathcal{M}}(q; \mathcal{D})$ ranking from which it was derived, are considered of higher quality.

[3] To ensure inter-query compatibility of prediction values, documents' retrieval scores are scaled using that of the corpus.

The retrieval performed over the corpus using the query model induced from $L_{\mathcal{M}}^{[k]}$ is essentially pseudo-feedback-based query-expansion retrieval. As is known, such retrieval outperforms, on average, that of using only the original query. Thus, the result list of k highest ranked documents produced by using the induced query model could be considered as pseudo effective (PE) on average; let $L_{PE}^{[k]}$ denote this list. Accordingly, the overlap at cutoff n_{QF} between $L_{\mathcal{M}}^{[k]}$ and $L_{PE}^{[k]}$ serves as the inter-list similarity measure. Setting $\alpha(q) = 1$ and $\beta(q) = 0$, i.e., using only the similarity with the pseudo effective ranking just mentioned, we get that QF is a specific instance of our framework.

Utility Estimation Framework (UEF). The basic idea underlying UEF [16] is to devise a supposedly effective representation of the underlying information need (specifically, using a relevance model approach [12]). This representation is used to re-rank the given result list $L_{\mathcal{M}}^{[k]}$. The resultant re-ranked version of $L_{\mathcal{M}}^{[k]}$ is presumably of relatively high quality, and is thereby denoted here $L_{PE}^{[k]}$. The similarity between $L_{\mathcal{M}}^{[k]}$ and $L_{PE}^{[k]}$ ($Sim(L_{\mathcal{M}}^{[k]}, L_{PE}^{[k]})$) is measured using Kendall's-τ, Pearson's coefficient, or Spearman's-ρ. The similarity value is scaled by an estimate for the quality of the information need representation. The motivation is to model the confidence in the ability to derive an effective representation of the information need, and use the level of confidence so as to adjust the prediction value. Thus, UEF is a specific instance of our proposed framework wherein $\beta(q) = 0$ (i.e., no pseudo ineffective result list is used), and $\alpha(q)$ is the estimate for the quality of the information need representation.

Autocorrelation. Applying score regularization — specifically, adjusting the retrieval score of a document using information induced from similar documents — upon the given result list $L_{\mathcal{M}}^{[k]}$ so that the resultant retrieval scores "respect" the cluster hypothesis is another way to produce a pseudo effective result list [8]. The (Pearson) correlation between the retrieval scores in $L_{\mathcal{M}}^{[k]}$ and $L_{PE}^{[k]}$ serves for an inter-list similarity measure. Hence, this (spatial) *autocorrelation* approach [8] is also an instance of our framework (with $\alpha(q) = 1$ and $\beta(q) = 0$).

Utilizing fusion. All predictors discussed above are based on a single retrieval (if at all) used to create a pseudo (in)effective ranking. Alternatively, fusion of multiple rankings can be used to produce a pseudo effective ranking [8]. Indeed, the merits of fusion, in terms of retrieval effectiveness, have been acknowledged [9]. Pearson's correlation between the given result list and that produced by fusion served for query-performance prediction [8]. Clearly, this prediction approach is a specific instance of our framework (with $\alpha(q) = 1$ and $\beta(q) = 0$).

Intermediate summary. As was shown above, various post-retrieval predictors can be derived from Eq. 4. The predictors use either a pseudo effective ranking or a pseudo ineffective ranking but not both. The pseudo effective rankings were induced using pseudo-feedback-based retrieval [23,16], score regularization [8], and fusion [8]. Pseudo-ineffective rankings were induced using the corpus [6,23], a centroid of the result list [15], and a query-independent retrieval method [3]. The inter-ranking similarity measures used were based on (i) the $L1$

[23] and $L2$ [15] distances of retrieval scores and their Pearson correlation [8,16], (ii) the KL divergence between induced language models [6], (iii) Kendall's-τ [3,16] and the document overlap [23] between result lists.

4 Experiments

We next present an empirical study of the potential merits of our framework. In Section 4.2 we explore the basic premise underlying the framework, the utilization of both pseudo effective and pseudo ineffective rankings, and a use case demonstrating the intricacies of utilizing pseudo ineffective rankings.

Parts of the study are based on utilizing (little) relevance feedback to control the effectiveness of reference rankings. Although feedback is often not available for query-performance prediction, the exploration using it shows that the ability to devise effective reference rankings to be used in our framework yields prediction quality that substantially transcends state-of-the-art.

4.1 Experimental Setup

We used the following TREC collections and queries for experiments:

Collection	Data	Num Docs	Topics
TREC4	Disks 2&3	567,529	201-250
TREC5	Disks 2&4	524,929	251-300
WT10G	WT10g	1,692,096	451-550
ROBUST	Disk 4&5-CR	528,155	301-450,601-700

Topics' titles serve for queries; for TREC4 topics' descriptions are used as titles are not available. Porter stemming and stopword removal (using INQUERY's list) were applied using the Lemur toolkit (www.lemurproject.org), which was also used for retrieval.

To measure prediction quality, we follow common practice [4] and compute Pearson's correlation between the values assigned by a predictor to queries, and the "true" average precision (AP, computed at cutoff 1000) values for these queries determined based on TREC's relevance judgments.

Language modeling framework. The goal of the predictors we study is predicting the effectiveness of rankings induced in response to the queries specified above by the *query likelihood* (QL) retrieval method [17]. Let $p(w|x)$ denote the probability assigned to term w by a (smoothed) unigram language model induced from text (collection) x. (Specific language-model induction details are provided below.) The (log) query likelihood score of document d with respect to query q ($= \{q_i\}$), which is used for ranking the corpus, is $Score_{QL}(q;d) \stackrel{def}{=} \log p(q|d) \stackrel{def}{=} \log \prod_{q_i \in q} p(q_i|d)$. The result list of k highest ranked documents is denoted $L^{[k]}_{q;QL}$.

Some of the predictors we explore utilize *relevance language models* [12]. Let R^S be a relevance model[4] constructed from a document set S: $p(w|R^S) \stackrel{def}{=}$

[4] We use the RM1 relevance model. While for retrieval purposes, RM3 [1], which interpolates RM1 with the query model is more effective, RM1 is more effective for performance prediction with the predictors we study as previously reported [16].

$\sum_{d \in S} p(w|d)wt(d)$; $wt(d)$ is d's weight ($\sum_{d \in S} wt(d) = 1$). To score document d with respect to R^S, so as to induce ranking, the minus cross entropy between R^S and d's language model is used: $Score_{CE}(R; d) \stackrel{def}{=} \sum_w p(w|R) \log p(w|d)$.

The standard pseudo-feedback-based relevance model, denoted R^{Res}, is constructed from the result list ($S \stackrel{def}{=} L^{[k]}_{q;QL}$); $wt(d) \stackrel{def}{=} p(d|q) \stackrel{def}{=} \frac{p(q|d)}{\sum_{d' \in L^{[k]}_{q;QL}} p(q|d')}$ [12]. To control the effectiveness of some reference rankings, we also use a relevance model, R^{Rel}, that is constructed from a set S of r relevant documents that are the highest ranked by QL (r is a free parameter); $wt(d) \stackrel{def}{=} \frac{1}{r}$.

Implementation. We use three state-of-the-art predictors that were shown above to be specific instances of our framework. The first is a (conceptually) generalized version of the QF method [23]: the overlap at top (n_{QF}) ranks between the given result list, $L^{[k]}_{q;QL}$, and a result list created from the corpus using a relevance model constructed from documents in $L^{[k]}_{q;QL}$ serves for prediction. Changing the relevance model enables to study the effect of using reference rankings of varying effectiveness. The other two predictors are clarity [6] and NQC [15].

We use Dirichlet-smoothed unigram document language models with the smoothing parameter set to 1000 [20]. For constructing a relevance model, a non-smoothed maximum likelihood estimate is used for document language models [16]; and, all relevance models use 100 terms [16]. For the QF and clarity predictors, the QL result list size, k, is set to 100, which yields high quality prediction [16]; for NQC, the effect of k is studied.

4.2 Experimental Results

Using Effective Rankings as Reference Comparisons. In Fig. 1 we present the effect on QF's prediction quality of using reference rankings of varying effectiveness. Specifically, we construct a relevance model R^{Rel} from r (≥ 1) relevant documents. We then depict the MAP performance of using R^{Rel} for retrieval over the corpus; and, the resultant prediction quality of QF when using the corpus ranking induced by R^{Rel} for a reference ranking. We set the overlap cutoff parameter, n_{QF}, to 10; the patterns observed in the graphs are quite similar for $n_{QF} \in \{10, 25, 50, 100\}$. For $r = 0$, we use the result-list-based relevance model, R^{Res}, which corresponds to the original QF [23].

We can see in Fig. 1 that, as is known, the retrieval effectiveness of the relevance model increases when increasing the number of relevant documents from which it is constructed. Accordingly, the resultant prediction quality of QF increases when increasing the effectiveness of the ranking induced by the relevance model; specifically, the prediction quality becomes much better than that of using the result-list-based relevance model ($r = 0$), which is the current state-of-the-art QF approach.

Hence, we see that using reference rankings of higher effectiveness, which are induced here by using relevance models of higher quality, results in improved query-performance prediction. This finding provides support to the underlying premise of our framework. That is, high quality query-performance prediction

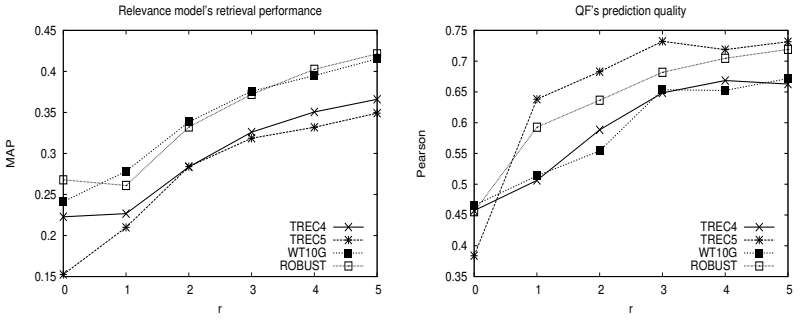


Fig. 1. Using a relevance model, R^{Rel}, constructed from r (≥ 1) relevant documents in QF; for $r = 0$, the (pseudo feedback) result-list-based relevance model, R^{Res}, is used. The left figure presents the MAP performance of using the relevance model for retrieval over the corpus. (The list of top-retrieved documents serves for reference in QF.) The right figure presents QF's resultant prediction quality.

can be attained by using an estimate of the "optimal" ranking as a reference comparison in estimating the effectiveness of the given ranking.

Using Both Effective and Ineffective Rankings. As the predictors discussed in Section 3 use either a (pseudo) effective or ineffective reference rankings, but not both, we now study the potential merits of using both.

We create an effective corpus ranking using a relevance model, R^{Rel}, constructed from 5 relevant documents. To measure the similarity between the corpus ranking and the QL ranking, the quality of which we want to predict, we use the **drift** method [7]. That is, we construct a relevance language model, denoted R_{QL}, from the QL result list ($L^{[k]}_{q;QL}$; $k = 100$); and, from the top-100 documents retrieved from the corpus using R^{Rel}, denoted $R_{R^{Rel}}$; uniform weights ($wt(d) \stackrel{def}{=} \frac{1}{100}$) are used, and R_{QL} and $R_{R^{Rel}}$ use 100 terms; $R_{R^{Rel}}$ is Jelinek-Mercer smoothed using a smoothing weight of 0.1. The minus KL divergence, $-KL\big(p(\cdot|R_{QL})||p(\cdot|R_{R^{Rel}})\big)$, serves for inter-list similarity measure. The resultant drift-based predictor is a variant of QF that used document overlap for inter-list similarity. We use this variant to have proper interpolation in Eq. 4 with the dissimilarity to an ineffective corpus-based ranking used by clarity.

Recall from Section 3 that clarity is defined as $KL\big(p(\cdot|R^{Res})||p(\cdot|L^{[k]}_{PIE})\big)$; $L^{[k]}_{PIE}$ is an ineffective list composed of k ($= 100$) copies of the corpus. We set $\alpha(q) \stackrel{def}{=} \lambda$ and $\beta(q) \stackrel{def}{=} (1 - \lambda)$ in Eq. 4 ($\lambda \in \{0, 0.1, \ldots, 1\}$) and derive the (novel) **drift+clarity** predictor, the quality of which is reported in Table 1. To study the potential prediction quality of utilizing both effective and ineffective lists, λ is set to a value that yields optimal prediction quality per corpus: 0.5, 0.3, 0.5, and 0.3 for TREC4, TREC5, WT10G and ROBUST, respectively.

It is important to conceptually differentiate the drift+clarity predictor just presented from the general case of linear interpolation of *prediction values*. Such interpolation can be based on the output of predictors that can use, for example, different inter-ranking similarity measures [23,10,16]. In contrast, drift+clarity is

derived as a single predictor from Eq. 4, wherein the similarity of the given result list with an effective reference list (created using R^{Rel}), and dissimilarity with an ineffective reference list (created from the corpus) are interpolated; the (minus) KL divergence between lists' language models serves for inter-list similarity measure. In implementation, however, drift+clarity amounts to interpolating the prediction values of drift and clarity.

We see in Table 1 that although drift is much inferior to clarity, drift+clarity is much superior to clarity. This finding supports the potential merits of using both effective and ineffective reference rankings for performance prediction.

On Using Ineffective Rankings as Reference Comparisons. The NQC predictor [15] turns out to be an interesting example for demonstrating the merits, and intricacies, of using a pseudo ineffective reference ranking. NQC measures the standard deviation of retrieval scores in the result list ($L^{[k]}_{q;QL}$). As noted above, the mean retrieval score was shown to be the retrieval score of a centroid-based representation of the list; and, the centroid was argued to serve as a pseudo non-relevant document that exhibits high query similarity [15]. We showed above that NQC can be derived from our framework using a pseudo ineffective list that is composed of multiple copies of the centroid. In Fig. 2 we present the effect on NQC's prediction quality of varying the result list size, k. Below we argue that varying k affects the usefulness, in terms of resultant query-performance prediction, of the pseudo ineffective list created from the centroid.

We see in Fig. 2 that NQC's prediction quality monotonically improves when increasing k up till a point from which it monotonically decreases. Indeed, with very few documents in the result list (small k), the centroid is much affected by highly ranked relevant documents; thereby, it is not a very good basis for a useful ineffective reference list. Having more documents in the list when increasing k towards its optimal value, results in considering more non-relevant query-similar documents; thus, the centroid's usefulness for constructing ineffective reference ranking grows, and accordingly, prediction is improved.

Increasing k beyond its optimal value results in the centroid being much affected by non-relevant documents that exhibit low query similarity. Consequently, the centroid gradually becomes a "general" non-relevant document (as the corpus), rather than a query-similar non-relevant one. Now, the centroid's high query similarity was argued to be an important factor in NQC's high quality prediction [15]. Accordingly, further increasing k makes the centroid-based list less informative as a reference comparison thus decreasing prediction quality.

Table 1. Using both effective (drift) and ineffective (clarity) reference rankings for prediction (drift+clarity). Boldface marks the best result per column

	TREC4	TREC5	WT10G	ROBUST
drift	0.406	0.081	0.317	0.130
clarity	0.448	0.426	0.330	0.508
drift+clarity	**0.588**	**0.461**	**0.412**	**0.521**

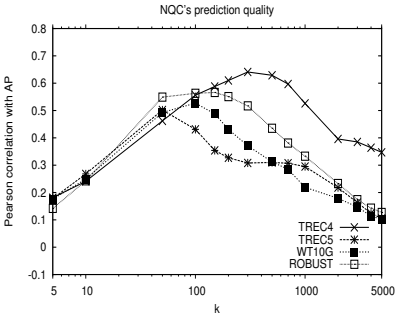


Fig. 2. NQC's prediction quality as a function of the result list size, k

We conclude that it is not only the ineffectiveness, in terms of retrieval performance, of the reference list that is important for successful performance prediction, but also the characteristics of the documents in it.

5 Summary and Future Work

We presented a novel unified framework for post-retrieval query-performance prediction which we used for deriving previously proposed predictors that are supposedly based on completely different principles. The framework uses (pseudo) effective and/or ineffective rankings as reference comparisons in estimating the effectiveness of a given ranking. Empirical exploration, based in part on exploiting little relevance feedback to induce effective reference rankings, provided support to the underlying principles, and potential merits, of the framework. Devising improved pseudo (in)effective reference rankings *for a given ranking* with zero feedback, and applying the framework to devise new post-retrieval predictors, is a future venue.

Acknowledgments. We thank the reviewers for their comments. This paper is based upon work supported in part by the Israel Science Foundation under grant no. 557/09, and by IBM's SUR award. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

1. Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, F., Larkey, L., Li, X., Smucker, M.D., Wade, C.: UMASS at TREC 2004 — novelty and hard. In: Proceedings of TREC-13, pp. 715–725 (2004)
2. Aslam, J.A., Pavlu, V.: Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 198–209. Springer, Heidelberg (2007)
3. Bernstein, Y., Billerbeck, B., Garcia, S., Lester, N., Scholer, F., Zobel, J.: RMIT university at trec 2005: Terabyte and robust track. In: Proceedings of TREC-14 (2005)

4. Carmel, D., Yom-Tov, E.: Estimating the Query Difficulty for Information Retrieval. In: Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, San Francisco (2010)
5. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: Proceedings of SIGIR, pp. 390–397 (2006)
6. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proceedings of SIGIR, pp. 299–306 (2002)
7. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: A language modeling framework for selective query expansion. Tech. Rep. IR-338, Center for Intelligent Information Retrieval, University of Massachusetts (2004)
8. Diaz, F.: Performance prediction using spatial autocorrelation. In: Proceedings of SIGIR, pp. 583–590 (2007)
9. Fox, E.A., Shaw, J.A.: Combination of multiple searches. In: Proceedings of TREC-2 (1994)
10. Hauff, C., Azzopardi, L., Hiemstra, D.: The combination and evaluation of query performance prediction methods. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 301–312. Springer, Heidelberg (2009)
11. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 43–54. Springer, Heidelberg (2004)
12. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proceedings of SIGIR, pp. 120–127 (2001)
13. Robertson, S.E.: The probability ranking principle in IR. Journal of Documentation, 294–304 (1977)
14. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313–323. Prentice Hall, Englewood Cliffs (1971)
15. Shtok, A., Kurland, O., Carmel, D.: Predicting query performance by query-drift estimation. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 305–312. Springer, Heidelberg (2009)
16. Shtok, A., Kurland, O., Carmel, D.: Using statistical decision theory and relevance models for query-performance prediction. In: Proceedings of SIGIR, pp. 259–266 (2010)
17. Song, F., Croft, W.B.: A general language model for information retrieval (poster abstract). In: Proceedings of SIGIR, pp. 279–280 (1999)
18. Vinay, V., Cox, I.J., Milic-Frayling, N., Wood, K.R.: On ranking the effectiveness of searches. In: Proceedings of SIGIR, pp. 398–404 (2006)
19. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: Proceedings of SIGIR, pp. 512–519 (2005)
20. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR, pp. 334–342 (2001)
21. Zhou, Y.: Retrieval Performance Prediction and Document Quality. PhD thesis, University of Massachusetts (September 2007)
22. Zhou, Y., Croft, W.B.: Ranking robustness: A novel framework to predict query performance. In: Proceedings of CIKM, pp. 567–574 (2006)
23. Zhou, Y., Croft, W.B.: Query performance prediction in web search environments. In: Proceedings of SIGIR, pp. 543–550 (2007)

Predicting the Performance of Recommender Systems: An Information Theoretic Approach

Alejandro Bellogín, Pablo Castells, and Iván Cantador

Universidad Autónoma de Madrid
Escuela Politécnica Superior, Departamento de Ingeniería Informática
Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{alejandro.bellogin,pablo.castells,ivan.cantador}@uam.es

Abstract. Performance prediction is an appealing problem in Recommender Systems, as it enables an array of strategies for deciding when to deliver or hold back recommendations based on their foreseen accuracy. The problem, however, has been barely addressed explicitly in the area. In this paper, we propose adaptations of query clarity techniques from ad-hoc Information Retrieval to define performance predictors in the context of Recommender Systems, which we refer to as user clarity. Our experiments show positive results with different user clarity models in terms of the correlation with single recommender's performance. Empiric results show significant dependency between this correlation and the recommendation method at hand, as well as competitive results in terms of average correlation.

Keywords: performance prediction, recommender systems, language models.

1 Introduction

Performance prediction has gained increasing attention in Information Retrieval (IR) since the late 90's, and has become an established research topic in the field [6]. It has been mostly addressed as a query performance issue, which refers to the performance of an IR system in response to a specific query. Particularly effective predictors have been defined based on language models by the so-called clarity score, which captures the ambiguity in a query with respect to the collection, or a specific result set [6].

Performance prediction finds a special motivation in Recommender Systems (RS). Contrary to query-based retrieval, as far as the initiative relies on the system, it may decide to produce recommendations or hold them back, depending on the expected level of performance on a per case basis, delivering only the sufficiently reliable ones. The problem of performance prediction, however, has barely been addressed in RS to date. The issue is in fact tackled in the RS literature by ad hoc heuristic tweaks – evidencing the relevance of the problem–, but has not been studied and addressed in a principled way. Examples of such heuristic approaches are significance weighting [12] and confidence [18], where additional computations (mainly normalizations) are introduced in order to better estimate the final prediction ratings.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 27–39, 2011.

Performance prediction finds further motivation in RS, as the performance of individual recommendation methods is highly sensitive to different conditions, such as data sparsity, quality, and reliability, which in real settings are subject to an ample dynamic variability. Hence, being able to estimate in advance which recommenders are likely to provide the best output in a particular situation opens up an important window for performance enhancement. Alternatively, estimating which users in the system are likely to receive worse recommendations allows for modifications in the recommendation algorithms to predict this situation, and react in advance.

In the research presented here, we consider the adaptation –and area-specific elaborations thereupon– to RS of principles that have been proposed and developed in ad-hoc IR. In particular, the approaches based on Information Theory principles and measures, as developed in the query clarity models, have shown to be useful in many ways to deal effectively with poorly-performing queries [19]. We propose different vocabulary spaces where clarity definition may be applied to, in order to better capture the ambiguity in user preferences. Moreover, we define alternative statistical models and estimating approaches, under different independence assumptions. In conducted experiments, we have obtained similar correlation values to those of state-of-the-art predictors in terms of average correlation. We also find significant differences in correlation between different recommenders and the same predictor.

2 Performance Prediction in Information Retrieval

Query performance prediction in IR refers to the performance of an IR system in response to a specific query. It also relates to the appropriateness of a query as an expression for a user information need. In the literature, prediction methods have been classified into two groups depending on the available data used for prediction [9]: pre-retrieval approaches, which make the prediction before the retrieval stage, and post-retrieval approaches, which use the rankings produced by the retrieval engine.

Pre-retrieval approaches have the advantage that the prediction can be taken into account to improve the retrieval process itself. These predictors, however, have the potential handicap, with regards to their accuracy, that the extra retrieval effectiveness cues available after the system response are not exploited [19]. Query scope [11] is an example of this type of predictors. It is a measure of the specificity of a query, which is quantified as the percentage of documents in the collection that contain at least one query term. Other examples such as statistic approaches based on Inverse Document Frequency (IDF), and variations thereof, have also been proposed [11, 16]. He & Ounis [11] propose a predictor based on the standard deviation of the IDF of the query terms. Plachouras et al. [16] represent the quality of a query term by a modification of IDF, where instead of the number of documents, the number of words in the whole collection is used, and the query length acts as a normalizing factor. These IDF-based predictors obtained moderate correlation with respect the query performance. Linguistic approaches have also been investigated [14].

Secondly, post-retrieval predictors make use of retrieved results. Broadly speaking, techniques in this category provide better prediction accuracy [2, 19]. However, computational efficiency is usually a problem for many of these techniques, and furthermore, the predictions cannot be used to improve the retrieval strategies, unless some

kind of iteration is applied, as the output from the retrieval system is needed to compute the predictions in the first place. Most effective predictors have been defined based on language models by the so-called clarity score, which captures the (lack of) ambiguity in a query with respect to a specific result set, or the whole collection [6, 19] (the second case thus can be considered as a pre-retrieval predictor, since it does not make use of the result set). Besides query clarity, other post-retrieval predictors have been defined based on the differences in ranking between the original input and after query or document perturbation (see [9] for a summary of these methods).

In this work, we focus on the clarity score predictor, which is measured as the Kullback-Leibler divergence, and estimates the coherence of a collection with respect to a query q in the following way, given the vocabulary $\mathcal{V}$ and a subset of the document collection R:

$$\text{clarity}(q) = \sum_{w \in \mathcal{V}} p(w|q) \log_2 \frac{p(w|q)}{p_c(w)}$$

$$p(d|q) = p(q|d)p(d);\ p(q|d) = \prod_{w_q \in q} p(w_q|d)$$

$$p(w|q) = \sum_{d \in R} p(w|d)p(d|q)\,; p(w|d) = \lambda p_{\text{ml}}(w|d) + (1-\lambda)p_c(w)$$

The clarity value can be reduced, thus, to an estimation of the prior $p_c(w)$ and the posterior $p(w|q)$ of query terms w over documents d, based on term frequencies and smoothing. Cronen-Townsend et al [6] showed that clarity is correlated with performance, demonstrating that the result quality is largely influenced by the amount of uncertainty involved in the inputs the system takes. In this sense, queries whose likely documents are a mix of documents from disparate topics receive lower score than if they result in a topically-coherent retrieved set. Several works have exploited its functionality and predictive capabilities [5, 7, 8], supporting its effectiveness in terms of performance prediction and high degree of adaptation.

3 Predictive Models of Recommendation Performance

Predicting the performance of recommender systems requires the definition of the key element we want to predict the performance for. In this paper, we identify the user having the role of the query in an IR system, although an equivalent development could be made for items instead of users.

In the following, we define different user performance predictors, whose main goal is to infer how good or bad the system is expected to perform for a given user. We propose a fairly general adaptation of query clarity, which may be instantiated in different models, depending on the input spaces considered. Specifically, our adaptation of query clarity has the following formulation:

$$\text{clarity}(u) = \sum_{x \in X} p(x|u) \log_2 \frac{p(x|u)}{p(x)} \quad (1)$$

As we can observe, the clarity formulation strongly depends on a "vocabulary" space X, which further constrains the user-conditioned model (or user model for short) $p(x|u)$, and the background probability $p(x)$. In ad-hoc IR, this space is typically the space of words, and the query language model is a probability distribution over words [6]. In RS, however, we may have different interpretations, and thus, different formulations for such a probabilistic framework, as we shall show. In all cases, we will need to model and estimate two probability distributions: first, the probability that some event (depending on the current probability space X) is generated by the user language model (*user model*); and second, the probability of generating that event without any constraint (*background model*). In Table 1, we propose three different vocabulary spaces for X, along with the associated probabilistic models.

Table 1. Three possible user clarity formulations, depending on the interpretation of the vocabulary space

User clarity	Vocabulary Space	User model	Background model
Rating-based	Ratings	$p(r\|u)$	$p_c(r)$
Item-based	Items	$p(i\|u)$	$p_c(i)$
Item-and-rating-based	Items rated by the user	$p(r\|i,u)$	$p_{ml}(r\|i)$

In all the above formulations, user clarity is in fact the difference (Kullback-Leibler divergence) between a user model and a background model. The use of user and background distributions as a basis to predict recommendation performance lies on the hypothesis that a user probability model being close to the background (or collection) model is a sign of ambiguity or vagueness in the evidence of user needs, since the generative probabilities for a particular user are difficult to singularize from the model of the collection as a whole. In IR, this fact is interpreted as a query whose ranked documents are a mix of articles about different topics [6].

As stated in [6], language models capture statistical aspects of the generation of language. Therefore, if we use different vocabularies, we may capture different aspects of the user. Specifically, for each of the vocabulary spaces defined in Table 1, we assume different user-specific interpretations. The rating-based clarity model captures how differently a user uses rating values (regardless of the items the values are assigned to) with respect to the rest of users in the community. The item-based clarity takes into account which items have been rated by a user, and therefore, whether she rates (regardless of the rating value) the most rated items in the system or not. Finally, the item-and-rating-based clarity computes how likely a user would rate each item with some particular rating value, and compares that likelihood with the probability that the item is rated with some particular rating value.

In this sense, the item-based user dependent model makes the assumption that some items are more likely to be generated for some users than for others depending on their previous preferences. The rating-based model, on the other hand, captures the likelihood of a particular rating value being assigned by a user, which is an event not as sparse as the previous one with a larger number of observations. Finally, the item-and-rating-based model is a combination of the previous models, by assuming unified models which incorporate items and ratings.

In the next section, we get into details on the formal definition of the u, i, and r random variables introduced in the above equations, along with the practical estimation of the involved distributions.

4 Ground Models

We ground the different clarity measures defined in the previous section upon a rating-oriented probabilistic model very similar to the approaches taken in [13] and [18]. The sample space for the model is the set $\mathcal{U}\times\mathcal{I}\times\mathcal{R}$, where $\mathcal{U}$ stands for the set of all users, $\mathcal{I}$ is the set of all items, and $\mathcal{R}$ is the set of all possible rating values. Hence an observation in this sample space consists of a user assigning a rating to an item. We consider three natural random variables in this space: the user, the item, and the rating value, involved in a rating assignment by a user to an item. This gives meaning to the distributions expressed in the different versions of clarity as defined in the previous section. For instance, $p(r|i)$ represents the probability that a specific item i is rated with a value r –by a random user–, $p(i)$ is the probability that an item is rated –with any value by any user–, and so on.

The probability distributions upon which the proposed clarity models are defined can use different estimation approaches, depending on the independence assumptions and the amount of involved information. Background models are estimated using relative frequency estimators, that is:

$$p_c(r) = \frac{|\{(u,i) \in \mathcal{U}\times\mathcal{I} \,|r(u,i) = r\}|}{|\{(u,i) \in \mathcal{U}\times\mathcal{I} \,|r(u,i) \neq \emptyset\}|};\; p_c(i) = \frac{|\{u \in \mathcal{U} \,|r(u,i) \neq \emptyset\}|}{|\{(u,j) \in \mathcal{U}\times\mathcal{I} \,|r(u,j) \neq \emptyset\}|}$$

$$p_{ml}(r|i) = \frac{|\{u \in \mathcal{U} \,|r(u,i) = r\}|}{|\{u \in \mathcal{U} \,|r(u,i) \neq \emptyset\}|};\; p_{ml}(r|u) = \frac{|\{i \in \mathcal{I} \,|r(u,i) = r\}|}{|\{i \in \mathcal{I} \,|r(u,i) \neq \emptyset\}|}$$

These are maximum likelihood estimations in agreement with the meaning of the random variables as defined above. Starting from these estimations, user models can be reduced to the above terms by means of different probabilistic expansions and reformulations, which we define next for each of the models introduced in the previous section.

Item based model. The $p(i|u)$ model can be simply expanded through ratings, but under two different assumptions: the item generated by the model only depends on the rating value, independently from the user or, in the contrary, depends on both the user and the rating). These alternatives lead to the following development, respectively:

$$p_R(i|u) = \sum_{r\in\mathcal{R}} p_{ml}(i|r)p_{ml}(r|u)$$

$$p_{UR}(i|u) = \sum_{r\in\mathcal{R}} p(i|u,r)p_{ml}(r|u)$$

Rating based model. This model assumes that the rating value generated by the probability model depends on both the user and the item at hand. For this model, we sum over all possible items in the following way:

$$p(r|u) = \sum_{r(u,i)=r} p(r|\mathrm{i}, u)p(i|u)$$

where the $p(i|u)$ term can be developed as in the item-based model above. The term $p(r|\mathrm{i}, u)$ requires further development, which we define in the next model.

Item-and-rating based model. Three different models can be derived depending on how the Bayes' rule is applied. In the same way as proposed in [18], three relevance models can be defined, namely a user-based, an item-based, and a unified relevance model:

$$p_U(r|i,u) = \frac{p(u|r,i)p_{ml}(r|i)}{\sum_{r\in\mathcal{R}} p(u|r,i)p_{ml}(r|i)}$$

$$p_I(r|i,u) = \frac{p(i|r,u)p_{ml}(r|u)}{\sum_{r\in\mathcal{R}} p(i|u,r)p_{ml}(r|u)}$$

$$p_{UI}(r|i,u) = \frac{p(u,i|r)p_{\mathrm{c}}(r)}{\sum_{r\in\mathcal{R}} p(u,i|r)p_{\mathrm{c}}(r)}$$

The first derivation induces a user-based relevance model because it measures by $p(u|r,i)$ how probable it is that a user rates item i with a value r. The item-based relevance model is factorized proportional to an item-based probability, i.e., $p_I(r|i,u) \propto p(i|r,u)$. Finally, in the unified relevance model, we have $p_{UI}(r|i,u) \propto p(u,i|r)$.

Different combinations of distribution formulations and estimations result in a fair array of alternatives. Among them, we focus on a subset that is shown in Table 2, which provide the most interesting combinations, in terms of experimental efficiency, of user and background distributions for each clarity model. These alternatives are further analyzed in detail in the next sections.

Table 2. Different user clarity models implemented

User clarity name	User dependent model	Background model
RatUser	$p_U(r\|i,u); p_{UR}(i\|u)$	$p_{\mathrm{c}}(r)$
RatItem	$p_I(r\|i,u); p_{UR}(i\|u)$	$p_{\mathrm{c}}(r)$
ItemSimple	$p_R(i\|u)$	$p_{\mathrm{c}}(i)$
ItemUser	$p_{UR}(i\|u)$	$p_{\mathrm{c}}(i)$
IRUser	$p_U(r\|i,u)$	$p_{ml}(r\|i)$
IRItem	$p_I(r\|i,u)$	$p_{ml}(r\|i)$
IRUserItem	$p_{UI}(r\|i,u)$	$p_{ml}(r\|i)$

5 Qualitative Observation

In order to illustrate the proposed prediction framework and give an intuitive idea of what the user characteristics predictors are capturing, we show the relevant aspects of specific users that result in clearly different predictor values, in a similar way to the examples provided in [6] for query clarity. We compare three user clarity models out of the seven models presented in Table 2: one for each formulation included in Table 1. In order to avoid distracting biases on the clarity scores that a too different number of ratings between users might cause, we have selected pairs of users with a similar number of ratings. This effect would be equivalent to that found in IR between the query length and its clarity for some datasets [9].

Table 3. Two example users, showing the number of ratings they have entered, and their performance prediction values for three user clarity models

User	Number of ratings	ItemUser clarity	RatItem clarity	IRUserItem clarity
u_1	51	216.015	28.605	6.853
u_2	52	243.325	43.629	13.551

Table 3 shows the details of two sample users on which we will illustrate the effect of the predictors. As we may see in the table, u_2 has a higher clarity value than u_1 for the three models analyzed. That is, according to our theory, u_2 is less "ambiguous" than u_1.

Figure 1 shows the clarity contribution in a term-by-term basis for one of the item-and-rating-based clarity models –where, in this case, terms are equivalent to a pair (rating, item)– as done in [6]. In the figure, we plot $p(r|u,i)\log_2(p(r|u,i)/p(r|i))$ for the different terms in the collection, sorted in descending order of contribution to the user model, i.e., $p(r|u,i)$, for each user. For the sake of clarity, only the top 20 contributions are plotted. We may see how the user with the smaller clarity value receives lower contribution values than the other user. This observation is somewhat straightforward since the clarity value, as presented in equation 1, is simply the sum of all these contributions, over the set of terms conforming the vocabulary. In fact, the figures are analogous for the rest of the models, since one user always obtains higher clarity value than the other.

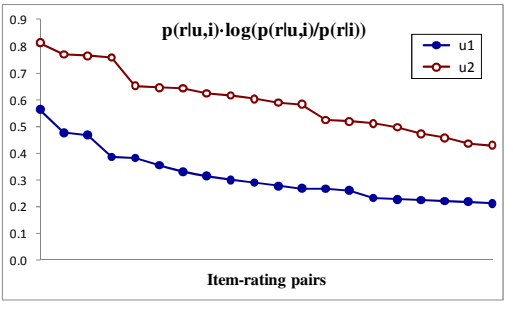


Fig. 1. Term contributions for each user, ordered by their corresponding contribution to the user language model. IRUserItem clarity model.

Let us now analyze more detailed aspects in the statistical behavior of the users that explain their difference in clarity. The IRUserItem clarity model captures how differently a user rates an item with respect to the community. Take for instance the top item-rating pairs for users 1 and 2 in the above graphic. The top pair for u_2 is (4, "McHale's Navy"). This means that the probability of u_2 rating this movie with 3 is much higher than the background probability (considering the whole user community) of this rating for this movie. Indeed, we may see that u_2 rated this movie with a 3, whereas the community mode rating is 1 –quite farther away from 4. This is the trend in a clear user. On the other extreme of the displayed values, the bottom term in the figure for user 1 is (2, "Donnie Brasco"), which is rated by this user with a 5, and the community mode rating for this item is 4, thus showing a very similar trend between both. This is the characteristic trend of a non-clear user.

Furthermore, if we compare the background model with the user model, we obtain more insights about how our models are discriminating distinctive from mainstream behavior. This is depicted in Fig. 2. In this situation, we select those terms which maximize the difference between the user and background models. Then, for this subset of the terms, we sort the vocabulary with respect to its collection probability, and then we plot the user probability model for each of the terms in the vocabulary.

These figures show how the most ambiguous user obtains a similar distribution to that of the background model, while the distribution of the less ambiguous user is more different. In the rating-based model this effect is clear, since the likelihood of not so popular rating values (i.e., a '5') is larger for user 2 than for user 1, and at the same time, the most popular rating value (a '4') is much more likely for user 1. The figure about the ItemUser model is less clear in this aspect, although two big spikes are observed for user 1 with respect to the collection distribution, which correspond with two strange movies: 'Waiting for Guffman' and 'Cry, the beloved country', both with a very low collection probability. Finally, the figure about the IRUserItem model successfully shows how user 2 has more spikes than user 1, indicating a clear divergence from the backgrund model; in fact, user 1's distribution partially mimics that of the collection. In summary, the different models proposed are able to successfully separate information concerning the user and that from the collection, in order to infer whether a user is different or similar from the collection as a whole.

Finally, it is worth noticing the relation between the clarity value and the performance metric. For instance, the value of nDCG@50 for user 1 is 0.288, and for user 2 is 0.371. In this situation, thus, the relation is linear, since performance values increase with clarity values. As we shall show in the next sections, this is coherent with the empirical correlation, which is, in median, between 0.25 and 0.50. This seems to indicate that users who follow mainstream trends are more difficult to be suggested successful items by a recommender system. In IR, one can observe a similar trend: more ambiguous (mixture of topics) queries perform worse than higher-coherence queries [6]. Note that this result might seem contradictory with the popular intuition of the *gray sheep* user who is difficult to get accurate recommendations because he lacks enough similarity with the rest of users. This trend may suggest a revision or perhaps just a more precise definition of what a gray sheep –as a performance challenging situation– really is.

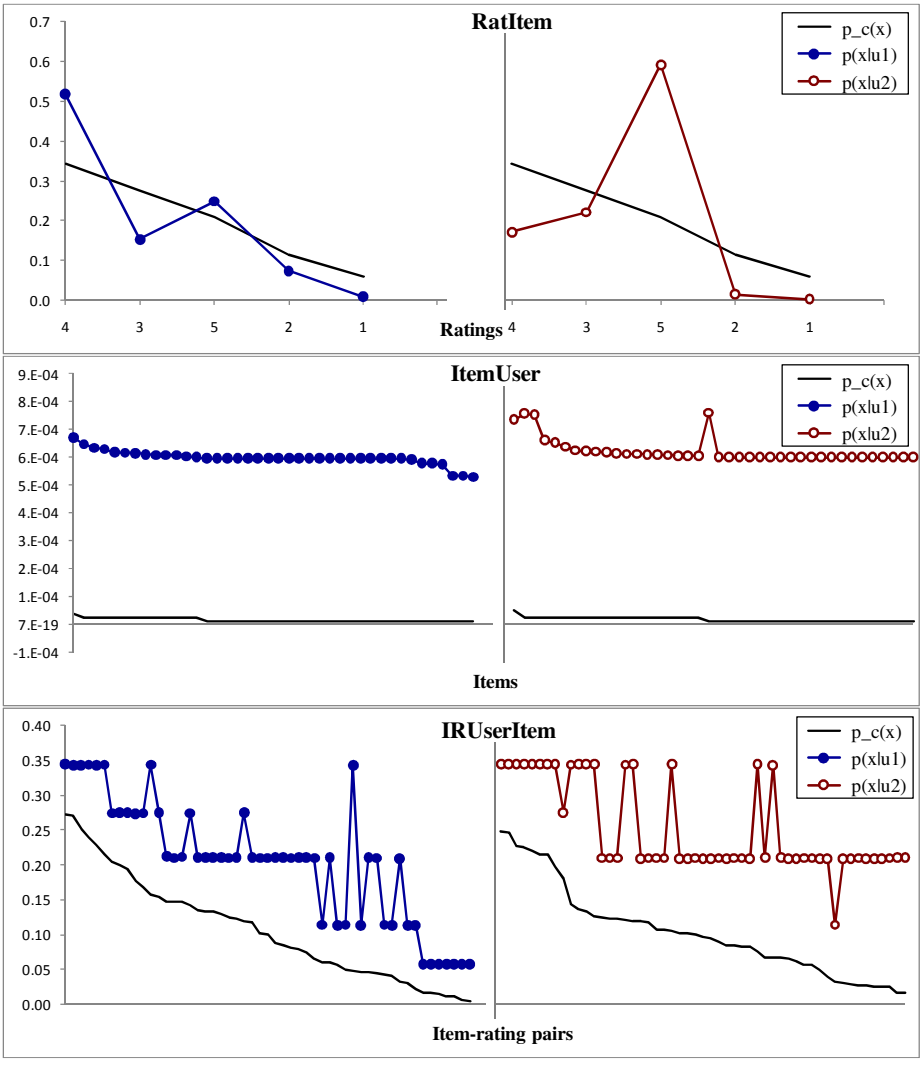


Fig. 2. User language model sorted by collection probability

6 Experiments

In this section, we study the correlation of the user performance predictors defined in previous sections and the performance of different recommenders. We use for this purpose the Movielens 100K[1] dataset, with a 5-fold cross validation of all tests. We test two state-of-the-art CF algorithms [1] (user-based with 50 neighbors, denoted as

[1] Available at http://www.grouplens.org/node/73

UB, and item-based, as IB) as implemented in the Mahout library[2]. We used two additional algorithms, recently developed, which obtain very good performance in terms of precision metrics, which we denote as TF-L1 and TF-L2 [4]. They implement an item-based CF approach with different normalization and weighting functions for the similarity or rating values. Finally, we implemented a content-based recommender (denoted as CBF) using movie genre, director, and country, from IMDb[3], as item attributes.

Table 4 shows the Pearson's correlation values between the predictors presented in previous sections, and the nDCG when only the top 50 items are considered (nDCG@50). We can observe fairly high correlation values for recommenders TF-L1 and TF-L2, comparable to results in the query performance literature. A slightly lower correlation is found for UB, whereas an insignificant value is observed for CBF and IB. These results are consistent when other performance metrics are used such as MAP, and at different cutoff lengths. Spearman's correlation yields similar values.

Table 4. Pearson's correlation between predictors and nDCG@50 for different recommenders

Predictor	CBF	IB	TF-L1	TF-L2	UB	**Median**	**Mean**
ItemSimple	0.257	0.146	0.521	0.564	0.491	0.491	0.396
ItemUser	0.252	0.188	0.534	0.531	0.483	0.483	0.398
RatUser	0.234	0.182	0.507	0.516	0.469	0.469	0.382
RatItem	0.191	0.184	0.442	0.426	0.395	0.395	0.328
IRUser	0.171	-0.092	0.253	0.399	0.257	0.253	0.198
IRItem	0.218	0.152	0.453	0.416	0.372	0.372	0.322
IRUserItem	0.265	0.105	0.523	0.545	0.444	0.444	0.376

The standard procedure in IR for this kind of evaluation is to compute correlations between the predictor(s) and one retrieval model (like in [6, 10]) or an average of several methods [14]. This approach may hide the correlation effect for some recommenders, as we may observe from the median and mean correlation values, which are still very large despite the fact that two of the recommenders analyzed have much lower correlations. These aggregated values, i.e., the mean and the median, provide competitive correlation values when compared with those in the literature.

We believe the difference in correlation for CBF and IB recommenders may be explained considering two factors: the actual recommender performance, and the input sources used by the recommender. With regards to the first factor, the IB algorithm performs poorly (in terms of the considered ranking quality metrics, such as nDCG and MAP) in comparison to the rest of recommenders. It seems natural that a good predictor for a well performing algorithm (specifically, TF-L2 is the best performing recommender in this context) would hardly correlate at the same time with a poorly performing one.

This does not explain however the somewhat lower correlation with the content-based recommender, which has better performance than UB. The input information that this recommender and the predictors take are very different: the latter compute probability distributions based on ratings given by users to items, while the former

[2] Available at http://mahout.apache.org
[3] Internet Movie Database, http://www.imdb.com

uses content features from items, such as directors and genres. Furthermore, the CBF recommender is not coherent with the inherent probabilistic models described by the predictors, since the events modeled by each of them are different: CBF would be related with the likelihood an item is described by the same features as those items preferred by the user, whereas predictors are related with the probability that an item is rated by a user. Moreover, the predictors' ground models coherently fit in the standard CF framework [18], which reinforces the suitability of the user performance predictors presented herein, at least for CF recommenders.

It is worth noting to this respect that most clarity-based query performance prediction methods in IR study their predictive power on language modeling retrieval systems [6, 10, 20] or similar approaches [11]. This suggests that a well performing predictor should be defined upon common spaces, models, and estimation techniques as the retrieval system the performance of which is meant to be predicted.

7 Conclusion

We have proposed adaptations of query clarity techniques from ad-hoc Information Retrieval to define performance predictors in Recommender Systems. Taking inspiration in the query performance predictor known as query clarity, we have defined and elaborated in the Recommender Systems domain several predictive models according to different formulations and assumptions.

We obtain strong correlation values confirming that our approach results in a high predictive power for recommender systems performance. As a side-effect, our study introduces an interesting revision of the gray sheep user concept. A simplistic interpretation of the gray sheep intuition would suggest that users with a too unusual behavior are a difficult target for recommendations. It appears however in our study that, on the contrary, users who somewhat distinguish themselves from the main trends in the community are easier to give well-performing recommendations. This suggests that perhaps the right characterization of a gray sheep user might be one who has scarce overlap with other users. On the other hand, the fact that a clear user distinguishes herself from the aggregate trends does not mean that she does not have a sufficiently strong neighborhood of similar users.

Besides the theoretic interest per se, we envision two potential applications for the proposed prediction techniques: dynamic neighbor weighting in collaborative filtering, and the dynamic adjustment of recommender ensembles. The first problem was already researched in [3], where a dynamic collaborative filtering algorithm outperformed the standard formulation by promoting neighbors that are expected to perform better in a nearest-neighbor recommendation algorithm. We are currently working on the second problem, namely, how to dynamically choose the best weights in a recommender ensemble. An additional application –somewhat obvious, albeit not less useful– is to use performance prediction to trigger recommendations only when the predicted performance is above some threshold, thus saving the user potential misses, plus the computational cost. We also plan to continue exploring further performance predictors. Specifically, we are interested in incorporating explicit recommender dependence into the predictors, so as to better exploit the information managed by the recommender, in order to achieve an even higher final correlation between them.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), University Autónoma de Madrid and the Community of Madrid (CCG10-UAM/TIC-5877).

References

1. Adomavicius, G., Tuzhilin, T.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 734–749 (2005)
2. Amati, G., Carpineto, C., Romano, G.: Query difficulty, robustness, and selective application of query expansion. In: McDonald, S., Tait, J.I. (eds.) ECIR 2004. LNCS, vol. 2997, pp. 127–137. Springer, Heidelberg (2004)
3. Bellogín, A., Castells, P.: A performance prediction approach to enhance collaborative filtering performance. 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. 382–393. Springer, Heidelberg (2010)
4. Bellogín, A., Wang, J., Castells, P.: Text retrieval methods for item ranking in collaborative filtering. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 301–306. Springer, Heidelberg (2011)
5. Buckley, C.: Topic prediction based on comparative retrieval rankings. In: 27th ACM Conference on Research and Development in Information Retrieval, pp. 506–507. ACM Press, New York (2004)
6. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: 25th ACM Conference on Research and Development in Information Retrieval, pp. 299–306. ACM Press, New York (2002)
7. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: A framework for selective query expansion. In: 13th ACM Conference on Information and Knowledge Management, pp. 236–237. ACM Press, New York (2004)
8. Dang, V., Bendersky, M., Croft, W.B.: Learning to rank query reformulations. In: 33rd ACM Conference on Research and Development in Information Retrieval, pp. 807–808. ACM Press, New York (2010)
9. Hauff, C.: Predicting the Effectiveness of Queries and Retrieval Systems. PhD thesis, University of Twente, Enschede (2010)
10. Hauff, C., Hiemstra, D., de Jong, F.: A Survey of Pre-Retrieval Query Performance Predictors. In: 17th ACM Conference on Information and Knowledge Management, pp. 439–448. ACM Press, New York (2008)
11. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 43–54. Springer, Heidelberg (2004)
12. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: 22nd ACM Conference on Research and Development in Information Retrieval, pp. 230–237. ACM Press, New York (1999)
13. Hofmann, T.: Latent semantic models for Collaborative filtering. ACM Trans. Inf. Syst. 22(1), 89–115 (2004)
14. Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty. In: ACM SIGIR Workshop on Predicting Query Difficulty – Methods and Applications (2005)

15. Pérez-Iglesias, J., Araujo, L.: Ranking List Dispersion as a Query Performance Predictor. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 371–374. Springer, Heidelberg (2009)
16. Plachouras, V., He, B., Ounis, I.: University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In: 13th Text Retrieval Conference, Gaithesburg, Maryland (2004)
17. Shtok, A., Kurland, O., Carmel, D.: Predicting Query Performance by Query-Drift Estimation. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 305–312. Springer, Heidelberg (2009)
18. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unified relevance models for rating prediction in collaborative filtering. ACM Trans. Inf. Syst. 26(3), 1–42 (2008)
19. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: 28th ACM Conference on Research and Development in Information Retrieval, pp. 512–519. ACM Press, New York (2005)
20. Zhou, Y., Croft, B.W.: Query performance prediction in web search environments. In: 30th ACM Conference on Research and Development in Information Retrieval, pp. 543–550. ACM Press, New York (2007)

Trading Spaces: On the Lore and Limitations of Latent Semantic Analysis

Eduard Hoenkamp

Queensland University of Technology, Brisbane, Australia
hoenkamp@acm.org

Abstract. Two decades after its inception, *Latent Semantic Analysis* (LSA) has become part and parcel of every modern introduction to IR. For any tool that matures so quickly, it is important to check its lore and limitations, or else stagnation will set in. We focus here on the three main aspects of LSA that are well accepted, and the gist of which can be summarized as follows: (1) that LSA recovers latent *semantic factors* underlying the document space, (2) that such can be accomplished through lossy compression of the document space by eliminating *lexical noise*, and (3) that the latter can best be achieved by *Singular Value Decomposition*.

For each aspect we performed experiments analogous to those reported in the LSA literature and compared the evidence brought to bear in each case. On the negative side, we show that the above claims about LSA are much more limited than commonly believed. Even a simple example may show that LSA does not recover the optimal semantic factors as intended in the pedagogical example used in many LSA publications. Additionally, and remarkably deviating from LSA lore, LSA does not scale up well: the larger the document space, the more unlikely that LSA recovers an optimal set of semantic factors. On the positive side, we describe new algorithms to replace LSA (and more recent alternatives as pLSA, LDA, and kernel methods) by trading its l_2 space for an l_1 space, thereby guaranteeing an optimal set of semantic factors. These algorithms seem to salvage the spirit of LSA as we think it was initially conceived.

1 Introduction

When users search for on-line documents they are usually looking for content, not words. So it is at least remarkable that the user's information need can be satisfied with search results based on keywords. This may stem from the user's ability to quickly learn to formulate an effective query, and the possibility to refine it. Or perhaps it is due to statistical properties of large corpora. Yet, most IR researchers would agree that trying to target the semantics underlying documents more directly could lead to better search results. The issue has become more and more acute in recent years, where the swelling amount of multi-media available, so far defies effective indexing techniques other than through textual annotation. The need to address underlying meaning, however, has been known

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 40–51, 2011.

in IR for decades. A good technique for this would obviate or circumvent the lexicon problem (the influence of synonymy and polysemy). So several techniques had been proposed early on, such as using hand-crafted domain models and thesauri, most notably WordNet [22]. This paper is about the technique that has been around for two decades, is not labor intensive, and has become part of the toolbox for every aspiring IR researcher: Latent Semantic Analysis (LSA). And as with every technique that has matured enough to enter the text books, the time has come to evaluate its lore and limitations.

2 How 'Semantic' Is LSA?

For people studying natural language processing it is often important, useful, or necessary to distinguish between meaning and language. The latter is a vehicle to express meaning, a way to convey thoughts, denote concepts. But while there is little dispute that the two should be distinguished, there is much disagreement about the nature of semantics, and especially how to represent it. Some disciplines are more rigorous than others in representing semantics formally, as testified by any textbook in logic, category theory, programming languages, or artificial intelligence. One may disagree with any particular definition, but at least there is a tangible entity to disagree about. An issue we had in studying LSA is that a precise definition of the word 'semantic' in the expression 'Latent Semantic Analysis' could not be found in the LSA literature. There is a caveat in the first footnote in [6] stating that semantic "implies only the fact that terms in a document may be taken as referents to the document itself or to its topic" which begs the question what 'topic' is. And in later papers, most notably [19] we find that the "LSA representation of the passages must also be related to the overall inferred meaning." Unfortunately, this makes the term 'semantic' immune for dispute, and so we will have to make do with the vaguer notion of something underlying a language utterance, or topic of a passage, or inferred meaning.

2.1 US Patent No. 4,839,853

For a description of LSA one could use any of the papers by its originators, yet we thought it safest to use the patent application for it from 1989 [5]. Since it concerns a patent, it must by definition be the most accurate and unambiguous description. The problem addressed by the patent is that "people want to access information based on meaning, but the words they select do not adequately express intended meaning." The patent then proposes to circumvent this problem by "treating the unreliability of observed word-to-text object association data as a statistical problem." One could argue that LSA might accomplish this without targeting or relying on underlying semantics (especially in light of the care taken not to define 'semantic', see the quotations in the previous paragraph). Nonetheless this is the lore we observed in many sources that explain LSA. So next what we will do is take a closer look at how LSA relates to underlying meaning of

documents. This will be done using a comparison with publications by other authors who do explicitly use the term 'semantic' as a term for underlying meaning of words and text documents.

2.2 Semantics and Cross-Language Retrieval

The lore of Latent Semantic Analysis that we have found over and over again, is that it targets semantics, the underlying concepts that are communicated in a document. So we wanted to further investigate whether LSA is indeed successful because it handles semantics, as the name suggests, or because it excels in statistical sophistication. As indirect proof of the former that stands out in the LSA literature are experiments in cross-language information retrieval (CLIR) as reported in e.g. [26] and [21]. More recent CLIR experiments that improve on LSA by adding kernel methods (e.g. [23]) make no such claims about semantics. Yet, if there is one thing that should be invariant under translation of a text, then it must be its meaning. So success in CLIR experiments could indeed be a sign that LSA operates on the level of meaning as opposed to the word level. There is a different approach to retrieval that tries to incorporate semantic relationships in the corpus, but which does this explicitly, named the 'Hyperspace Analog to Language' (HAL). We previously published a study that compared the two approaches in how they fare in cross language retrieval [12], so for the present paper we need only briefly describe the method and the conclusion of that study. Recall that CLIR experiments in the literature have used multilingual, document-aligned corpora, where documents in one language are paired with their translation in the other. In our study we developed a technique we called 'fingerprinting' in analogy to DNA fingerprinting. Imagine the documents of one language stacked on a pile, next to a pile that has the translations in the same order as the original. For a given query, a search technique will assign relevance weights to the documents. These weights can be expressed as a grayscale for each document, from black (not relevant) to white (highly relevant). The pile with original documents will show bands reminiscent of the bands in a DNA fingerprint. If the search technique is invariant under translation, than the bands in the piles should be in the same place. The less invariant, the fewer bands the piles will have in common. Instead of paired corpora, we used two paired translations: Hemingway's "The old man and the sea" with its translation in German and Italian, and Hawking's "A brief history of time" also with German and Italian translations. We used the book as a corpus, with passages in it as documents to be retrieved. The comparison between LSA and HAL (or rather our 'ergodic process interpretation' epi-HAL) was measured as the correlation between the fingerprints. As queries we used every passage contained in the book, so we expected to find at least that passage, and possibly related passages. HAL gave an average correlation over all these queries of around 99%, whereas LSA scored barely 70% maximum (the average was lower). In brief, we found that HAL was considerably more invariant under translation than LSA. Note however, that this does *not* show that one or the other derives its results from being based on semantics, because either technique could be conducive to

translation invariance for other reasons. It does show however that if there is a claim that the search technique is based on underlying semantics, then HAL is much more justified to this claim than LSA. But with so glaring a difference in performance, the lore that LSA targets underlying semantics certainly becomes dubious.

2.3 The Notion of 'Lexical Noise' in LSA

In the LSA literature the term 'noise' is used in at least two different ways. One refers to the 'lexicon problem' that arises from synonymy and polysemy. In case a word in a document is used where its synonym could have been used instead, is treated as a (seemingly) random event. In the case of one word with several distinct meanings (polysemy) the IR system could pick up the wrong concept, also as an almost random event. The other use of the term 'noise' is used to explain errors that originate from the LSA technique itself, as explained in the patent: "If the number of dimensions is too large, random noise or variations in word usage will be remodeled." This obviously cannot apply to the pedagogical example in the patent application as it uses only two dimensions. But perhaps 'noise' refers only to the first meaning. Since the LSA technique produces a splendid partition of the sentences in the example (whereas the correlations before LSA was very low) this is taken as evidence that noise was removed. Yet, this could be a spurious effect, not withstanding the evidence brought about in many LSA publications. We can illustrate this with a simple example. We simulated random noise in the data (the sentences) by drawing from a uniform

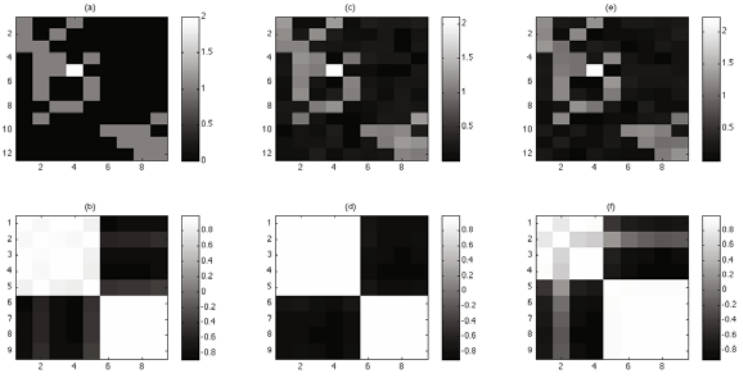


Fig. 1. The influence of 'noise' on the performance of LSA. (a) the original term by document matrix which appears in many LSA publications represented in gray scale. (c) and (e) is the original data with two samples of iid uniform noise added to (a). In the bottom row are the the correlations between documents after LSA. The same noise conditions may produce a better separation of the documents as (d) or worse as in (f) where one of the documents is assigned the wrong classification. The separation observed in (b) may hence be spurious.

distribution and adding it to the original term by document matrix. If LSA would work as contended, namely by removing lexical noise, than each time it should produce the same grouping of documents. Figure 1 shows that LSA may produce a different grouping for independent but identical noise samples. The result from the pedagogical example may therefore be spurious. The result was to be expected: SVD is known to often produce spurious results (or bottom effects) for categorial data. And these are categorial data, as the words are either present or absent.

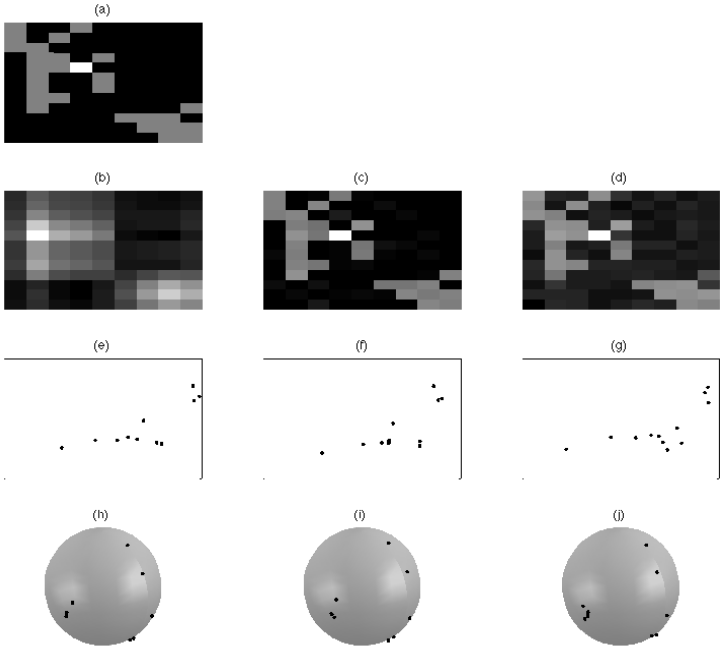


Fig. 2. Dimension reduction as used in LSA is just one example of lossy compression, for which legion algorithms can be found in the signal processing literature. The figure compares several of those: SVD, JPEG, and the Haar wavelet. (a) The word-by-document matrix from the patent as a gray-scale image. (b) after SVD, truncation at the three greatest eigenvalues, and inverse transform. (c) as (b) for JPEG (with quality .75), and (d) for Haar (coefficients below .3 ignored). (e), (f), and (g) show the word vectors projected on the two highest factors. (h), (i), and (j) show the documents projected on a unit-sphere in the three highest factors. The documents separated: human-computer interaction went to the left hemisphere, the graph-theory to the right. The alternatives achieve the same or better separation than SVD but much more efficiently.

2.4 SVD for LSA?

Other lore we found in the literature is the equating of LSA with SVD. No wonder, in virtually every chapter on LSA we have seen, a picture explaining

SVD stands out. But LSA is based on an old technique introduced by Eckart and Young in 1936 [8] who used singular value decomposition (SVD) to discover underlying factors in psychological data, represented as a matrix of subjects by observations. They proposed to reduce that matrix to one of a lower rank by applying SVD and ignoring the smallest singular values. That technique found its way in many areas where data reduction was sought, or signals had to be de-correlated. LSA uses the same technique, but where Eckart and Young used a matrix of subject by observations, LSA used the term by document matrix, i.e. a representation of the 'document space' as introduced by Salton [25]. The technique is an example of a broader set of methods to achieve so called 'lossy compression'. In [10] I described what it is that LSA tries to achieve using SVD, and several methods that could achieve the same but that are computationally much more efficient. That paper introduces the metaphor of the term by document matrix as a picture. Figure 2 shows several examples from that article. In contrast to SVD (which has complexity at least square in the number of documents) for example the Haar transform can be computed in linear time and constant space. So the lore of equating SVD with LSA, puts a limitation on LSA. Combinations of using lossy compression for dimension reduction and SVD to select underlying factors can be found e.g. in [17]. What all the techniques have in common is that no criterion suggests itself of how much dimension reduction is required in the case of LSA. If LSA really uncovers semantic factors, how many are needed? The LSA literature suggests between 100 [5] and 300 [19]. Not knowing what and how to choose the number of underlying factors is a limitation of LSA. Another limitation is the bottom effect for categorial data, we already spoke about. And finally, the fact that SVD produces an optimal subspace assumes that the noise is normally distributed. This is a plausible assumption for psychological data and in the area of signal processing. But since there is no such definition of noise in the LSA literature, there must be better ways to find underlying semantic factors. This is the topic of the final section of this paper. But before we finish, let us briefly look at other approaches to finding semantic factors: probabilistic LSA and Latent Dirichlet Allocation.

2.5 US Patent No. 6,687,696, pLSI, and LDA

LSA was developed for the vector space model of IR. A noteworthy alternative for LSA from the other IR paradigm, language modeling, is probabilistic LSI [15]. It is also covered by a US patent [16], and which like LSA searches for semantic factors underlying documents. It similarly can deal with synonymy and polysemy (the lexicon problem), an indication that it can target underlying meaning. It is a probabilistic model, where the semantic factors are represented by a probability distribution over a fixed set of 'topics'. For small document sets it improved over LSA as measured by precision and recall, yet it seems that the number of topics to be chosen is rather arbitrary. We will not go into the limitations of the technique because a later proposal was able to overcome most of these [3]. That later proposal was Latent Dirichlet Allocation (LDA) where again the distribution of 'topics' is taken as representation of documents.

Finally, pLSI and LDA can be shown to be equivalent under plausible simplifying assumptions [9]. Other proposals are based on kernel methods (e.g. [23]). None of these references clarify or even begin to define the nature of the underlying topics, or put any restrictions on the number of topics to represent a given corpus. The value of the techniques is measured mainly by how well they split documents into clusters that make sense. So we may soon see yet another technique that shows how the best result it achieves improves on than that of others who previously published their best results. That is good and valid IR obviously, especially from an engineering standpoint. Yet, when we as humans read a document, or a set of documents, we can intuit the topic or the number of topics being conveyed. So, can topics really be chosen arbitrarily? Is any choice and number of semantic factors as good as the next? As we have argued in other publications, models in IR are often overly general: they approach the material as raw data without concern of how they were produced. Instead, we have shown repeatedly that taking the nature of the data into account, namely that documents are produced by people, will lead to better results [13,14,11]. The remainder of this paper will take that orientation with regard to semantic factors in order to avoid the limitations of LSA and its cousins. We start with very general observations about documents, terms, and underlying concepts, and develop the theoretical framework from there.

3 Locating Semantic Factors

We begin with a few observations about the number of words and the number of semantic factors. Landauer and Dumais, for example, find experimentally that 300 semantic factors are about right (see figure 3 in [19]). But even without experiments some relationships between number of words and number of semantic factors can be observed. Let us temporarily use 'concepts' instead of 'factors' to avoid confusion with the mathematical techniques to find them:

Observation 1. The number of concepts conveyed in a document is only a small subset of all possible concepts,

Observation 2. The number of words in a document is also much smaller than the number of all possible concepts,

Observation 3. The number of words in a document is generally much greater than the number of concepts that it conveys,

Similar simple everyday observations lead to our theory of epi-HAL mentioned in section 2.2, and we shall see how the above observations will lead to an interesting theory about semantic factors as well.

The document space model of IR describes documents as a vector space. Let us now assume that the concepts also form a vector space. This can be backed up by the many cognitive theories that have been advanced about the structure of such a space, variously known by such names as semantic space and conceptual space as alluded to in section 2. We assume the geometry of the semantic space similar to that used in LSA, i.e. an n-dimensional space of presumably n elementary semantic factors as coordinates.

Suppose now a writer expresses the concept x in the n-dimensional semantic space as a document y of dimension m (in the document space). To make headway, assume that this is a linear operation, denoted by A, so that $y = Ax$. From observation 1 above it is clear that most coordinates of x must be zero, i.e. x is sparse. From observation 2 it follows that $m \ll n$. Finally, if x contains k non-zeros, than from observation 3 it follows $m \gg k$.

Now, given a y in the document space, approaches like LSA and HAL try to locate the concepts underlying this document. As A can be represented by an $m \times n$ matrix with $m \ll n$, the system $y = Ax$ is underdetermined, and for a given document y there are infinitely many solutions for x. In terms of concepts it means that given a set of documents, there may be many different combinations of concepts that could be expressed as the given documents. So which ones to choose? LSA has an outspoken preference for particular solutions, which we will discuss in the next section together with alternative preferences one might have.

3.1 Parsimonious Alternatives to LSA

LSA and its recent alternatives try, for a given set of documents, to compute a subspace of the semantic space from which the documents can be produced. Based on SVD, LSA will produce a subspace closest to the original in terms of euclidian distance, that is, according to the l_2 norm[1] (least squares). The dimensions uncovered this way are the 'latent semantic factors.' As opposed to HAL there is no a priori notion of semantics, which is simply defined by the SVD procedure for a dimension set by the researcher. However, there is little advice in the literature on how to choose the number of factors[2], and hence no advice on which one of the infinite number of solutions for x would be most appropriate. We do know however that x is sparse (observation 1 above). And adopting the good principle of parsimony, we opt to take the sparsest solution as the preferred one. That is, we don't want to postulate many semantic factors if fewer will do. Interestingly, even if we relax $m \gg k$ to $m \geq 2k$, then for a full-rank A the sparse solution of $y = Ax$ is unique, and all other solutions are not sparse[3]. In principle one could exhaustively try all combinations of coordinates of x and take the one with the smallest number of non-zeros. This is the same as minimizing $\sum_i |x_i|^0$, in other words the l_0 norm of x. But that problem is equivalent to the sub-set sum problem and hence in NP, which makes such a solution infeasible. To proceed we need a fourth observation about the relation between the document space and the semantic space.

[1] The p-norm l_p of x or $||x||_p$ is defined as $||x||_p^p = \sum_i |x_i|^p$. So $||x||_2$ is the euclidian distance, $||x||_1$ the city block metric, and $||x||_0$ the number of non-zeros in x.

[2] To quote the patent: "The number of dimensions to represent adequately a particular domain is largely an empirical matter."

[3] Proof: assume another sparse solution u exists. Then $Ax = Au$ hence $A(x - u) = 0$, where $x - u \neq 0$ would have at most m non-zeros. But then $m \geq 2k$ contradicts that A is full-rank.

Observation 4. Documents that are near in the document space have underlying concepts that are near in the semantic space.

This property is mostly implicit in the LSA literature, but sometimes explicit as in [20] where SVD is used to uncover those distances (as correlations). The observation is important, because in cases where such near-isometry exists between spaces, the l_1 minimization of x for $y = Ax$ finds the same sparse solution as l_0 [4]. And, while minimizating l_0 is in NP, l_1 minimization can be cast as a linear programming problem, for which many tractable solutions exist. We will not elaborate these techniques here, as the theory about l_1 minimization for finding sparse x solutions has yielded a new discipline by itself in the area of signal processing. There is no principled way that LSA chooses the number of semantic factors, whereas we start from a principle of parsimony to arrive at a unique solution with the minimum number of semantic factors. Yet, they could still lead to the exact same solutions as LSA. This is extremely unlikely however, and we will illustrate why, by comparing the solutions of l_2 minimization (LSA) with that of l_1 norm minimization. So we will refer to that literature for practical details and downloadable implementations [4,7,24]. The precise conditions under which these algorithms will work is governed by the Johnson-Lindenstrauss lemma [18,1] of which observation 4 is an example in everyday language. This section could be perceived as just a mathematical promenade that accomplishes

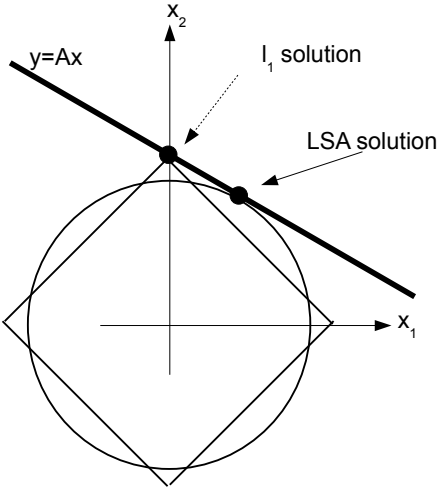


Fig. 3. The difference in solutions for $y = Ax$ visualized in 2D. The black line indicates the solution space. LSA solutions minimize l_2 hence lie on the circle, while the l_1 solutions lie on the square. As in this picture, l_2 solutions are almost never sparse, whereas l_1 solutions are unique and almost always sparse, with higher probability the higher the dimension.

the same as LSA. Of course we would not have told the whole story if that were the case. So next we will show where the two part company.

3.2 How l_0 and l_1 Norms Are Better Than LSA

The previous section approached the discovering of semantic factors not as a statistical problem as LSA sees it (cf. 2.1), but as a linear algebra problem.

Note first that if x is a solution of $y = Ax$, then so is every vector in the null-space of A translated over x. Recall that the null-space of A are the vectors u for which $Au = 0$, so $y = A(x) = A(x) + A(u) = A(x + u)$ hence any point in u translated over x is also a solution. The null-space of A are all those concepts not expressed in the document y. This is another, trivial, reason for wanting the sparsest solution. By definition, points for which $||x||_p = c$ for constant c lie on the surface of $\sum_i |x_i|^p = c$. For $p = 2$ this is a hypersphere and for $p = 1$ a polytope. Figure 3 visualizes in 2D how the LSA approach, which is based on l_2 minimization, differs from l_1 minimization.

A final remark about noise. As we mentioned in section 2.3, the term 'noise' in the context of LSA is used metaphorically. We can imagine that if a speaker or writer expresses concepts, that process is under the influence of noise in the technical sense as in 'noisy channel'. In other words, we have $y = Ax + \epsilon$ where ϵ stands for the noise introduced in the translation from concepts to language. In that case we can can still look for the sparsest solution, but reintroduce l_2 to minimize the noise needed to explain the model. That is we minimize $||x||_1$ adding the condition to minimize $||\epsilon||_2$. Again, for more details about this and practical applications outside of IR we refer to the vast literature on compressive sensing, and especially [2] for a brief introduction.

This last section could only introduce the work that we are currently undertaking, but we are sure that it can show the direction of a more principled way of discovering 'latent semantic factors' that removes the limitations of the current definition of LSA.

4 Conclusion

We have looked at several aspects of Latent Semantic Analysis that have found their way into many introductions and tutorials on information retrieval. The lore we found was in the definition of semantics, the role of noise, and the identification of LSA with SVD. At the same time, we presented a new direction by replacing l_2 minimization as used by most current dimension reduction techniques, most notably SVD, by l_1 minimization. We are confident that our current research can remove the limitations of LSA as discussed above, while preserving the spirit of what LSA tries to achieve.

References

1. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constructive Approximation 28(3), 253–263 (2008)

2. Baraniuk, R.G.: Compressive Sensing. IEEE Signal Processing Magazine 24(118-120,124) (July 2007)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
4. Candès, E.J., Tao, T.: Decoding by linear programming. IEEE Transasctions on Information Theory 51, 4203–4215 (2005)
5. Deerwester, S.C., Dumais, S.T., Furnas, G.W., Harshman, R.A., Landauer, T.K., Lochbaum, K.E., Streeter, L.A.: U.S. Patent No. 4,839,853. U.S. Patent and Trademark Office, Washington, DC (June 1989)
6. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
7. Donoho, D.L.: Compressed Sensing. IEEE Transasctions on Information Theory 52, 1289–1306 (2006)
8. Eckart, G., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika 1, 211–218 (1936)
9. Girolami, M., Kaban, A.: On an equivalence between pLSI and LDA. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 433–434 (2003)
10. Hoenkamp, E.: Unitary operators on the document space. Journal of the American Society for Information Science and Technology 54(4), 314–320 (2003)
11. Hoenkamp, E., Bruza, P., Song, D., Huang, Q.: An effective approach to verbose queries using a limited dependencies language model. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 116–127. Springer, Heidelberg (2009)
12. Hoenkamp, E., van Dijk, S.: A fingerprinting technique for evaluating semantics based indexing. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 397–406. Springer, Heidelberg (2006)
13. Hoenkamp, E., Song, D.: The document as an ergodic markov chain. In: Proceedings of the 27th Conference on Research and Development in Information Retrieval, pp. 496–497 (2004)
14. Hoenkamp, E.: Why information retrieval needs cognitive science: A call to arms. In: Proceedings of the 27th Annual Conference of the Cognitive Science Society, pp. 965–970 (2005)
15. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR Forum Special issue, pp. 50–57. ACM, New York (1999)
16. Hofmann, T., Christian, J.: U.S. Patent No. 6,687,696. U.S. Patent and Trademark Office, Washington, DC (February 1989)
17. Jaber, T., Amira, A., Milligan, P.: TDM modeling and evaluation of different domain transforms for LSI. Neurocomputing 72(10-12), 2406–2417 (2009); Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007)
18. Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics 26, 189–206 (1984)
19. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104(2), 211–240 (1997)

20. Landauer, T.K., Laham, D., Rehder, B., Schreiner, M.E.: How well can passage meaning be derived without using word order: A comparison of latent semantic analysis and humans. In: Proc. of the 19th Annual Meeting of the Cognitive Science Society, pp. 412–417. Erlbaum, Mahwah (1991)
21. Littman, M., Dumais, S.T., Landauer, T.K.: Automatic cross-language information retrieval using latent semantic indexing. In: Cross-Language Information Retrieval, ch. 5, pp. 51–62. Kluwer Academic Publishers, Dordrecht (1998)
22. Miller, G.: WordNet: A lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
23. Park, L.A.F., Ramamohanarao, K.: Kernel latent semantic analysis using an information retrieval based kernel. In: International Conference on Information and Knowledge Management, pp. 1721–1724 (2009)
24. Baraniuk, R.G., Wakin, M.B.: Random projections of smooth manifolds. Foundations of Computational Mathematics 9, 65–74 (2009)
25. Salton, G.: Automatic Information Organization and Retrieval. McGraw-Hill, New York (1968)
26. Yang, Y., Carbonell, J.G., Brown, R.D., Frederking, R.E.: Translingual information retrieval: Learning from bilingual corpora. Artificial Intelligence 103(1-2), 323–345 (1998)

Appendix

Example from US Patent No. 4,839,853

For ease of reference we reproduce the example of the LSA patent here. It uses the following sentences to stand for documents:

c1 *Human* machine *interface* for ABC *computer* applications
c2 A *survey* of *user* opinion of *computer system response time*
c3 The *EPS user interface* management *system*
c4 *System* and *human system* engineering testing of *EPS*
c5 Relation of *user* perceived *response time* to error measurement
m1 The generation of random, binary, ordered *trees*
m2 The intersection *graph* of paths in *trees*
m3 *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4 *Graph minors*: A *survey*

Words that occur in at least two titles (italicized) are entered in the term by document matrix, which was depicted in gray-scale in figure 2a.

Quantum Latent Semantic Analysis

Fabio A. González and Juan C. Caicedo

Bioingenium Research Group
Computing Systems and Industrial Engineering Dept.
National University of Colombia
{fagonzalezo,jccaicedoru}@unal.edu.co

Abstract. The main goal of this paper is to explore latent topic analysis (LTA), in the context of quantum information retrieval. LTA is a valuable technique for document analysis and representation, which has been extensively used in information retrieval and machine learning. Different LTA techniques have been proposed, some based on geometrical modeling (such as latent semantic analysis, LSA) and others based on a strong statistical foundation. However, these two different approaches are not usually mixed. Quantum information retrieval has the remarkable virtue of combining both geometry and probability in a common principled framework. We built on this quantum framework to propose a new LTA method, which has a clear geometrical motivation but also supports a well-founded probabilistic interpretation. An initial exploratory experimentation was performed on three standard data sets. The results show that the proposed method outperforms LSA on two of the three datasets. These results suggests that the quantum-motivated representation is an alternative for geometrical latent topic modeling worthy of further exploration.

Keywords: quantum mechanics, quantum information retrieval, latent semantic analysis, latent semantic indexing, latent topic analysis, singular value decomposition, probabilistic latent semantic analysis.

1 Introduction

Since its inception, latent topic analysis[1] (LTA) (also known as latent semantic analysis/indexing) has been a valuable technique for document analysis and representation in both information retrieval (IR) and machine learning. The main assumption behind LTA is that the observed term-document association may be explained by an underlying latent topic structure. Different methods for latent topic analysis have been proposed, the most prominent include: latent semantic analysis (LSA) [2], probabilistic latent semantic analysis (PLSA) [4], and latent Dirichlet allocation (LDA) [1]. LSA was the first latent analysis method proposed

[1] For the remaining part of the text we will use the term *latent topic analysis* to allude the general modeling strategy avoiding confusion with *latent semantic analysis* which refers to the particular method.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 52–63, 2011.

and its approach is geometrical in nature, while PLSA and LDA have a sound probabilistic foundation.

Quantum information retrieval (QIR) [12,10], is a relatively new research area that attempts to provide a foundation for information retrieval building on the mathematical framework that supports the formulation of quantum mechanics (QM). QIR assimilates the traditional vector space representation to Hilbert spaces, the fundamental concept in QM. Notions such as system state, measurement, uncertainty and superposition are interpreted in the context of IR. QIR is been actively researched and some results suggest that it can go beyond an interesting analogy to become a valuable theoretical and methodological framework for IR [10].

The main goal of this paper is to explore latent topic analysis in the context of QIR. Same as in the vector space model, QIR represents documents/queries as vectors in a vector space (more precisely, a Hilbert space), however, QIR exploits the subspace structure of the Hilbert space and corresponding probability measures to define important IR notions, such as relevance, in a principled way [12]. A question that emerges is whether the richer QIR representation could provide new insights into the latent topic analysis problem. One important motivation for this question is the fact that QIR naturally combines both geometry and probability. Latent topic analysis methods proposed so far are either geometrical or probabilistic in nature, but not both. A quantum-motivated latent semantic analysis method could potentially combine both perspectives.

Some works in QIR [3,9,13,8] have already suggested the relationship between LTA and a quantum-based representation of documents. Up to our knowledge, there has not been proposed yet an original LTA algorithm in a quantum representation context. The work of Melucci [8] probably is the closest one to the work presented in this paper. In that work, a framework for modelling contexts in information retrieval is presented. The framework uses both a quantum representation of documents and LSA to model latent contexts, but do not propose a new LTA method.

This paper proposes a new LTA method, quantum latent semantic analysis (QLSA). The method starts from a quantum-motivated representation of a document set in a Hilbert space H. The latent topic space is modeled as a sub-space of H, where the document set is projected. The method is analysed from geometrical and probabilistic points of view, and compared with LSA and PLSA. An exploratory experimentation was performed to evaluate how the quantum-motivated representation impacts the performance of the method. The results show that the method outperforms LSA on two of the three datasets, and we hypothesize that it is due to an improved quantum representation.

The paper is organized as follows: Section 2 provides a brief overview of quantum information retrieval; Section 3 describes the method and discusses its similarities and differences with LSA and PLSA; Section 3 covers the exploratory experimental evaluation of the method; finally, Section 4 presents some conclusions and the future work.

2 Quantum Information Retrieval

QIR provides an alternative foundation for information retrieval. The main ideas were initially proposed by Van Rijsbergen [12], and different subsequent works have contributed to the continuous development of the area. The main idea in QIR is to use the quantum mechanics formalism to deal with fundamental information retrieval concepts exploting clear analogies between both areas. For instance, a quantum system state is represented by a wave function, which can be seem as a finite or infinite complex vector indexed by a continuous or discrete variable (usually representing space or momentum). In a vector space model, documents are represented by vectors, but in this case finite real vectors indexed by a discrete variable that represents text terms. In the next paragraphs we will briefly present some basic concepts from QIR that are necessary to introduce the proposed method.

Lets $D = \{d_i\}_{i=1\ldots n}$ be a set of documents, $T = \{t_j\}_{j=1\ldots m}$ be a set of terms, and $TD = \{td_{ji}\}$ be the corresponding term-document matrix. The quantum representation of a document d_i is given by a wave function φ_i defined by:

$$\varphi_i(j) = \sqrt{\frac{td_{ji}}{\sum_{j=1}^{m} td_{ji}}}, \text{ for all } j = 1 \ldots m,$$

This representation has the following convenient properties:

$$\forall i, \|\varphi_i\| = 1$$

$$< \varphi_i, \tau_j >^2 = P(t_j|d_i) \tag{1}$$

where $< \cdot, \cdot >$ is the dot product operator, τ_j is the wave function of the term t_j corresponding to a unitary vector with a one in the j-th position. This representation corresponds in fact to a representation of the documents in the term space, which we will call H and whose basis is $\{|\tau_j\rangle\}_{j=1\ldots m}$.

Dirac notation is a convenient notation formalism extensively used in quantum mechanics. The two basic building blocks of Dirac notation are the *bra* and the *ket*, notated respectively as $\langle\varphi|$ and $|\beta\rangle$. A ket represents a vector in a Hilbert space and a bra a function from the Hilbert space to a real (or complex) space. The application of a bra to a ket coincides with the dot product of the corresponding vectors and is notated $\langle\varphi|\beta\rangle$. In a finite-dimensional Hilbert space, a bra may be seem as a row vector and a ket as a column vector, in this case the application of a bra to a ket would correspond to a conventional matrix multiplication.

A bra and a ket can be composed in a reverse way, $|\beta\rangle \langle\varphi|$, and this can be interpreted as the outer product of the corresponding vectors. This is useful, for instance, to define notions such as subspace projectors. A subspace is determined by a basis that generates it or by a projector operator that projects any vector in the space to the subspace. If the basis of a given subspace S is $\{\beta_1, \ldots, \beta_m\}$, the corresponding projector is $P_s = \sum_{i=1\ldots m} |\beta_i\rangle \langle\beta_i|$. Projectors with trace one

are called density operators and have an important role in quantum mechanics, they are used to represent the statistical state of a quantum system.

Using Dirac notation the second property in Eq. 1 can be expressed as $\langle\varphi_i|\tau_j\rangle^2 = P(t_j|d_i)$. This property can be interpreted, in a QIR context, as the density operator $\rho_i = |\varphi_i\rangle\langle\varphi_i|$ (corresponding to the document d_i) acting on the subspace $P_{\tau_j} = |\tau_j\rangle\langle\tau_j|$ (which is induced by the term t_j) according to the rule:

$$P(P_{\tau_j}|\rho_i) = \mathrm{tr}(\rho_i P_{\tau_j}) = \mathrm{tr}(|\varphi_i\rangle\langle\varphi_i|\tau_j\rangle\langle\tau_j|) = \langle\varphi_i|\tau_j\rangle^2,$$

where $tr(\cdot)$ is the matrix trace operator. The above procedure could be extended to more complex subspaces, i.e., with dimension higher than one.

3 Quantum Latent Semantic Analysis

In general, LTA modeling assumes that the high diversity of terms in a set of documents may be explained by the presence or absence of latent semantic topics in each document. This induces a new document representation where documents are projected to a latent topic space by calculating the relative degree of presence of each topic in each document. Since the set of latent semantic topics is usually one or two orders of magnitude smaller than the set of terms, the effective dimension of the latent topic space is smaller than the dimension of the original space, and the projection of the document to it is, in fact, a dimensionality reduction process.

A latent topic space is a subspace S of H defined implicitly by its projector as:

$$P_S = \sum_{k=1}^{r} |\sigma_k\rangle\langle\sigma_k|,$$

where $\{|\sigma_k\rangle\}_{k=1\ldots r}$ is an orthonormal basis of the sub-space S and each $|\sigma_k\rangle$ corresponds to the wave function of a latent topic z_k. A projection of a document represented by $|\varphi_i\rangle$ on the latent space is given by:

$$|\bar{\varphi}_i\rangle = P_S|\varphi_i\rangle.$$

From a quantum mechanics perspective, this projection can be interpreted as the measurement of the observable corresponding to S on the system state $|\varphi_i\rangle$. This measurement will make the state of the system collapse to a new state $|\hat{\varphi}_i\rangle = \frac{|\bar{\varphi}_i\rangle}{\||\bar{\varphi}_i\rangle\|}$. Accordingly, the conditional probability of latent topic z_k given a document d_i represented in the latent space can be calculated by:

$$P(z_k|d_i) = \langle\hat{\varphi}_i|\sigma_k\rangle^2 = \frac{\langle\varphi_i|\sigma_k\rangle^2}{\|P_S|\varphi_i\rangle\|^2}.$$

Now, the main problem is to find an appropriate latent semantic topic space S. This can be accomplished by imposing some conditions. In particular, we expect that the latent topic representation loses as few information as possible and be

Algorithm 1. Quantum latent semantic analysis

Quantum-LSA(TD,r)
$TD = \{td_{ij}\}$: term-document matrix with $i = 1 \dots m$ and $j = 1 \dots n$.
r: latent topic space dimension

1: Build the document wave function matrix $\Phi \in \mathbb{R}^{m \times n}$ setting

$$\Phi_{ij} = \sqrt{\frac{td_{ji}}{\sum_{j=1}^{m} td_{ji}}}$$

2: Perform a SVD of $\Phi = U \Sigma V^T$
3: Select the first r columns of U, $\{\sigma_1 \dots \sigma_r\}$, corresponding to the r principal Eigenvectors of $\Phi\Phi^T$.
4: Project each document wave function $|\varphi_i\rangle = \Phi_{\cdot i}$

$$|\bar{\varphi}_i\rangle = \sum_{k=1}^{r} \langle \varphi_i | \sigma_k \rangle \langle \sigma_k |$$

5: Normalize the vector

$$|\bar{\varphi}_i\rangle = \frac{|\bar{\varphi}_i\rangle}{\| |\bar{\varphi}_i\rangle \|}$$

6: The smoothed representation of a document d_i in the term space is given by

$$P(t_j | d_i) = \bar{\varphi}_i(j)^2$$

7: The document representation in the latent topic space is given by

$$P(z_k | d_i) = \langle \bar{\varphi}_i | \sigma_k \rangle^2$$

as compact as possible. This can be expressed trough the following optimization problem:

$$\min_{\substack{S \\ dim(S)=r}} \sum_{i=1}^{n} \| |\bar{\varphi}_i\rangle - |\varphi_i\rangle \|^2 = \min_{\substack{S \\ dim(S)=r}} \sum_{i=1}^{n} \| P_S |\varphi_i\rangle - |\varphi_i\rangle \|^2$$

This problem is solved by performing a singular value decomposition (SVD) on the matrix formed by the vectors corresponding to the wave functions of the documents in the document set. Specifically, a matrix where the i-th column corresponds to the ket $|\varphi_i\rangle$, $\Phi = [\varphi_1 \dots \varphi_n]$, with

$$\Phi = U \Sigma V^T,$$

its SVD decomposition. The columns of $U = [\sigma_1 \dots \sigma_r]$, correspond to the vectors of an orthonormal basis of the latent subspace S. The process is summarized in Algorithm 1.

3.1 QLSA vs. LSA

Both QLSA and LSA use SVD as the fundamental method to find the latent space. However, there is an important difference: LSA performs the SVD decomposition of the original term-document matrix, whereas QLSA decomposes the document wave function matrix, whose entries are proportional to the square root of the original term-document matrix. This makes QLSA a different method, since the decomposition is happening on a different representation space.

Both methods have a clear geometrical motivation, however QLSA has, in addition, a natural probabilistic interpretation. LSA produces a representation that may include negative values, this has been pointed as a negative characteristic of latent topic representations based on SVD [7,14], since a document may be represented by both the presence and the absence of terms or topics in it. QLSA, in contrast, always produces positive values when documents are mapped back to the term/topic space.

3.2 QLSA vs. PLSA

The approach followed by PLSA is quite different to the one of QLSA. PLSA has a strong statistical foundation that models documents as a mixture of term probabilities conditioned on a latent random variable [4]. The parameters of the model are estimated by a likelihood maximization process based on expectation maximization. The mixture calculated by PLSA induces a factorization of the original term-document matrix:

$$P(t_j|d_i) = \sum_{k=1}^{r} P(t_j|z_k)P(z_k|d_i), \tag{2}$$

where $P(t_j|z_k)$ codifies the latent topic vectors and $P(z_k|d_i)$ corresponds to the representation of documents on the latent space.

QLSA also induces a factorization, but of the matrix formed by the wave functions corresponding to the documents in the set. To illustrate this lets check how the wave function of a document d_i is codified by QLSA:

$$\begin{aligned} \varphi_i(j) &= \langle \tau_j | \varphi_i \rangle \\ &\approx \langle \tau_j | \hat{\varphi}_i \rangle \\ &= \frac{\langle \tau_j | \, P_S \, | \varphi_i \rangle}{\| P_S \, | \varphi_i \rangle \|} \\ &= \sum_{i=1}^{k} \langle \tau_j | \sigma_k \rangle \, \frac{\langle \sigma_k | \varphi_i \rangle}{\| P_S \, | \varphi_i \rangle \|} \end{aligned} \tag{3}$$

Eq. 3 induces a factorization of the document wave function matrix Φ into two matrices, one codifying the latent topic wave functions $|\sigma_k\rangle$ represented in the term space, and the other one representing the interaction between documents and latent topics.

Using 1 and 3 we can calculate the approximation of $P(t_j|d_i)$ generated by QLSA:

$$\begin{aligned} P(t_j|d_i) &\approx \left[\sum_{i=1}^{r} \langle \tau_j|\sigma_k\rangle \frac{\langle \sigma_k|\varphi_i\rangle}{\|P_S\,|\varphi_i\rangle\|}\right]^2 \\ &= \sum_{i=1}^{r} \langle \tau_j|\sigma_k\rangle^2 \frac{\langle \sigma_k|\varphi_i\rangle^2}{\|P_S\,|\varphi_i\rangle\|^2} + I_{ji} \\ &= \sum_{k=1}^{r} P(t_j|z_k)P(z_k|d_i) + I_{ji}, \end{aligned} \tag{4}$$

where $I_{ji} = \left[\sum_{k,l=1\ldots r,k\neq l} \langle \tau_j|\sigma_k\rangle \langle \sigma_k|\varphi_i\rangle \langle \tau_j|\sigma_l\rangle \langle \sigma_l|\varphi_i\rangle\right] / \|P_S\,|\varphi_i\rangle\|^2$. Checking 2 and 4 it is easy to see the difference between both approximations, QLSA adds the additional term I_{ji}. This term could be interpreted as an interference term [15].

4 Experimental Evaluation

In this section we perform an exploratory experimentation that evaluates the performance of QLSA against LSA. As discussed in Section 3.1, both methods share a common geometrical approach that finds a low-dimensional space using SVD. The main difference resides in the document representation used. Thus, the goal of the experimental evaluation is to establish the effect of the quantum representation when using a latent topic indexing strategy for document retrieval.

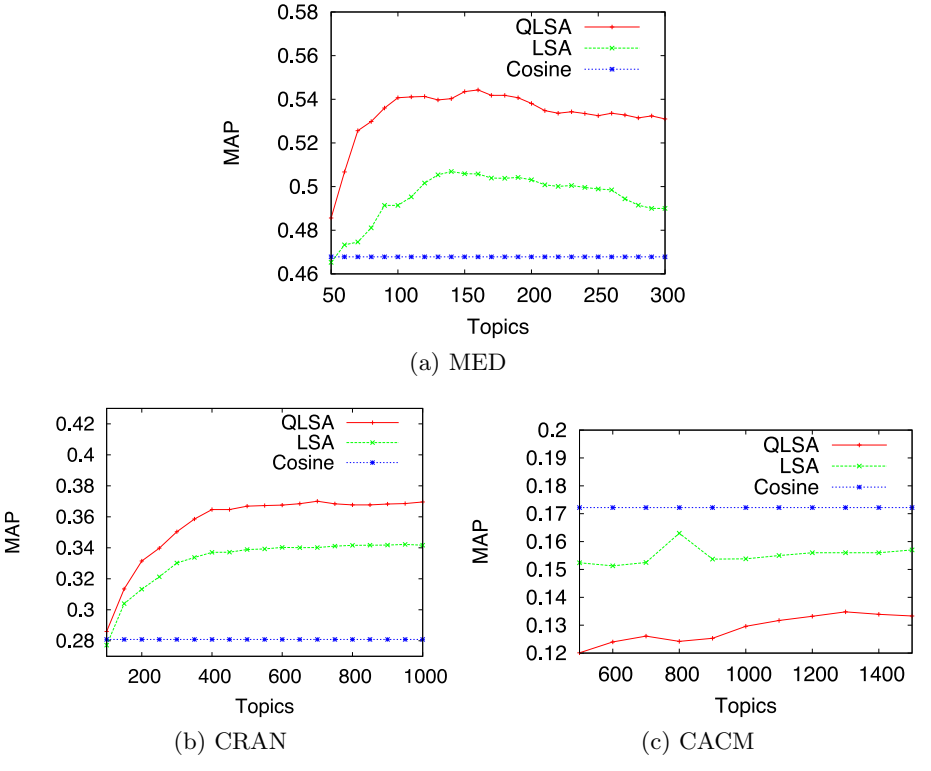
In our experiments we evaluated the automatic indexing task to support query based retrieval. The performance is measured in terms of Mean Average Precision (MAP) for two standard datasets to assess the empirical differences between the formulated method and two baseline approaches: direct matching in a Vector Space Model, using cosine similarity, and the LSA approach. The experimental setup is intentionally kept simple, only term frequency is used without any kind of weighting, simple stop-word removal and stemming preprocessing is applied. Document search is performed by projecting the query terms and using the cosine similarity with respect to other documents in the latent space, i.e., ranking scores are taken directly from the latent space.

4.1 Collections

To follow an evaluation of ranked retrieval, we used three collections with relevance assessment: (1) the MED collection, a common dataset used in early information retrieval evaluation, composed of 1033 medical abstracts and 30 queries, all indexed with about 7000 terms; (2) the CRAN collection, another standard dataset with 1400 document abstracts on aeronautics from the Cranfield institute of Technology and 225 queries, is indexed with about 3700 terms. (3) The CACM collection, with 3204 abstracts from the Communications of the ACM Journal with 64 queries, is indexed with about 3000 terms.

4.2 Dimensions of the Latent Space

Figure 1 presents the variation of MAP with respect to the number of latent factors for the evaluated collections. It shows that latent indexing methods provide an improvement over the cosine similarity baseline for the MED and CRAN collections. The dimension of the latent space was varied from 50 to 300 factors taking steps of 10 units for the MED collection and from 100 to 1000 factors taking steps of 50 units for the CRAN collection. The CACM collection, however, does not show improvements when using latent factors for document indexing.



(a) MED

(b) CRAN

(c) CACM

Fig. 1. Variation of number of topics for the different collections

For the first two collections, results show that QLSA performs better than LSA for every evaluated dimension of the latent topic space. In the MED collection, the performance of both methods increases to reach a maximum value around the same latent space dimensionality (between 140 and 160) and then starts to decrease slowly again. In the CRAN collection, the performance of both methods increases and tends to get stable after 500 topics. The best number of topics is very similar for both methods, however, the performance is significantly improved in favor of QLSA.

The CACM collection is particularly challenging for LSA, and QLSA does not perform better. In fact, QLSA seems to amplify the bad performance of LSA. In the case of LSA, this is consistent with previously reported performances in the literature, that showed no benefit for query based retrieval, but instead, a decreasing in performance.

4.3 Recall-Precision Evaluation

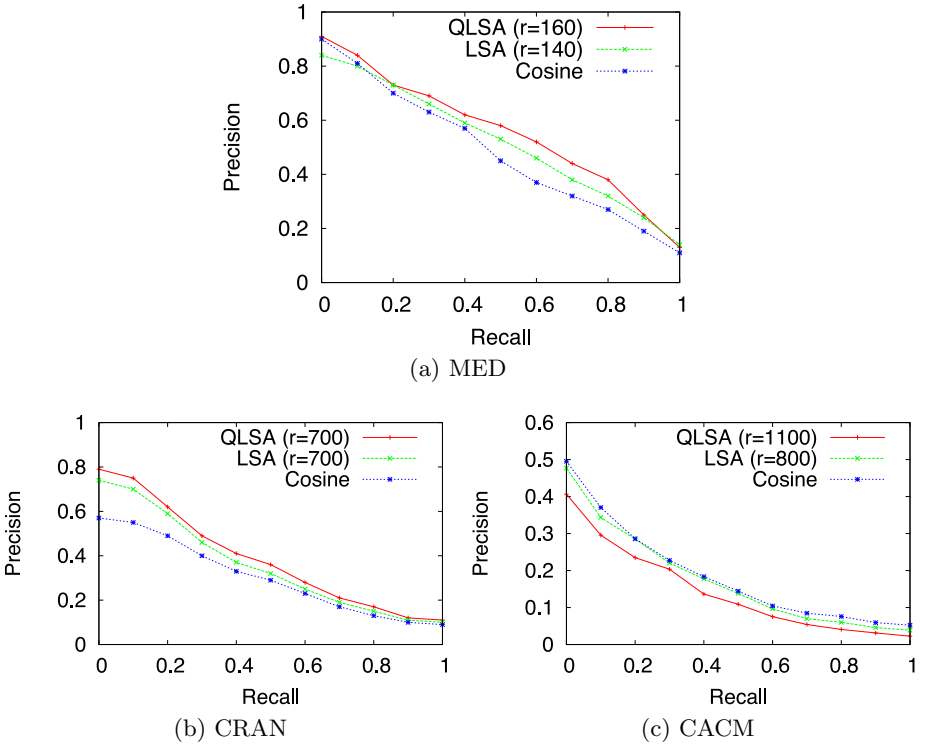
Figure 2 shows the interpolated Recall-Precision graph for the 3 evaluated approaches, averaged over the available set of queries. Each model has been configured with the best latent space dimensionality, according to the analysis on the previous Section. Again, results show that latent topic indexing provides a better response over the direct matching approach in the MED and CRAN collections. The plots also show an improved response of QLSA over both cosine and LSA approaches, in these two collections.

In the MED collection, QLSA provides a slightly better response with respect to the cosine similarity in the early stages of the retrieval process, and then starts to show a larger improvement. LSA starts worse than cosine but after the first part of the results it overtakes the baseline and shows a better response in the long term retrieval. QLSA presents a better response than LSA during the whole retrieval process. In the case of the CRAN collection, QLSA and LSA show a general improvement over the baseline, both in the early and long term retrieval. QLSA again offers better results than the other two methods, showing a consistent improvement in terms of precision for the ranked retrieval task.

Figure 2-c shows the response of the indexing methods on the CACM collection, showing an important decreasing for QLSA. We hypothesize that, for this collection, discriminative terms are mixed with other terms in latent factors, leading to a lose of discerning capacity of the ranking method.

Table 1 summarizes the results obtained in this exploratory evaluation, showing that QLSA results in an important improvement with respect to LSA for two collections even though both algorithms are based on a SVD. These results complement the theoretical differences between both algorithms and highlight the empirical benefits of using a QIR-based algorithm for modelling latent topics. In the case of the CACM collection, both LSA and QLSA show a decreasing in performance with respect to the baseline, with a larger margin for QLSA. It is interesting to see that when LSA performs better than the baseline, QLSA is able to outperform both, the baseline and LSA. But, when LSA does not improve, QLSA performs even worse.

A comparison against PLSA was not performed, however, the results reported by [4] could serve as a reference, despite they were obtained with a slightly different experimental setup that favors the performance of the algorithms. It reports an average precision of 63.9, 35.1 and 22.9 for MED, CRAN and CACM respectively, using PLSA. According to these results, QLSA does not outperforms PLSA, however, it shows a competitive performance on two of the datasets, on the other one the performance was remarkable bad.



(a) MED

(b) CRAN

(c) CACM

Fig. 2. Recall-Precision graphs for the three collections and three methods with the best latent factor dimensions in each case

Table 1. Summary of the retrieval performance on the test collections. Reported values are Mean Average Precision over all the available queries.

	MED		CRAN		*CACM*	
Method	*Precision*	*Improv.*	*Precision*	*Improv.*	*Precision*	*Improv.*
cosine	0.4678	-	0.2809	-	0.1722	-
LSA	0.5069	+8.36%	0.3302	+17.55%	0.1630	-5.34%
QLSA	0.5443	+16.35%	0.3504	+24.74%	0.1315	-23.64%

5 Discussion and Conclusions

Given its exploratory nature, the experimental results are not conclusive. However, the results are encouraging and suggest that the quantum representation could provide a good foundation for latent topic analysis. The approaches followed by both QLSA and LSA are very similar, the main difference is the document representation used. It is interesting to see the effect of the quantum representation on LSA performance: it improved the performance on two of the datasets where LSA showed some advantage over the baseline, but also it

amplified the bad performance on the other dataset. However, QLSA has a clear advantage over LSA, its more principled representation of the geometry of the document space allows a probabilistic interpretation.

LTA methods based on probabilistic modelling, such as PLSA and LDA, have shown better performance than geometry-based methods. However, with methods such as QLSA it is possible to bring the geometrical and the probabilistic approaches together. Here we started from a geometrical stand point to formulate the model and then we provided a probabilistic interpretation of it. Thanks to the dual nature of the quantum representation, it is possible to do exactly the opposite: start from a probabilistic latent topic model and then give it a geometrical interpretation. A good start point would be the theory of quantum probabilistic networks [11,5,6].

There are many remaining open questions that justify further investigation: what is the interpretation of the interference term (Eq. 4) in the approximation of $P(t_j|d_i)$ generated by QLSA? How to implement quantum versions of probabilistic LTA methods such as PLSA and LDA? These questions are the main focus of our ongoing research work.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3(4-5), 993–1022 (2003)
2. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
3. Di Buccio, E., Lalmas, M., Melucci, M.: From entities to geometry: Towards exploiting multiple sources to predict relevance. In: Proc. of the First Italian Information Retrieval Workshop, IIR (2010)
4. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI 1999, pp. 21–28 (1999)
5. La Mura, P., Swiatczak, L.: Markov entanglement networks. In: Proceedings of the AAAI Spring Symposium on Quantum Interaction (2007)
6. Laskey, K.B.: Quantum Causal Networks. In: Proceedings of the AAAI Spring Symposium on Quantum Interaction (2007)
7. Lee, D.D., Sebastian Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
8. Melucci, M.: A basis for information retrieval in context. ACM Transactions on Information Systems 26(3), 1–41 (2008)
9. Piwowarski, B., Frommholz, I., Lalmas, M., Van Rijsbergen, K.: What can Quantum Theory bring to Information Retrieval? In: CIKM 2010 (2010)
10. Song, D., Lalmas, M., van Rijsbergen, C.J., Frommholz, I., Piwowarski, B., Wang, J., Zhang, P., Zuccon, G., Bruza, P.D.: How Quantum Theory is Developing the Field of Information Retrieval. In: Quantum Informatics Symposium. AAAI Fall Symposia Series, pp. 11–14. AAAI Press, Menlo Park (2010)
11. Tucci, R.R.: Quantum Bayesian Nets. Arxiv preprint quant-ph/9706039 9, 295–337 (1997)
12. Van Rijsbergen, C.J.: The Geometry of Information Retrieval. Cambridge University Press, Cambridge (2004)

13. Widdows, D., Cohen, T.: Semantic vector combinations and the synoptic gospels. In: Bruza, P., Sofge, D., Lawless, W., van Rijsbergen, K., Klusch, M. (eds.) QI 2009. LNCS, vol. 5494, pp. 251–265. Springer, Heidelberg (2009)
14. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization, pp. 267–273. ACM, New York (2003)
15. Zuccon, G., Azzopardi, L.A., van Rijsbergen, K.: The quantum probability ranking principle for information retrieval. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 232–240. Springer, Heidelberg (2009)

Pure High-Order Word Dependence Mining via Information Geometry

Yuexian Hou[1], Liang He[1], Xiaozhao Zhao[1], and Dawei Song[2]

[1] School of Computer Sci & Tec, Tianjin University, Tianjin, China
{krete1941,roba269,0.25eye}@gmail.com
[2] School of Computing, The Robert Gordon University, Aberdeen, United Kingdom
d.song@rgu.ac.uk

Abstract. The classical bag-of-word models fail to capture contextual associations between words. We propose to investigate the "high-order pure dependence" among a number of words forming a semantic entity, i.e., the high-order dependence that cannot be reduced to the random coincidence of lower-order dependence. We believe that identifying these high-order pure dependence patterns will lead to a better representation of documents. We first present two formal definitions of pure dependence: Unconditional Pure Dependence (UPD) and Conditional Pure Dependence (CPD). The decision on UPD or CPD, however, is a NP-hard problem. We hence prove a series of sufficient criteria that entail UPD and CPD, within the well-principled Information Geometry (IG) framework, leading to a more feasible UPD/CPD identification procedure. We further develop novel methods to extract word patterns with high-order pure dependence, which can then be used to extend the original unigram document models. Our methods are evaluated in the context of query expansion. Compared with the original unigram model and its extensions with term associations derived from constant n-grams and Apriori association rule mining, our IG-based methods have proved mathematically more rigorous and empirically more effective.

Keywords: Language Model, Word Association, High-order Pure Dependence, Information Geometry, Query Expansion, Log likelihood Ratio Test.

1 Introduction

The classical bag of words models, such as the Vector Space Model (VSM) [18] and unigram language model (LM) [16], represent a document as a weighted vector or probabilistic distribution of words. Although it has been proved useful in practice, there is a major limitation: the contextual information between words, which is the key to form meaningful semantic entities, is missing. In many cases, the semantic entities are not necessarily limited to syntactically valid phrases or named entities. More generally they can be high-order association (also referred as high-order *dependence*) patterns, which are often beyond pair-wise relations, e.g. {"climate", "conference", "Copenhagen"}.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 64–76, 2011.

Recently, there have been attempts to extract term relationships, e.g., through the Apriori method in [20], co-occurrence analysis [19], and Word-net relations [13]. In this paper, we propose to consider high-order *pure dependence*, i.e., the high-order dependence that cannot be reduced to the random coincidence of lower-order dependence. Usually these dependence patterns cannot be simply judged by co-occurrence frequencies. For example, the words *a*, *the* and *of* almost co-occur in every English article. However, we cannot say that they form a pattern representing a semantic entity. The high frequency of their co-occurrence can be explained as some kind of "coincidence", because each of them or pairwise combinations has a high frequency independently. On the other hand, the co-occurrence of the words "climate", "conference" and "Copenhagen" implies a un-separable high-level semantic entity, which can not be fully explained as the random coincidence of, e.g., the co-occurrence of "Copenhagen" and "conference" (which can be any other conferences in Copenhagen) and the occurrence of "climate". We consider a high-order dependence among words "pure", if and only if the joint probability distribution of these words is significantly different from the product w.r.t any possible decomposition into lower-order joint distributions or marginal distributions. In the language of graphical model, it requires that the joint distribution can not be factorized.

Formally, given a set of binary random variables $\mathbb{X} = \{X_1, \ldots, X_n\}$, where X_i denotes the occurrence ($X_i = 1$) or absence ($X_i = 0$) of the i-th word. Let $x_i \in \{0, 1\}$ denote the value of X_i. Let $p(\boldsymbol{x})$, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, be the joint probability distribution over $\mathbb{X}$. Then the n-order pure dependence over $\mathbb{X}$ can be defined as follows.

Definition 1. *(UPD): $\mathbb{X} = \{X_1, \ldots, X_n\}$ is of n-order Unconditional Pure Dependence (UPD), iff it can NOT be unconditionally factorized, i.e., there does NOT exist a k-partition $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_k\}$ of $\mathbb{X}$, $k > 1$, such that $p(\boldsymbol{x}) = p(\boldsymbol{c}_1) \cdot p(\boldsymbol{c}_2) \cdots p(\boldsymbol{c}_k)$, where $p(\boldsymbol{c}_i)$, $i = 1, \ldots, k$, is the joint distribution over $\mathbb{C}_i$.*

In practice, it is also useful to strengthen our definition of pure dependence in order to eliminate conditional random coincidences. This leads to the following definition of *conditional pure dependence*.

Definition 2. *(CPD): $\mathbb{X} = \{X_1, \ldots, X_n\}$ has n-order Conditional Pure Dependence (CPD), iff it can NOT be conditionally factorized, i.e., there does NOT exist $\mathbb{C}_0 \subset \mathbb{X}$ and a k-partition $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_k\}$ of $\mathbb{V} = \mathbb{X} - \mathbb{C}_0$, $k > 1$, such that $p(\boldsymbol{v}|\boldsymbol{c}_0) = p(\boldsymbol{c}_1|\boldsymbol{c}_0) \cdot p(\boldsymbol{c}_2|\boldsymbol{c}_0) \cdots p(\boldsymbol{c}_k|\boldsymbol{c}_0)$, where $p(\boldsymbol{v}|\boldsymbol{c}_0)$ is the conditional joint distribution over $\mathbb{V}$ given $\mathbb{C}_0$, and $p(\boldsymbol{c}_i|\boldsymbol{c}_0)$, $i = 1, 2, \ldots, k$, is the conditional joint distribution over $\mathbb{C}_i$ given $\mathbb{C}_0$.*

Remark 1. *Definition 2 permits an empty $\mathbb{C}_0$. Hence CPD entails UPD.*

To our best knowledge, there has not been any efficient method to characterize the above high-order pure dependence in both sufficient and necessary senses. For a given partition $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_k\}$ of $\mathbb{X}$, the method in [21] and [3] can efficiently decide whether $p(\boldsymbol{x}) = p(\boldsymbol{c}_1) \cdot p(\boldsymbol{c}_2) \cdots p(\boldsymbol{c}_k)$. However, it is an exponential task if we directly test all possible partitions of $\mathbb{X}$ and identify the n-order UPD. In

a configuration of graphical model, it can be shown that the decision problem of UPD or CPD is NP-hard [4].

Regarding the issue of efficiency, one may develop heuristics based on pair-wise dependence measures, e.g., covariance and correlation coefficient. Nonetheless, they usually suffer from the ad-hoc nature in tuning the threshold to decide significant pure dependence. Chi-square statistic can avoid the ad-hoc threshold, but it is indirect in the high-order case. Association rule mining can also be used to find highly frequent word associations. However, it does not guarantee the resulting associations are pure dependence. On the other hand, the complete n-gram method is straightforward, but it often leads to a large amount of redundant and noisy information.

In this paper, we propose to use Information Geometry (IG) [2], which provides relevant theoretical insights and useful tools, to tackle these difficulties in a consistent framework. IG studies joint distribution by way of differential geometry. A space of probability distributions is considered as a differentiable manifold, each distribution as a point on the manifold with the parameters of the model as coordinates. There are different kinds of coordinate systems to fit the manifold (detailed in Section 3), and it turns out that the so called mixed coordinate systems with orthogonality are especially useful for our purpose. Based on the coordinate orthogonality, we can derive a set of statistics and methods for analyzing word dependence patterns by decomposing the dependence into various orders. As a result, the 2nd-order, 3rd-order and higher-order pure dependence can be singled out and identified by the log likelihood ratio test.

The main theoretical contributions of this paper are that we propose a series of theoretically proven sufficient criteria for identifying UPD or CPD, respectively, and the corresponding efficient implementations that use the log likelihood test to the θ-coordinate of IG. The proposed IG-based methods can control confidence level theoretically. Then we apply the extracted high-order pure dependence (UPD or CPD) patterns in query expansion by incorporating then into the unigram document representation in the Relevance Model [9].

2 Related Work

This paper focuses on effective extraction and utilization of high-order pure word dependence patterns in the context of information retrieval (IR). There have been studies on incorporating dependence in language models. For example, Niesler et al. [15] presented a variable-length category-based n-gram language model, and Zhang et al. [23] proposed a framework for combining n-grams in different orders. Gao et al. presented a dependence language model to incorporate grammatical linkages [5]. The Markov Random Field (MRF) model captures short and long range term dependencies [11] [12]. Song et al. [20] presented methods generating word associations based on association rule mining. Many enhancements to the classical bag-of-word representation of documents have been introduced, e.g., via the use of second-order co-occurrence information to build context vectors for word sense discrimination [19] and the combination of text

data with external knowledge (Wordnet) [13]. However, none of them explicitly considered high-order pure dependence.

The IG is systematically introduced by Amari [2] and has been successfully applied in the fields such as the study of neural spikes [14]. Based on IG, Hofmann [6] defined a Fisher kernel for learning document similarities by Support Vector Machines (SVM). However, the issue of high-order pure dependence was not considered in his work. In general, the application of IG in text processing tasks is not yet widely studied.

3 Preliminaries of Information Geometry

To illustrate our theoretical results and the corresponding algorithmic framework, it is necessary to explain the relevant background of IG [1] [2] [17] [8].

3.1 Coordinates of Probability Distributions

In IG, a family of probability distributions is considered as a differentiable manifold with certain coordinate system. In the case of binary random variables, we use three basic coordinate systems, namely *p-coordinates*, *η-coordinates*, and *θ-coordinates* [14]. To be specific, if we define an assignment over $\mathbb{X}$, denoted by $a_{\mathbb{X}} =< a_1, a_2, \ldots, a_n >$ (or $a_{\mathbb{X}} = a_1 a_2 \ldots a_n$ in short), which determines a certain value of $\boldsymbol{x}$ by assigning $a_i \in \{0, 1\}$ to X_i, $1 \leq i \leq n$, then the coordinate systems of IG can be defined as follows:

1. *p*-coordinates:

$$p_{a_{\mathbb{X}}} = p_{a_1 a_2 \ldots a_n} = Pr\{X_1 = a_1, \ldots, X_n = a_n\} > 0 \tag{1}$$

   where $p_{a_{\mathbb{X}}}$ is the joint probability and $a_i \in \{0, 1\}, 1 \leq i \leq n$. Note that it is sufficient to determine a n-variable joint distribution using $2^n - 1$ probabilities, due to the constraint $\sum_{a_1, a_2, \ldots, a_n} p_{a_1 a_2 \ldots a_n} = 1$. Also note that IG requires that any probability term is not zero. This requirement can be met by using any common smoothing method.
2. η-coordinates:

$$\begin{aligned} \eta_i &= E[x_i], \quad 1 \leq i \leq n \\ \eta_{ij} &= E[x_i x_j], \quad 1 \leq i < j \leq n \\ &\vdots \\ \eta_{12\ldots n} &= E[x_1 x_2 \ldots x_n] \end{aligned} \tag{2}$$

   Note we define the order of a η-coordinate by the number of its subscripts. For example, η_1 is 1-order, and η_{23} is 2-order. In the information retrieval context, a η-coordinate is effectively equivalent to the document frequency of a single term or a term combination, up to a normalization factor.
3. θ-coordinates: The coordinate system specially relevant to our goal is the θ-coordinates, which can be derived from the log-linear expansion of $p(\boldsymbol{x})$:

$$\log p(\boldsymbol{x}) = \sum_i \theta_i x_i + \sum_{i<j} \theta_{ij} x_i x_j + \cdots + \theta_{12\ldots n} x_1 x_2 \ldots x_n - \Psi \tag{3}$$

where Ψ is the normalization term corresponding to $\Psi = -\log p(\mathbf{0})$. It is easy to check that Formula (3) is an exact expansion since all x_i's are binary [14]. Note that we can also define the order of a θ-coordinate the same as in the η-coordinates.

As an example, we consider the case of $n = 3$. For the p-coordinate system, tuple-word joint distribution can be determined by arbitrary 7 out of 8 probabilities, e.g. $\{p_{000}, p_{001}, p_{010}, p_{011}, p_{100}, p_{101}, p_{110}\}$. The transform between p-coordinates and η-coordinates is trivial, say, $p_{111} = \eta_{123}, p_{011} = \eta_{23} - \eta_{123}, p_{100} = \eta_1 - \eta_{12} - \eta_{13} + \eta_{123}$. Based on formula (3), θ-coordinates can be given by the following equation if we have known p-coordinates:

$$\theta_{12\ldots n} = \log \prod_{k=0}^{n} \prod_{a_{\mathbb{X}} \in A_{\mathbb{X}}^{(k)}} p_{a_{\mathbb{X}}}^{(-1)^{n-k}} \tag{4}$$

where $A_{\mathbb{X}}^{(k)}$ denotes the set of all assignments, which assign 1 to k out of n variables, exactly. And based on formula (4), $\mathbb{X} = \{X_1, X_2, X_3\}$, $A_{\mathbb{X}}^{(0)} = \{000\}$, $A_{\mathbb{X}}^{(1)} = \{100, 010, 001\}$, $A_{\mathbb{X}}^{(2)} = \{101, 011, 110\}$, $A_{\mathbb{X}}^{(3)} = \{111\}$. Then we have

$$\theta_{123} = \log \frac{p_{111} p_{100} p_{010} p_{001}}{p_{110} p_{101} p_{011} p_{000}}.$$

Using the coordinate systems defined by the above, the set of all n-order joint probability distributions forms a d-dimensional manifold S_n, where $d = 2^n - 1$.

3.2 Coordinate Orthogonality

The Fisher information of two coordinate parameters ξ_i and ξ_j is defined as

$$g_{ij}(\boldsymbol{\xi}) = E\left[\frac{\partial \log p(\boldsymbol{x}, \boldsymbol{\xi})}{\partial \xi_i} \frac{\partial \log p(\boldsymbol{x}, \boldsymbol{\xi})}{\partial \xi_j}\right]$$

Here $E[\cdot]$ means the expectation with respect to $p(\boldsymbol{x}, \boldsymbol{\xi})$. In IG, the coordinate parameters ξ_i and ξ_j are called *orthogonal* when $g_{ij}(\boldsymbol{\xi}) = 0$ at any $\boldsymbol{\xi}$ [14].

From the definition of Fisher information, a direct observation is that, if ξ_i is orthogonal to ξ_j, the log-likelihood increment induced by $\Delta\xi_i$ is uncorrelated to the log-likelihood increment induced by $\Delta\xi_j$. Based on this observation, it can show that the maximum likelihood estimations of orthogonal parameters are independent to each other, and hence it entails a simple procedure of hypothesis test [14]. Note that such a simplification does not hold for other non-orthogonal parameterizations, e.g., correlation coefficients.

In Section 4, we will explicitly prove the theoretical connection between the n-order θ-coordinate and CPD (or UPD), which justifies that the θ-coordinate is

a relevant metric of high-order pure dependence. We thus aim to find a mixed coordinate system, denoted by ζ-coordinates, in which the high-order θ-coordinate parameter is orthogonal to all lower-order η-coordinates. This mixed coordinate system does exist: Generally, it can be shown that $\theta_{12\ldots n}$ is orthogonal to any η-coordinate less than n-order [14], and hence the (2^n-1)-dimensional ζ-coordinates can be given by $[\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta_{n-1}}, \theta_{12\ldots n}]^T$, where $\boldsymbol{\eta_1} = [\eta_1, \ldots, \eta_n]^T$, $\boldsymbol{\eta_2} = [\eta_{12}, \eta_{13}, \ldots, \eta_{(n-1)n}]^T$ and etc.

3.3 Coordinate Parameter Estimation

The θ-coordinates plays a central role in the identification of high-order pure dependence. However, a direct computation for high-order θ-coordinates can be numerically unstable. In addition, we desire a quantitative statistical significance level of the investigated θ-coordinate. Owing to the orthogonality between η-coordinates and θ-coordinates, Nakahara and Amari [14] develop a very efficient framework of Log Likelihood Ratio Test (LLRT) for θ-coordinates. However, Nakahara and Amari left the computation of high-order g_{dd} (the bottom-right element of the Fisher information matrix of ζ-coordinates) as an open problem, which is a necessary step for implementing the LLRT framework. To facilitate the LLRT framework, in the following Proposition 1, we develop a closed-form formula for computing g_{dd} in general[1].

Proposition 1

$$g_{dd} = \frac{1}{\sum_{\boldsymbol{x}} 1/p(\boldsymbol{x})} \tag{5}$$

The proof of Proposition 1 can be found in [7].

In the mixed ζ-coordinates, because of the orthogonality, the maximum likelihood estimation of the η's and the $\theta_{12\ldots n}$ can be performed independently [14]. Usually we can first estimate the η's from the corpus, and then calculate the $\hat{\theta}_{12\ldots n}$. In general, a larger absolute value of $\hat{\theta}_{12\ldots n}$ indicates a greater possibility that the word pattern is of pure dependence.

To guarantee a theoretic confidence level of the estimation for θ, the hypothesis test is needed. Here the null hypothesis $H_0 : \theta = \theta_0$, against $H_1 : \theta \neq \theta_0$. And we consider their log likelihood:

$$l_0 = \log p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \theta_0), \quad l_1 = \log p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \hat{\theta}).$$

We adopt the statistic of likelihood ratio test used in [14]

$$\begin{aligned}
\lambda &= 2\log\frac{l_1}{l_0} = 2\sum_{i=1}^{N}\log\frac{p(\boldsymbol{x}_i; \hat{\boldsymbol{\eta}}, \hat{\theta})}{p(\boldsymbol{x}_i; \hat{\boldsymbol{\eta}}, \theta_0)} \\
&\approx 2N \cdot E\big[\log\frac{p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \hat{\theta})}{p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \theta_0)}\big] = 2N \cdot D\big[p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \hat{\theta}) : p(\boldsymbol{x}; \hat{\boldsymbol{\eta}}, \theta_0)\big] \\
&\approx N g_{dd}(\hat{\theta} - \theta_0)^2
\end{aligned} \tag{6}$$

[1] Recently, Nakahara independently gets a theoretical result similar to Proposition 1 (according to our personal communication with Nakahara).

Here N is the number of documents, $D[\cdot : \cdot]$ denotes the Kullback-Leibler divergence, $\hat{\theta}$ can be estimated by (4), g_{dd} is the Fisher information of the mixed coordinates ζ in the θ-direction at point $(\hat{\boldsymbol{\eta}}; \hat{\theta})$ and can be given by Proposition 1. Also note that the last approximation equation is entailed by the well-known approximate relation between Kullback-Leibler divergence and Riemannian distance [14]. In this paper, we are interested in identifying significant pure dependence w.r.t the θ-parameter (the relation between pure dependence and the θ-parameter is discussed in Section 4). Hence we let $\theta_0 = 2$ and only apply the LLRT to those $|\hat{\theta}|$'s that are greater than θ_0. On the other hand, if $|\hat{\theta}| \leq \theta_0$, we simply consider that the pure dependence is absent.

Asymptotically, according to Wilks' theorem, we have $\pm\sqrt{N g_{dd}(\hat{\theta} - \theta_0)^2} \sim N(0,1)$. Here N(0,1) denotes the standard normal distribution. Hence $\lambda \sim \chi^2(1)$, that is, the χ^2 distribution with degree of freedom 1. Then we can control the probability of error theoretically.

4 The Spectrum of High-Order Pure Dependence

In this Section, we first introduce two extra definitions on high-order pure dependence, namely Pair-wise Pure Dependence (PPD) and Theta Pure Dependence (TPD), which are the sufficient criteria of UPD and CPD, respectively. Note that, from an algorithmic perspective, PPD or TPD are far more feasible than directly deciding UPD or CPD. Finally, we clarify the spectrum of all kinds of high-order pure dependence defined by this paper.

Definition 3. *(PPD): $\mathbb{X} = \{X_1, \ldots, X_n\}$ has n-order Pair-wise Pure Dependence (PPD), iff every 2-order θ-coordinate θ_{ij}, $1 \leq i < j \leq n$, is significantly different from zero.*

Definition 4. *(TPD): $\mathbb{X} = \{X_1, \ldots, X_n\}$ has n-order Theta Pure Dependence (TPD), iff the n-order θ coordinate $\theta_{12\ldots n}$ is significantly different from zero.*

In Definitions 3 and 4, the significance level can be decided w.r.t an appropriate confidence interval of the LLRT described in Section 3.3. The following two propositions show the spectrum relation between PPD, TPD, UPD, and CPD.

Proposition 2. *PPD $\Rightarrow$ UPD.*

Proof. We will prove ¬UPD $\Rightarrow$ ¬PPD. Assume $\mathbb{X} = \{X_1, \ldots, X_n\}$ does NOT have the n-order UPD, i.e., there exists a nontrivial partition $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_k\}$ of $\mathbb{X}$, such that $p(\boldsymbol{x}) = p(\boldsymbol{c}_1) \cdot p(\boldsymbol{c}_2) \cdots p(\boldsymbol{c}_k)$. Without loss of generality, we assume that X_1 and X_2 belong to $\mathbb{C}_1$ and $\mathbb{C}_2$, respectively. Summarize all variables of $p(\boldsymbol{x})$, except for X_1 and X_2. We have $\sum_{x_3 \ldots x_n} p(\boldsymbol{x}) = p(x_1)p(x_2)$. Hence, X_1 is independent to X_2, and θ_{12} vanishes by the definition of θ-coordinates (Formula 4). The proposition follows. □

Table 1. 2-order and 3-order pure dependence patterns (TREC AP8889)

Orders	2-order PD		3-order PD		
1	soviet	union	bush	jackson	vote
2	bush	democrat	bush	democrat	dole
3	bush	dole	republican	elect	presidenti
4	israel	palestinian	israel	palestinian	peac
5	attornei	judg	attornei	judg	trial
6	govern	rebel	militari	troop	rebel
7	militari	soldier	militari	troop	soldier

Index by Lemur toolkits v4.1 with Porter Stemmer

Proposition 3. *TPD* $\Rightarrow$ *UPD; TPD* $\Rightarrow$ *CPD*

Proof. We will first prove $\neg$UPD $\Rightarrow$ $\neg$TPD. First, we give several definitions and notations. Let $\mathbb{C} \subset \mathbb{X}$, $a_{\mathbb{C}}$ is a sub-assignment of $a_{\mathbb{X}}$ iff $a_{\mathbb{C}}$ assigns the same value to $\mathbb{C}$ as $a_{\mathbb{X}}$. We call an assignment (or sub-assignment) odd iff it assigns odd number of 1's to variables. Otherwise, it is an even assignment.

Let us consider the term inside the logarithmic function of $\theta_{12\ldots n}$, i.e., $\prod_{k=0}^{n} \prod_{a_{\mathbb{X}} \in A_{\mathbb{X}}^{(k)}} p_{a_{\mathbb{X}}}^{(-1)^{n-k}}$. According to Formula 4, if n is odd, the numerator and denominator of this term can be rewritten as $\prod_{a_{\mathbb{X}}\ is\ odd} p_{a_{\mathbb{X}}}$ and $\prod_{a_{\mathbb{X}}\ is\ even} p_{a_{\mathbb{X}}}$, respectively. On the other hand, if n is even, the numerator and denominator will be interchanged.

If the joint distribution $p(\boldsymbol{x})$ can be factorized, without loss of generality, assume that there exists a partition $\{\mathbb{C}_1, \mathbb{C}_2\}$ of $\mathbb{X}$, such that $p(\boldsymbol{x}) = p(\boldsymbol{c}_1) \cdot p(\boldsymbol{c}_2)$. Then, for an arbitrary given assignment $a_{\mathbb{X}}$, we have $p_{a_{\mathbb{X}}} = p_{a_{\mathbb{C}_1}} p_{a_{\mathbb{C}_2}}$. Let's count the occurring number of $p_{a_{\mathbb{C}_1}}$ in the numerator and denominator, respectively. We can see that the occurring number of $p_{a_{\mathbb{C}_1}}$ in the numerator is the same as the occurring number of $p_{a_{\mathbb{C}_1}}$ in the denominator, since the number of odd assignments is exactly the same as the number of even assignments. It turns out that every occurrence of $p_{a_{\mathbb{C}_1}}$ or $p_{a_{\mathbb{C}_2}}$ in the numerator can be eliminated by the corresponding occurrence in the denominator. Hence, we have $\prod_{k=0}^{n} \prod_{a_{\mathbb{X}} \in A_n^{(k)}} p_{a_{\mathbb{X}}}^{(-1)^{n-k}} = 1$, which entails a vanishing $\theta_{12\ldots n}$. Up to now, we indeed prove that TPD $\Rightarrow$ UPD.

If $p(\boldsymbol{x})$ can be conditionally factorized, we could show that $\theta_{12\ldots n}$ also vanishes by a similar approach. Hence, TPD $\Rightarrow$ CPD follows. □

5 Implementation and Complexity Analysis

PPD requires that every pair of variables is significantly dependent. In order to decide whether n variables form a PPD pattern, we need perform C_n^2 times of LLRT on the involved 2-order θ parameters. In each 2-order LLRT procedure, we need sum all samplings to obtain the corresponding 4 p-coordinates and compute the corresponding g_{33}. These steps takes $O(N)$ time, where N is the number of samplings. Hence the identifying procedure of n-order PPD takes $O(n^2 N)$ time

in total. In practice, we are often interested in finding all maximal PPD patterns up to a given order $n_0 < n$. Here the maximal PPD pattern refers to the PPD pattern that cannot be enlarged. This problem is the maximal clique problem of the graph generated by the following rule: 1 A variable is denoted by a vertex; 2 An edge connects two vertices iff the corresponding two variables form a 2-order PPD pattern. As Tsukiyama et al. showed [22], it is possible to list all maximal cliques in a graph in an amount of time that is polynomial per generated clique. Hence our problem can be efficiently solved if the number of all maximal PPD patterns, up to n_0-order, is a polynomial function of n_0. The number of PPD patterns can be controlled by an appropriate significance level of LLRT.

In order to decide whether n variables form a TPD pattern, we need only to perform a single LLRT on the involved n-order θ parameter. The estimate of a n-order θ takes $O(N)$ time. Hence, the identifying procedure of a n-order TPD only takes $O(N)$ time in total.

Mining all TPD patterns, up to n_0-order, are much time-consuming since high-order TPD patterns can not be directly derived from the lower-order TPD patterns. Hence we adopt two pre-selection sets as the candidates of TPD patterns: 1 all PPD patterns up to n_0-order; 2 all frequent co-occurrence patterns, up to n_0-order, w.r.t certain frequency threshold. We then test whether the corresponding θ-coordinates of the candidate patterns are significantly different from zero. The TPD generated from the above two pre-selection sets are called TPD1 and TPD2, respectively.

As an illustration, here we show some interesting dependence patterns extracted from TREC AP8889 by PPD methods in Table 1.

6 Application

6.1 An Extended Relevance Model

In the framework of Relevance Model (RM), we estimate the probability distribution $P(w|R)$, where w is an arbitrary word and R is the unknown underlying relevance model, which is usually approximated by the topmost documents (e.g. n=50) of the initial retrieval. Then we pick up the words w with high probability $P(w|R)$, forming an expanded query.

The mining of $P(w|R)$ can be extended to incorporate the word patterns with high-order pure dependence. In this section, we provide an extended relevance model, which employs the high-order pure dependence as a complement of the classic relevance model. We pick the top n returned documents of the initial retrieval, and extract the high-order dependence patterns using various different methods. For each dependence pattern c in the dependence set C, we calculate

$$P(c|R) = \frac{\text{Number of chunks containing } c}{\text{Total number of chunks}}.$$

Intuitively, we belief that a word in some high-order pure dependence patterns should carry more semantic importance. Hence we interpolate the weight due to high-order pure dependence with the weight estimated using the interpolated relevance model RM3 [9, 10].

$$D_{combine}(w|R) = \lambda D(w|R) + (1-\lambda)P(w|R). \quad (7)$$

where $D(w|R) = \sum_{c:w\in c} P(c|R)$.

We consider $D_{combine}(w|R)$ as the new weight for word w in our extended relevance model. The following experimental results shows that this extended model outperforms the classical model significantly in most cases.

6.2 Experimental Setup

We evaluate our model using four TREC collections: AP8889 with topic 101-150 (the *title* field), AP8889 with topic 151-200 (the *title* field), AP8889 with topic 201-250 (the *desc* field), and WSJ9092 with topic 201-250 (the *desc* field). Lemur 4.12 is used for indexing and retrieval.The first-round retrieval is carried out by a baseline language modeling (LM) approach with $\mu = 1000$. The Relevance Model (RM) is selected as the second baseline method with 50 feedback documents.

6.3 Results and Analysis

Figure 1 shows the 11-point interpolated average precision on TREC AP8889 and WSJ9092 datasets. We can see that all the query expansion method outperform the baseline language model, while the combined extended model is the best.

To further examine the merit of our IG-based high-order pure dependence model, we furthermore compare it with several other high-order dependence models, as shown in Table 2 (To keep it clean, we do not draw the curves of all methods on Figure 1). In Table 2, "Apr" indicates the Apriori method, which has many successful applications for finding the interesting item patterns. "CO" ("ConstOrder") indicates considering all the possible k-order word patterns. Due to the time and space limitations, we only examined the $k \leq 3$ case. "PPD", "TPD1" and "TPD2" indicate the methods described in Section 5. The combined methods are described in Section 6.1.

We can see that all high-order models outperform the baseline uni-gram RM. This verifies our intuition that the uni-gram RM and the high-order model are complementary to each other. Note that the best result can be achieved when the coefficient λ in (7) is set to about 0.1.

Table 2. MAP Performance comparison

QE Methods	AP8889 101-150	AP8889 151-200	AP8889 201-250	WSJ9092 201-250
LM	0.2331	0.3138	0.0862	0.1948
RM	0.3086	0.4042	0.0879	0.2060
PPD	0.2963 (-4.99%)	0.3859 (-4.53%)	0.0865 (-1.59%)	0.2402 (+16.60%)*
RM+CO	0.3109 (+0.75%)*	0.4101 (+1.46%)	0.0949 (+7.96%)	0.2121 (+2.96%)
RM+Apr	0.3093 (+0.23%)*	0.4168 (+3.12%)*	0.0900 (+2.39%)	0.2176 (+5.63%)*
RM+PPD	**0.3173** (+2.82%)*	0.4218 (+4.35%)*	0.0999 (+13.65%)*	**0.2488** (+20.78%)*
RM+TPD1	0.3153 (+2.17%)*	**0.4232** (+4.70%)*	**0.1003** (+14.11%)*	0.2441 (+18.50%)*
RM+TPD2	0.3166 (+2.58%)*	0.4191 (+3.69%)*	0.0972 (+10.58%)*	0.2211 (+7.33%)*

Significant improvements (at level 0.05) over RM are marked with "".

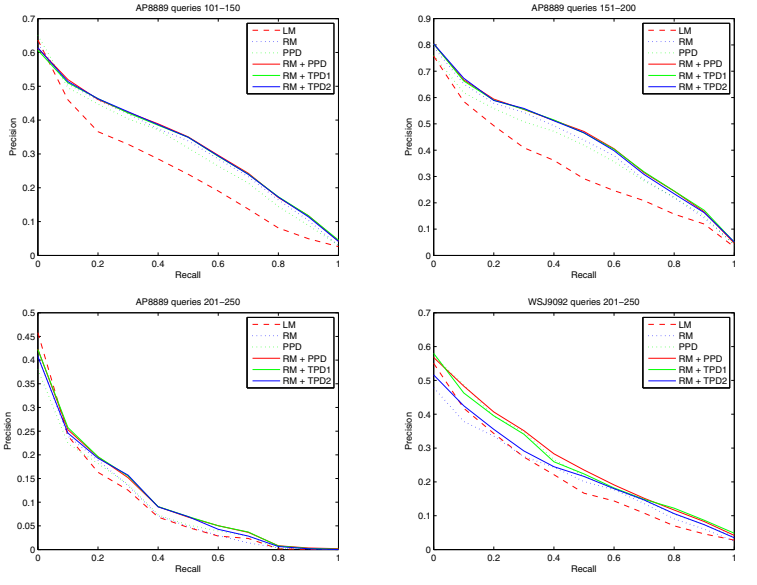


Fig. 1. P-R curve on TREC AP and WSJ

We can also note the PPD/TPD method outperform ConstOrder method and Apriori method significantly, especially on the WSJ9092 dataset. We believe one of the reasons is that the query we selected for WSJ9092 dataset (the *desc* field of topic 201-250) are long and complicated, in which case our IG-based high-order pure model have more advantages.

To show the different performance between TPD and PPD, we compare the results from different parameter λ's. It is shown that the averaged performance is almost the same, but the TPD method is more stable on sub-optimal parameter setting, suggesting that, if we cannot afford the time to train the parameters of the model, TPD method is "safer". In addition, the set of TPD patterns is often much reduced, which can offer a more economic high-order model.

7 Conclusions and Future Work

We analytically clarified a spectrum of high-order pure dependence, and proposed a novel framework based on Information Geometry to extract high-order pure word dependence patterns from documents. In this IG-based framework, we developed a set of rigorously-established justifications and feasible algorithms to single out high-order pure dependence by a well-founded statistical procedure (i.e. the log likelihood ratio test). We also integrate the automatically derived high-order pure dependence patterns into the Relevance Model. Evaluation results demonstrated the usefulness of the high-order pure dependence, and the effectiveness and robustness of our IG-based approach.

Our future work will be focused on addressing the following issues. First, we will perform a systematic analysis to clarify the semantic distinctions between PPD and TPD. Second, we will compare our approach with stronger baselines that utilize term dependence in IR, e.g., the dependence language model [5] and the MRF model [11]. Finally, we exploit the integration of a suitable level of syntactical dependence information into our framework.

Acknowledgements. The authors would like to thank anonymous reviewers for their constructive comments. This work is supported in part by the Natural Science Foundation of China (NSFC, grant 61070044); NSF of Tianjin (grant 09JCYBJC00200); the NSFC-RSE (Royal Society of Edinburgh) International Joint Project Scheme; the EU FP7 through its Marie Curie IRSES (grant 247590); and the UK's Engineering and Physical Sciences Research Council (grant EP/F014708/2).

References

1. Amari, S.: Information geometry on hierarchy of probability distributions. IEEE Transactions in Information Theory 47(5), 1701–1711
2. Amari, S., Nagaoka, H.: Methods of Information Geometry. American Mathematical Society, Providence (2001)
3. Bakirov, N.K., Rizzo, M.L., Székely, G.J.: A multivariate nonparametric test of independence. Journal of Multivariate Analysis 79(8), 1742–1756
4. Chickering, D., et al.: Large-sample learning of bayesian networks is np-hard. The Journal of Machine Learning Research 5, 1287–1330
5. Gao, J., Nie, J.Y., et al.: Dependence language model for information retrieval. In: Proceedings of SIGIR 2004, pp. 170–177 (2004)
6. Hofmann, T.: Learning the similarity of documents: An information-geometric approach to document retrieval and categorization
7. Hou, Y., et al.: Efficient factorization test and high-order pure dependence mining. Submitted to NIPS 2011 (2011)
8. Jeffreys, H.: An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A 186 (1946)
9. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proceedings of SIGIR 2001, pp. 120–127 (2001)
10. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: Proceedings of CIKM 2009, pp. 1895–1898 (2009)
11. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of SIGIR 2005, pp. 472–479 (2005)
12. Metzler, D., Croft, W.B.: Latent concept expansion using markov random fields. In: Proceedings of SIGIR 2007, pp. 311–318 (2007)
13. Mihalcea, R., Corley, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of AAAI 2006, pp. 775–780 (2006)
14. Nakahara, H., Amari, S.: Information geometric measure for neural spikes. Neural Computation 14(10), 2269–2316
15. Niesler, T.R., Woodland, P.C.: A variable-length category-based n-gram language model. In: Proceedings of IEEE ICASSP 1996, pp. 164–167 (1996)

16. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of SIGIR 1998, pp. 275–281 (1998)
17. Rao, C.R.: Information and Accuracy Attainable in the Estimation of Statistical Parameters. Bull. Calcutta. Math. Soc. 37 (1945)
18. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11)
19. Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–123
20. Song, D., Huang, Q., Rueger, S., Bruza, P.: Facilitating query decomposition in query language modeling by association rule mining using multiple sliding windows. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 334–345. Springer, Heidelberg (2008)
21. Taskinen, S., Oja, H., Randles, R.H.: Multivariate nonparametric tests of independence. Journal of the American Statistical Association 100(471), 916–925
22. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing 6(3), 505–517
23. Zhang, S., Dong, N.: An effective combination of different order n-grams. In: Proceedings of O-COCOSDA 2003, pp. 251–256 (2003)

Promoting Divergent Terms in the Estimation of Relevance Models

Javier Parapar and Álvaro Barreiro

IRLab, Computer Science Department
University of A Coruña, Spain
{javierparapar,barreiro}@udc.es

Abstract. Traditionally the use of pseudo relevance feedback (PRF) techniques for query expansion has been demonstrated very effective. Particularly the use of Relevance Models (RM) in the context of the Language Modelling framework has been established as a high-performance approach to beat. In this paper we present an alternative estimation for the RM promoting terms that being present in the relevance set are also distant from the language model of the collection. We compared this approach with RM3 and with an adaptation to the Language Modelling framework of the Rocchio's KLD-based term ranking function. The evaluation showed that this alternative estimation of RM reports consistently better results than RM3, showing in average to be the most stable across collections in terms of robustness.

1 Introduction and Motivation

In the history of the Information Retrieval research, efforts to improve retrieval effectiveness have been centred in both developing better retrieval models by including new features or using different theoretical frameworks; and in designing new techniques to be incorporated on top of existing models to improve their performance. Particularly on the later, Query Expansion (QE) has proven to be effective from very early research stages. QE approaches can be classified between global techniques which produce a query rewriting without considering the original rank produced by the query, and local techniques in which the expanded query is generated using the information of the initial retrieval list.

In [19] Salton presented the initial efforts on exploiting the local information to improve the query formulation introducing, among others, Rocchio approach [16] working on the Vector Space Model framework. This family of local techniques is called Relevance Feedback (RF) [17] and it is based on using the relevant documents in the initial retrieval set in order to reformulate the query based on their content. Nevertheless, in a real retrieval scenario it is not realistic to assume that relevance judgements are available. Because of this, Pseudo Relevance Feedback (PRF) algorithms have been investigated [6,21]. PRF methods are based on assuming relevance of a set of documents retrieved by the original query. The set of documents which are assumed to be relevant and the way in

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 77–88, 2011.

which their information is exploited to improve the original query varies from one PRF method to another.

Lately, a PRF technique has been presented in the Language Modelling framework and proven very successful to improve retrieval effectiveness. This approach, called Relevance Models (RM) [10], has been established as a high-performance PRF approach showing great improvements over the results obtained with the initial ranking. Sicen it was originally presented in [10] it has been used in combination with other approaches such as the employment of query variants [5], cluster based retrieval [11], passage retrieval [12] or sentence retrieval [3]. Originally, Lavrenko and Croft presented [10] two different estimations of a relevance model: RM1 and RM2.

Despite the success of the RM, it was only recently when Lv and Zhai [14] tackled the necessity of comparing different estimations for the RMs. In [14] they compared five methods to estimate the query language models: RM3 and RM4 [1]; a divergence minimization model (DMM) and a simple mixture model (SMM) [23]; and a regularized mixture model (RMM) [20]. The main finding of this paper was that, in general, RM3 is the best and most stable method among the others. RM3 and RM4 [1] are extensions of the originally formulated RM1 and RM2 approximations, respectively. These extensions linearly interpolate the original query with the terms selected for expansion using RM1 or RM2.

The contributions of our paper are two PRF techniques that promote divergent terms and their comparison with RM3. Back in 2001 Carpineto et al. [4] presented a discriminational model to score candidate expansion terms in the Rocchio's framework based on the Kullback-Leibler Distance (KLD). This method improved the results of the standard Rocchio method. In this work we adapt this approach to work under the Language Modelling framework, improving also the performance of the original method by interpolating the selected expansion terms with the original query as in RM3. In our second contribution we present a new RM estimation that promotes divergent terms for expansion, i.e., terms that are far from the collection language model. We adopted the evaluation methodology from [14] and the results showed that the new estimated relevance model performs better than RM3 and that its behaviour, in terms of robustness across collections, is more stable than the other methods.

The rest of the paper is as follows. Section 2 presents the background. Section 3 explains the proposed methods for PRF with promotion of divergence. In Section 4 the evaluation and its results are reported. Section 5 describes the related work and, finally, conclusions and future work are reported in Section 6.

2 Background

In this section we will introduce the theoretical basis for this work: the retrieval method for the initial ranking, the different formulations for the RM and the KLD based discriminational model presented in [4].

2.1 Language Modelling for the Initial Ranking

The RM for PRF was presented within the Language Modelling (LM) theoretical framework. In Language Modelling the probability of a document given a query, $P(d|q)$, is estimated using the Bayes' rule as presented in Eq. 1.

$$P(d|q) = \frac{P(q|d) \cdot P(d)}{P(q)} \stackrel{rank}{=} \log P(q|d) + \log P(d) \tag{1}$$

In practice $P(q)$ is dropped for document ranking purposes. The prior $P(d)$ encodes a-priori information on documents and the query likelihood, $P(q|d)$, incorporates some form of smoothing. In this paper we consider uniform priors and uni-gram language models with Dirichlet smoothing [24], see Eq. 2.

$$P(q|d) = \prod_{i=1}^{n} P(q_i|d) = \prod_{i=1}^{n} \frac{tf(q_i, d) + \mu \cdot P(q_i|C)}{NT_d + \mu} \tag{2}$$

where n is the number of query terms, $tf(q_i, d)$ is the raw term frequency of q_i in d, NT_d is the document length expressed in number of terms, and μ is a parameter for adjusting the amount of smoothing applied. $P(q_i|C)$ is the probability of the term q_i occurring in the collection C that is usually obtained with the maximum likelihood estimator computed using the collection of documents.

After obtaining the initial ranking using the original query, the PRF methods assume relevance over a subset of retrieved documents. This set is usually called *relevance set*. The information of those documents is then used to improve the initial retrieval. The most common way of achieving this objective is expanding the original query and producing a second retrieval with the reformulated query. Next, different models to produce expanded queries are analysed.

2.2 Relevance Models

The RM approach builds better query models using the information given by the pseudo relevant documents. Two estimations were originally presented in [10]. RM1 assumes that the words in the relevant documents and the query words are sampled identically and independently from the relevance model. The result is an estimation where the query likelihood for every document is used as the weight for the document and the probability of a word is averaged over every document language model. In contrast, RM2 assumes that the query words are independent of each other, but they are dependent of the words of the relevant documents (conditional sampling). The result is that relevant documents containing query words can be used for computing the association of the their words with the query terms. A quite detailed explanation of the RM for PRF is given in the Chapter 7 of the book [7] by Croft et al.

In RM the original query is considered a very short sample of words obtained from the relevance model (R). If more words from R are desired then it is reasonable to choose those words with highest estimated probability when considering the words for the distribution already seen. So the terms in the lexicon of the

collection are sorted according to that estimated probability, which after doing the assumptions using the RM1 method, is estimated as in Eq. 3.

$$P(w|R) \propto \sum_{d \in C} P(d) \cdot P(w|d) \cdot \prod_{i=1}^{n} P(q_i|d) \tag{3}$$

Usually $P(d)$ is assumed to be uniform. $\prod_{i=1}^{n} P(q_i|d)$ is the query likelihood given the document model, which is traditionally computed using Dirichlet smoothing (see Eq. 2). Then for assigning a probability to the terms in the relevance model we have to estimate $P(w|d)$; in order to do so it is also common to use Dirichlet smoothing. The final retrieval is obtained by four steps:

1. Initially the documents in the collection C are ranked using their query likelihood. This query likelihood is usually estimated with some kind of smoothing, commonly Dirichlet smoothing as in Eq. 2.
2. A certain top r documents from the initial retrieval are taken for the estimation instead of the whole collection C, let us call this pseudo relevance set RS.
3. The relevance model probabilities $P(w|R)$ are calculated using the estimate presented in Eq. 3, with RS instead of C.
4. To build the expanded query the e terms with highest estimated $P(w|R)$ are selected. The expanded query is used to produce a second document ranking using negative cross entropy as in Eq. 4. In this second retrieval Dirichlet smoothing is commonly used.

$$\sum_{i=1}^{e} P(w_i|R) \cdot \log P(w_i|d) \tag{4}$$

RM3 is a later extension of RM that performs better than RM1 in terms of effectiveness. RM3 interpolates the terms selected by RM1 with the original query as in Eq. 5 instead of using them directly. The final query is used in the same way as in RM1 to produce a second ranking using negative cross entropy.

$$P(w|q') = (1 - \lambda) \cdot P(w|q) + \lambda \cdot P(w|R) \tag{5}$$

2.3 Kullback Leibler Divergence for Pseudo Relevance Feedback

In [4] Carpineto et al. presented a method for term scoring in the context of Rocchio's framework for PRF. Carpineto et al. tried to maximize the divergence between the probability distributions of the terms estimated in the pseudo relevance set (p_{RS}) and the distribution estimated over the whole collection (p_C). In order to do so they used the KLD calculated as in Eq 6 because it captures the relative entropy between both distributions. To build the expanded query they selected the terms that mostly contribute to the divergence of both distributions (higher KLD score). In that work they compared the KLD term ranking function with Rocchio's weights, Robertson's Selection Value [15], Chi-squared

and Doszkoc's variant of Chi-squared [8]. The results showed that the presented KLD term scoring function performed the best.

$$KLD(p_{RS}, p_C) = \sum_{w \in V} p_{RS}(w) \cdot \log \frac{p_{RS}(w)}{p_C(w)} \tag{6}$$

3 Promoting the Divergence in Pseudo Relevance Feedback

In this section we describe the two approaches presented under the Language Modelling framework to promote divergence in the PRF context.

3.1 Kullback Leibler Divergence Based Query Expansion in the Language Modelling Framework

Although the KLD method outperformed the other term ranking methods in the Rocchio's framework, it was not compared with RM in [4]. In our paper we compare the KLD method against the standard RM3 formulation adapting the KLD scoring from the Rocchio's framework to work under the Language Modelling framework. The KLD scoring function was computed as in Eq. 7

$$kld_{score}(w) = p_{RS}(w) \cdot \log \frac{p_{RS}(w)}{p_C(w)} \approx \frac{tf(tw, RS)}{NT_{RS}} \cdot \log \frac{tf(w, RS) \cdot NT_C}{NT_{RS} \cdot tf(w, C)} \tag{7}$$

where $tf(w, RS)$ is the term frequency of w in the pseudo relevance set, NT_{RS} is the number of terms in the pseudo relevance set RS, NT_C is the total number of terms in the collection and $tf(w, C)$ is the term frequency of w in the whole collection.

To obtain a probability for each of the e terms selected for expansion we re-normalized the scores obtained with Eq. 7 as in Eq. 8

$$KLD(w) = \frac{kld_{score}(w)}{\sum_{i=1}^{e} kld_{score}(w_i)} \tag{8}$$

In RM3 it was already demonstrated that the interpolation of the original query and the expanded query performs better. So we incorporated this idea in the KLD-based model interpolating the e terms selected as result of the KLD scoring formula with the original query. Therefore, the second retrieval is processed with an extended query as presented in Eq. in 9:

$$P(w|q') = (1 - \lambda) \cdot P(w|q) + \lambda \cdot KLD(w) \tag{9}$$

3.2 Relevance Models with Promotion of Divergent Terms

The KLD-based introduction of divergence in the Language Modelling framework presented above was made as a plug-in in the Language Modelling framework. According to the analysis presented in [14], the advantage in terms of

stability of RM3 was attributable to the use of the query likelihood scores in the estimation made by RM1, which is not present in the KLD approach. To take advantage of this, we present a new estimation that promotes divergent terms maintaining the benefits from the RM methods, i.e., the use of the query likelihood scores. This new estimation arises naturally when the objective is to select expansion terms that, having high estimated probability in the RS, diverge from the collection distribution, i. e. they are more discriminative terms.

Based on the original RM1 estimation presented in Eq. 3 the most straightforward way of introducing such idea is by replacing the $P(w|d)$ by $P(w|d) - P(w|C)$. In this way those terms whose density is higher in RS than in the collection are promoted, meanwhile those with low density in the RS are demoted. Another important point in order to reinforce the promotion of divergent terms is how $P(w|d)$ is smoothed. Usually in RM this is done using Dirichlet smoothing choosing as background distribution the collection distribution. In the presented method we decided to apply the smoothing but instead of using the collection distribution as background distribution we chose to use the distribution in the relevance set. Therefore, the objective is to get for expansion the best terms that describe the documents taking into account both the RS and the divergence from the collection distribution. The computation was performed as in Eq. 10.

$$P(w|d) - P(w|C) \propto \frac{tf(w,d) + \frac{\mu \cdot tf(w,RS)}{NT_{RS}}}{NT_d + \mu} - \frac{tf(w,C)}{NT_C} \tag{10}$$

Note that $P(w|d) - P(w|C)$ could provide negative scores for those terms with less estimated probability in the documents of the relevant set than in the whole collection. To avoid this a re-normalization of such subtraction is done, let us call the re-normalized term $P_{C^-}(w|d)$. With these considerations the final estimation is computed as in Eq 10.

$$P(w|R) \propto \sum_{d \in RS} P(d) \cdot P_{C^-}(w|d) \cdot \prod_{i=1}^{n} P(q_i|d) \tag{11}$$

After this, the second retrieval was performed as in RM3 (interpolating with the original query) as indicated in Section 2.2.

Another way of introducing the divergence idea would be the use of a document prior to promote documents that are far away from the collections' distribution, acting at document level rather than at term level. Nevertheless no improvements were achieved with our experiments applying that approach.

Now we have to remark an important point that, to the best of our knowledge, was never discussed properly in the context of RM: the different roles of smoothing the parameters in the distinct steps of the process. In RM3 smoothing is applied up to four times (see Section 2.2), and Dirichlet is commonly used in every occasion, so we can distinguish:

1. μ_1, the smoothing parameter in the initial retrieval (Eq 2, step 1).
2. μ_2, the smoothing parameter in $P(w|d)$ (Eq 3, step 3).

3. μ_3, the smoothing parameter in $\prod_{i=1}^{n} P(q_i|d)$ (Eq 3, step 3).
4. μ_4, the smoothing parameter in the second retrieval (Eq 4, step 4).

Usually in the literature all the four parameters are considered to be only one and the parameter is even not trained taking default values as for example in [14] ($\mu = 1000$). Although this may produce good values, being a very good property of the method, the roles of the different μ parameters are quite different. Meanwhile μ_1 and μ_3 parameters are clearly affecting the same query likelihood and should be kept equal, for the other two parameters this is not so clear. The parameters μ_1 and μ_4 control the smoothing in the document language model when calculating the query likelihood in order to produce a ranking but the nature of the queries of both retrieval processes is quite different: shorter queries against longer queries. Nevertheless it is demonstrated in [24] that the optimal μ values in both scenarios are quite similar, so we can fix $\mu_1 = \mu_3 = \mu_4$. On the contrary, the smoothing parameter μ_2 is used to control the smoothing when estimating the probability of the terms of the relevance model in order to select them to do the expansion. Although it is the language model of the document, here the document is not involved in the computation of a query likelihood, therefore, it can be considered a different parameter. For this reason it does not seem reasonable a-priori to fix the same values for the μ parameters used for retrieval as for the μ parameter used in the estimation of $P(w|d)$. This intuition was confirmed later in the experimentation, being the trained values quite different for both smoothing parameters. In fact, the optimal values trained in the evaluation process of both parameters in RM3 never matched.

4 Experiments and Results

This section describes the evaluation methodology and comments the results.

4.1 Collections

To evaluate the different approaches we chose the same collections used in previous works [14]: a subset of the Associated Press collection corresponding to the 1988 and 1989 years (AP88-89), the Small Web Collection WT2G and the disk 4 and 5 from TREC (TREC-678). Additionally, we decide to use the WT10G collection, which was not used in [14], to report test values in a web collection. In AP88-89, TREC-678 and WT10G we used training and test evaluation: we

Table 1. Collections and topics for training and test

Col.	*# of Docs*	*Topics*	
		Training	Test
AP88-89	164,597	51-100	101-200
WT2G	247,491	401-450	–
TREC-678	528,155	301-350	351-450
WT10G	1,692,096	451-500	501-550

performed training for MAP in a set of topics and testing over another set. In WT2G we report well-tuned values over the trained topics, as it was done in [14]. Short queries (title only) were used because they are the most suitable to be expanded. All the collections were preprocessed with standard stop-word removal and Porter stemmer. In Table 1 the evaluation settings are summarized.

4.2 Methods

We compared four methods:

- **LM:** the baseline Language Modelling retrieval model with Dirichlet smoothing as in Section 2.1.
- **RM3:** the standard formulation of RM3, as explained in Section 2.2.
- **KLD3:** the KLD based PRF method adapted as detailed in Section 3.1.
- **RM3DT:** the proposed formulation of RM with estimations promoting divergent terms as described in Section 3.2.

4.3 Training and Evaluation

As discussed before, we performed a training and test strategy, more precisely we perform training and test for AP88-89, TREC-678 and WT10G meanwhile well-tuned values are reported for WT2G as in [14].

The parameters tuned were: the smoothing parameter of the initial retrieval μ_1 ($\mu_1 \in \{10, 100, 1000, 2000, 3000, 4000, 5000, 6000\}$) that was also used for μ_3 and μ_4 and which was tuned for LM, KLD3, RM3 and RM3DT. The number of documents in the pseudo relevant set $r = |RS|$ ($r \in \{5, 10, 25, 50, 75, 100\}$) was tuned for KLD3, RM3 and RM3DT. The number of terms selected for expansion e ($e \in \{5, 10, 25, 50, 75, 100\}$) was tuned for KLD3, RM3, RM3DT. The interpolation weight λ ($\lambda \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$) was tuned for KLD3, RM3, RM3DT. The smoothing parameter μ_2 ($\mu_2 \in \{10, 100, 1000, 2000, 3000, 4000, 5000, 6000\}$) was tuned for RM3 and RM3DT.

Finally, test values are reported for Mean Average Precision (MAP) and Robustness Index (RI) over the initial retrieval (LM). The Robustness Index ($-1 \leq RI(q) \leq 1$), also called Reliability of Improvement Index, of a model with respect to a baseline was formulated by Sakai *et al.* in [18] as in Eq 12:

$$RI(q) = \frac{n_+ - n_-}{|q|} \tag{12}$$

where q is the set of queries over the RI has to be calculated, n_+ is the number of improved queries, n_- the number of degraded queries and $|q|$ the total number of queries in q.

4.4 Results

Analyzing the MAP values for the test topics (see Table 2) it has to be noted that the three PRF methods always outperform the baseline LM as expected.

Table 2. Values for Mean Average Precision (MAP) on the test topics. Statistical significant improvements (Wilcoxon $p < 0.1$, and Wilcoxon $p < 0.05$ underlined) with respect to LM, RM3, KLD3, and RM3DT are superscripted with l, r, k, and d respectively. Best values are bolded.

	MAP			
Col.	*LM*	*RM3*	*KLD3*	*RM3DT*
AP88-89	.2775	.3606$^{\underline{l}}$ (+30%)	**.3667**$^{\underline{l}}$ (+32%)	.3625$^{\underline{l}}$ (+31%)
WT2G	.3115	.3445$^{\underline{lk}}$ (+10%)	.3352$^{\underline{l}}$ (+7%)	**.3467**$^{\underline{lk}}$ (+11%)
TREC-678	.2190	.2589$^{\underline{l}}$ (+18%)	.2586$^{\underline{l}}$ (+18%)	**.2700**$^{\underline{lrk}}$ (+23%)
WT10G	.2182	.2468$^{\underline{l}}$ (+13%)	.2238 (+2%)	**.2478**$^{\underline{lrk}}$ (+13%)

The adaptation of the KLD method to the LM framework using query interpolation performs quite well, obtaining improvements up to the 32% in the AP88-89; this is a very interesting point considering that KLD3 has fewer parameters to tune. Nevertheless the other methods achieve statistically significant improvements over the KLD3 in four occasions.

The RM3 method performs also quite well in terms of effectiveness with great improvements over the baseline as expected, as it is the state-of-the art in PRF. RM3 performs better than KLD3 in three collections, achieving in one case statistically significance. In the AP88-89 collection the differences across the three PRF methods are negligible, not being never statistically significant.

The proposed RM3DT estimation achieves statistically significant improvements over the KLD3 method in three occasions and over the RM3 in two, being always better than the later in terms of MAP. Another important point to analyse is the robustness of the methods, and how this is maintained across collections. Considering the values presented in Table 3 we can conclude that the RI numbers of the KLD3 method are quite acceptable and similar across collections, except in the WT10G collection. RM3 values are still acceptable (always bigger than zero) but are considerable lower than the other methods in the AP88-89 and TREC-678 collections. Contrarily RM3 performs slightly better than the other methods in the WT2G collection. This fact may be explained because the values on the WT2G collection are well-tuned, suggesting that a good parameter setting affects to the robustness of the RM3 method. Comparing both RM methods RM3DT seems to be more stable in terms of RI across collections.

Table 3. Values for Robustness Index (RI) with respect to the LM baseline model for every collection. Best values are bolded.

	RI		
Col.	*RM3*	*KLD3*	*RM3DT*
AP88-89	.38	**.56**	**.56**
WT2G	**.44**	.38	.40
TREC-678	.16	**.52**	.38
WT10G	.28	-.04	**.36**

The differences in robustness between RM3 and RM3DT can be analysed observing the queries penalized by RM3 and improved by RM3DT. Let us take as example the query `Parkinsons disease`, for this query LM obtained an average precision of 0.3231, RM3 damaged the query to 0.2927, while RM3DT improved it to 0.5083. Observing the top 25 expansion terms selected in both approaches we can view that many good terms are selected by both methods (for example `patient, brain` or `alzheimer`) but the RM3 method introduces terms that are so common that, although being very present in the RS, they introduce a lot of noise in the retrieval such as `page, can, year, will, new, say, may` or `home`, meanwhile those terms are not present in the top 25 RM3DT expansion terms because they were penalized for being so common in the collection.

5 Related Work

In [23] the authors explored the divergence idea proposing a Divergence Minimization Model (DMM). The DMM approach tries to minimize the divergence between the query model and the model of the feedback documents. The DMM objective is to build a feedback model that is close to every pseudo relevant document language model and far away from the collection language model, which is assumed as the non-relevance model. This was stated as an optimization problem. The DMM approach was already compared in [14] with Relevance Models showing that DMM performs worse than RM3.

This paper is centred in the Language Modelling framework but it is necessary to say that the idea of using divergence to improve the retrieval performance has been already deeply studied under other retrieval models, to the point of existing whole models based on it. The Divergence From the Randomness (DFR) model [2] is based on a similar idea: the more the terms occurrences in the documents diverge from their expected occurrences considering a random distribution the more information carried by the terms. In the DFR model the QE process is done based on a generalization of the Rocchio's framework [9]. Different weighting schemes, including the aforementioned KLD, were tested being the Bose-Einstein Bo1 model the best in terms of effectiveness, which also select those terms that diverge most from the randomness, using for those estimations the collections' statistics. In another paper [22] the Rocchio's classical feedback method was integrated in the DFR framework for PRF.

In other IR tasks such as adaptive filtering this divergence idea has also been used. In [13] the authors presented different discriminative features for queries and documents to be used in a technique which learns for each query the interpolation weight of the original query with the expansion terms. Particularly the entropy of the feedback documents and the document clarity are used. With the entropy of the feedback documents basically they capture at term level how heterogeneous is the term distribution in the RS. With the clarity of the feedback documents they try to "explain away" common terms present in the RS.

6 Conclusions and Future Work

In this paper we have presented two different methods for PRF based on the idea of promoting the divergent terms in the RS. KLD3 is an adaptation to the LM framework of a KLD based method including the linear interpolation with the original query. RM3DT is a new estimation for the RM that computes the probability of a term given a feedback document by the subtracting to the terms' probability in the document its probability in the collection and applying the smoothing over the RS. It was also analysed the role of the different smoothing parameters involved in the RM methods, showing the different roles that those smoothing parameters play. We compared the new methods with the LM baseline andt he RM3 estimation. Particularly the RM3DT performed, for MAP, better than RM3 in every collection, showing, as the KLD3 method, a very good stability across collections in terms of robustness. We also want to study how the presented ideas may be applied to improve existing techniques for selective query expansion and adaptive relevance feedback.

Acknowledgments. This work was funded by *Ministerio de Ciencia e Innovación* under project TIN2008-06566-C04-04.

References

1. Abdul-Jaleel, N., Allan, J., Croft, W.B., Diaz, O., Larkey, L., Li, X., Smucker, M.D., Wade, C.: UMass at trec 2004: Novelty and hard. In: Proceedings of TREC-13 (2004)
2. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20, 357–389 (2002)
3. Balasubramanian, N., Allan, J., Croft, W.B.: A comparison of sentence retrieval techniques. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2007, pp. 813–814. ACM, New York (2007)
4. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. 19(1), 1–27 (2001)
5. Collins-Thompson, K., Callan, J.: Estimation and use of uncertainty in pseudo-relevance feedback. In: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 303–310. ACM, New York (2007)
6. Croft, W.B., Harper, D.: Using probabilistic models of document retrieval without relevance information. Journal of Documentation 35, 285–295 (1979)
7. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, 1st edn. Addison-Wesley Publishing Company, USA (2009)
8. Doszkocs, T.: Id, an associative interactive dictionary for online searching. Online Review 2, 163–173 (1978)
9. He, B., Ounis, I.: Combining fields for query expansion and adaptive query expansion. Inf. Process. Manage. 43, 1294–1307 (2007)
10. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 120–127. ACM, New York (2001)

11. Lee, K.S., Croft, W.B., Allan, J.: A cluster-based resampling method for pseudo-relevance feedback. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 235–242. ACM, New York (2008)
12. Li, X., Zhu, Z.: Enhancing relevance models with adaptive passage retrieval. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 463–471. Springer, Heidelberg (2008)
13. Lv, Y., Zhai, C.: Adaptive relevance feedback in information retrieval. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 255–264. ACM, New York (2009)
14. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 1895–1898. ACM, New York (2009)
15. Robertson, S.E.: On term selection for query expansion. J. Doc. 46, 359–364 (1991)
16. Rocchio, J.: Relevance feedback in information retrieval. In: The SMART Retrieval System, pp. 313–323 (1971)
17. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. Knowl. Eng. Rev. 18(2), 95–145 (2003)
18. Sakai, T., Manabe, T., Koyama, M.: Flexible pseudo-relevance feedback via selective sampling. ACM Transactions on Asian Language Information Processing (TALIP) 4(2), 111–135 (2005)
19. Salton, G.: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, Inc., Upper Saddle River (1971)
20. Tao, T., Zhai, C.: Regularized estimation of mixture models for robust pseudo-relevance feedback. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006, pp. 162–169. ACM, New York (2006)
21. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4–11. ACM, New York (1996)
22. Ye, Z., He, B., Huang, X., Lin, H.: Revisiting rocchios relevance feedback algorithm for probabilistic models. In: Cheng, P.-J., Kan, M.-Y., Lam, W., Nakov, P. (eds.) AIRS 2010. LNCS, vol. 6458, pp. 151–161. Springer, Heidelberg (2010)
23. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM 2001, pp. 403–410. ACM, New York (2001)
24. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004)

Is Document Frequency Important for PRF?

Stéphane Clinchant[1,2] and Eric Gaussier[2]

[1] Xerox Research Center Europe, Meylan, France
[2] LIG Université de Grenoble, UMR 5217/AMA team
stephane.clinchant@xrce.xerox.com, eric.gaussier@imag.com

Abstract. We introduce in this paper a new heuristic constraint for PRF models, referred to as the *Document Frequency (DF) constraint*, which is validated through a series of experiments with an oracle. We then analyze, from a theoretical point of view, state-of-the-art PRF models according to their relation with this constraint. This analysis reveals that the standard mixture model for PRF in the language modeling family does not satisfy the DF constraint on the contrary to several recently proposed models. Lastly, we perform tests, which further validate the constraint, with a simple family of *tf-idf* functions based on a parameter controlling the satisfaction of the DF constraint.

1 Introduction

Pseudo-relevance feedback (PRF) has been studied for several decades, and a lot of different models have been proposed, in all the main families of information retrieval (IR) models. In the language modelling approach to IR, for example, the mixture model for PRF is considered state-of-the-art, and numerous studies use it as a baseline. It has indeed been shown to be one of the most effective models in terms of performance and stability wrt parameter values in [11]. However, several recently proposed PRF models seem to outperform this mixture model, as models based on bagging, models based on a mixture of Dirichlet compound multinomial distributions, geometric relevance models or the log-logistic models of the recent information-based family [4,14,2,13]. This paper *aims at providing an explanation* of such improvements. In a nutshell, many of the recent models tends to favor terms with a high document frequency in the feedback set, a behavior we will capture with the Document Frequency constraint.

The notations we use throughout the paper are summarized in table 1, where w represents a term. We note n the number of pseudo relevant document used, F the feedback set and tc the number of term for pseudo relevance feedback. We call FTF, the feedback set term frequency and FDF, the feedback document frequency. The remainder of the paper is organised as follows. We give in Section 2 some basic statistics on three PRF models, which reveal global trends of PRF models. We then introduce in section 3 the Document Frequency constraint, that PRF models should satisfy, prior to reviewing standard PRF models according to their behavior wrt this constraint in section 4. We then introduce in section 5 a simple family of feedback functions which allows us to better understand the

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 89–100, 2011.

Table 1. Notations

Notation	Description
General	
q, d	Original query, document
$RSV(q,d)$	Retrieval status value of d for q
$c(w,d)$	# of occurrences of w in doc d
l_d	Length of doc d
avg_l	Average document length in collection
N	# of docs in collection
N_w	# of documents containing w
$\text{IDF}(w)$	$-\log(N_w/N)$
$tdfr(w,d)$	$c(w,d)\log(1+c\frac{\text{avg}_l}{l_d})$
PRF specific	
n	# of docs retained for PRF
$\mathbf{F}$	Set of documents retained for PRF: $\mathbf{F}=(d_1,\ldots,d_n)$
tc	*TermCount*: # of terms in $\mathbf{F}$ added to query
$FTF(w)$	$=\sum_{d\in F} c(w,d)$
$FDF(w)$	$=\sum_{d\in F} I(c(w,d)>0)$

relations between the different constraints, prior to discuss some related work in section 6.

2 Some Statistics on PRF

We begin this paper by analyzing the terms chosen and the performance obtained by three different, state-of-the-art, pseudo-relevance feedback (PRF hereafter) methods, namely the mixture model and the divergence minimization method in the language modeling family [15], and the mean log-logistic information model in the information-based family [2]. These models are reviewed later in section 4, and their exact formulation is not necessary here. In order to have an unbiased comparison, we use the same IR engine for the retrieval step. Thus, all PRF algorithms are computed on the *same* set of documents. Once new queries are constructed, we use either the Dirichlet language model (for the new queries obtained with the mixture model and the divergence minimization method) or the log-logistic model (for the new queries obtained with the mean log-logistic information model) for the second retrieval step, thus allowing one to compare the performance obtained by different methods on the same initial set of PRF documents. Two collections are used throughout this study: the ROBUST collection, with 250 queries, and the TREC 1&2 collection, with topics 51 to 200. Only query titles were used and all documents were preprocessed with standard Porter stemming, and all model parameters are optimized through a line search on the whole collection. The results obtained are thus the best possible results

Table 2. Statistics of the size of the Intersection

Collection	n	tc	Mean	Median	Std
robust	10	10	5.58	6.0	1.60
trec-12	10	10	5.29	5.0	1.74
robust	20	20	12	12	3.05
trec-12	20	20	11.8	13	3.14

one can get with these models on the retained collections. We first focus on a direct comparison between the mixture model and the mean log-logistic information model, by comparing the terms common to both feedback methods, i.e. the terms in the intersection of the two selected sets. Table 2 displays the mean, median and standard deviation of the size of the intersection, over all queries, for the collections considered. As one can note, the two methods agree on a little more than half of the terms (ratio mean by tc), showing that the two models select different terms. To have a closer look at the terms selected by both methods, we first compute, for each query, the total frequency of a word in the feedback set (i.e. $FTF(w)$) and the document frequency of this word in the feedback set (i.e. $FDF(w)$). Then, for each query we can compute the mean frequency of the selected terms in the feedback set as well as its mean document frequency, i.e. $q(ftf)$ and $q(fdf)$:

$$q(ftf) = \sum_{i=1}^{tc} \frac{ftf(w_i)}{tc} \text{ and } q(fdf) = \sum_{i=1}^{tc} \frac{fdf(w_i)}{tc}$$

We then compute the mean of the quantities over all queries.

$$\mu(ftf) = \sum_{q} \frac{q(ftf)}{|Q|} \text{ and } \mu(fdf) = \sum_{q} \frac{q(fdf)}{|Q|}$$

An average IDF can be computed in exactly the same way, where IDF is the standard inverse document frequency *in the collection*. Table 3 displays the above statistics for the three feedback methods: mixture model (MIX), mean log-logistic(LL) information model and divergence minimization model (DIV). Regarding the mixture and log-logistic models, on all collections, the mixture model chooses in average words that have a *higher FTF*, and a smaller *FDF*. The mixture model alos chooses words that are *more frequent in the collection* since the mean IDF values are smaller. On the other hand, the statistics of the divergence model shows that this model extracts very common terms, with low IDF and high FDF, which is one of the main drawback of this model. In addition to the term statistics, the performance of each PRF algorithm can also be assessed. To do so, we first examine the performance of the feedback terms *without* mixing them with the original queries, a setting we refer to as *raw*. Then, for each query we keep only terms that belong to the intersection of the mixture and log-logistic models (as the divergence model is a variant of the mixture model,

Table 3. Statistics of terms extracted by. Suffix A means $n = 10$ and $tc = 10$ while suffix B means $n = 20$ and $tc = 20$.

Settings	Statistics	MIX	LL	DIV
robust-A	$\mu(ftf)$	62.9	46.7	57.9
	$\mu(fdf)$	6.4	7.21	8.41
	Mean IDF	4.33	5.095	2.36
trec-1&2-A	$\mu(ftf)$	114 .0	79.12	98.76
	$\mu(fdf)$	7.1	7.8	8.49
	Mean IDF	3.84	4.82	2.5
robust-B	$\mu(ftf)$	68.6	59.9	68.2
	$\mu(fdf)$	9.9	11.9	14.4
	Mean IDF	4.36	4.37	1.7
trec-1&2-B	$\mu(ftf)$	137.8	100.0	118.45
	$\mu(fdf)$	12.0	13.43	14.33
	Mean IDF	3.82	4.29	2.0

we do not consider it in itself for this intersection), but keep their weight predicted by each feedback method. We call this setting *interse*. A third setting, *diff*, consists in keeping terms which do not belong to the intersection. Finally, the last setting, *interpo* for interpolation, measures the performance when new terms are mixed with the original query. This corresponds to the standard setting of pseudo-relevance feedback. Table 4 displays the results obtained. As one can note, the log-logistic model performs better than the mixture model, as found in [2]. What our analysis reveals is that it does so because it chooses better feedback terms, as shown by the performance of the *diff* setting. For the terms in the intersection, method *interse*, the weights assigned by the log-logistic model seem more appropriate than the weights assigned by the other feedback models.

Let's summarize our finding here. (a) The log-logistic model performs better than the mixture and divergence models for PRF. (b) The mixture and divergence models choose terms with a *higher* FTF. (c) The mixture model selects term with a smaller FDF, whereas (d) the divergence model selects terms with a smaller IDF. A first explanation of the better behavior of the log-logistic model can be that the FDF and IDF effect are dealt with more efficiently in this model, as shown by the statistics reported in table 3.

Table 4. MAP (%) Performance of different methods. Suffix A means $n = 10$ and $tc = 10$ while suffix B means $n = 20$ and $tc = 20$

	robust-A			trec-1&2			robust-B			trec-1&2-B		
FB Model	MIX	LL	DIV	MIX	LL	DIV	MIX	LL	DIV	MIX	LL	DIV
raw	23.8	26.9	24.3	23.6	25.7	24.1	23.7	25.7	22.8	25.1	27.0	24.9
interse	24.6	25.7	24.	24.2	24.5	23.4	25.3	26.2	22.6	26.1	26.5	24.7
diff	3	11.0	0.9	3	9	0.9	3.0	10.0	0.15	2.1	11.2	0.5
interpo	28.0	29.2	26.3	26.3	28.4	25.4	28.2	28.5	25.9	27.3	29.4	25.7

3 The Document Frequency Constraint

We adopt the axiomatic approach to IR [7] in order to present the Document Frequency constraint. Axiomatic methods were pioneered by Fang et al [7] and followed by many works. In a nutshell, axiomatic methods describe IR functions by constraints they should satisfy. According to [2], the four main constraints for an IR function to be valid are: the weighting function should (a) be increasing and (b) concave wrt term frequencies, (c) have an IDF effect and (d) penalize long documents. We first want to briefly discuss whether these constraints would make sense for PRF models.

In the context of PRF, the first two constraints relate to the fact that terms frequent in the feedback set are more likely to be effective for feedback, but that the difference in frequencies should be less important in high frequency ranges. The IDF effect is also relevant in feedback, as one generally avoids selecting terms with a low IDF, as such terms are scored poorly by IR systems. The constraint on document length is not as clear as the others in the context of PRF, as one (generally) considers sets of documents. What seems important however is the fact that occurrence counts are normalized by the length of the documents they appear in, in order not to privilege terms which occur in long documents.

Let $FW(w; \mathbf{F}, \mathbf{P}_w)$ denote the feedback weight for term w, with $\mathbf{P}_w$ a set of parameters dependent on w[1]. We now introduce a new PRF constraint which is based on the results reported in the previous section. Indeed, as we have seen, the best PRF results were obtained with models which favor feedback terms with a high *document frequency* ($FDF(w)$) in the feedback set, which suggests that, *all things being equal*, terms with a higher FDF should receive a higher score. This constraint can be formalized as follows:

PRF Constraint 1 [Document Frequency - DF]
Let $\epsilon > 0$, and w_a and w_b two words such that:
(i) $IDF(a) = IDF(b)$
(ii) The distribution of the frequencies of w_a and w_b in the feedback set are given by:

$$T(w_a) = (x_1, x_2, ..., x_j, 0, ..., 0)$$
$$T(w_b) = (x_1, x_2, ..., x_j - \epsilon, \epsilon, ..., 0)$$

with $\forall i,\ x_i > 0$ and $x_j - \epsilon > 0$ (hence, $FTF(w_a) = FTF(w_b)$ and $FDF(w_b) = FDF(w_a) + 1$).
Then: $FW(w_a; \mathbf{F}, \mathbf{P}_{w_a}) < FW(w_b; \mathbf{F}, \mathbf{P}_{w_b})$

In other words, FW is *locally* increasing with $FDF(w)$. The above constraint is sometimes difficult to check. The following theorem is useful to establish whether a PRF model, which can be decomposed in the documents of $\mathbf{F}$, satisfies or not the DF constraint:

[1] The definition of $\mathbf{P}_w$ depends on the PRF model considered. It minimally contains $FTF(w)$, but other elements, as IDF(w), are also usually present. We use here this notation for convenience.

Theorem 1. *Suppose FW can be written as:*

$$FW(w; \mathbf{F}, \mathbf{P}_w) = \sum_{d=1}^{n} f(x_w^d; \mathbf{P'}_w) \tag{1}$$

with $\mathbf{P'}_w = \mathbf{P}_w \setminus x_w^d$ *and* $f(0; \mathbf{P'}_w) \geq 0$. *Then:*

1. *If the function f is strictly concave, then FW meets the DF constraint.*
2. *If the function f is strictly convex, then FW does not meet the DF constraint.*

If f is strictly concave, then the function f is subadditive ($f(a+b) < f(a)+f(b)$). Let a and b be two words satisfying the conditions of the DF constraint. Then, we have:

$$FW(b) - FW(a) = f(x^j - \epsilon) + f(\epsilon) - f(x^j)$$

As the function f is subadditive, we have: $FW(b) - FW(a) > 0$. If f is strictly convex, then f is superadditive as $f(0) = 0$, and a comparable reasoning leads to $FW(b) - FW(a) < 0$. In the remainder of the paper, we will simply use the notation $FW(w)$ as a shorthand for $FW(w; \mathbf{F}, \mathbf{P}_w)$.

3.1 Validation of the DF Constraint

The DF constraint states that, all other parameters being equal, terms with higher DF should be preferred. Thus, in average, one should observe that terms with high DF scores yield larger increase in MAP values. To see whether this is the case, we computed the impact on the MAP of different terms selected from true relevance judgements, and plotted this impact against both TF and DF values. Our relying on true relevant documents and not documents obtained from pseudo-relevance feedback is based on (a) the fact that pseudo-relevance feedback aims at approximating relevance feedback, an (b) the fact that it is more difficult to observe clear trends in pseudo-relevance sets where the precision (e.g. P@10) and MAP of each query have large variances. The framework associated with true relevance judgements is thus cleaner and allows easier interpretation. In order to assess the impact of DF scores on the MAP values independently of any IR model, we make use of the following experimental setting:

- Start with a first retrieval with a Dirichlet language model;
- Let R_q denote the set of relevant documents for query q: Select the first 10 relevant documents if possible, else select the top $|R_q|(|R_q| < 10)$ relevant documents;
- Construct a new query (50 words) with the mixture model;
- Construct a new query (50 words) with the log-logistic model;
- Compute statistics for each word in the new queries.

Statistics include a normalized FDF, equal to $FDF(w)/|R_q|$, and a normalized FTF, first using a document length normalization, then using the transformation $\log(1 + FTF(w))/|R_q|$ to avoid too important a dispersion in plots. Each word w is added independently with weights predicted by the retained PRF model.

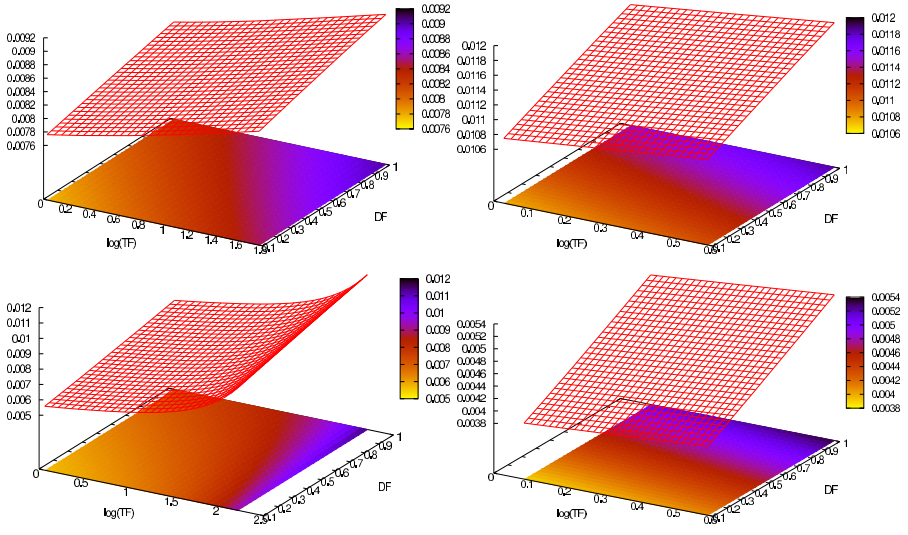


Fig. 1. (log(FTF),FDF) vs Δ MAP; true relevant documents are used with $n = 10$, $tc = 50$ and Gaussian kernel grids (30×30). Top row: log-logistic model; bottom row: mixture (language) model, left column:ROBUST Collection and right column: TREC-12 collection.

For each word w, we measure the MAP of the initial query augmented with this word. The difference in performance with the intial query is then computed as: $\Delta(\text{MAP}) = \text{MAP}(q+w) - \text{MAP}(q)$. We thus obtain, for each term, the following statistics: $\Delta(\text{MAP})$, $\log(1 + FTF(w))/|R_q|$, $FDF(w)/|R_q|$.

Figures 1 display a 3D view of these statistics for all queries, based on Gnuplot and two collections: TREC1&2 and ROBUST.

The TF statistics was normalized to account for different lengths and a Gaussian Kernel was used to smooth the data cloud. The shape of the plots obtained remains however consistent without any normalization and a different Kernel.

As one can note, on all plots of Figures 1, the best performing regions in the (TF,DF) space correspond to large DFs. Furthermore, for all TF values, the increase in MAP parallels the increase in DF (or, in other words, Δ(MAP) increases with DF for fixed TF). This validates the DF constraint and shows the importance of retaining terms with high DF in relevance feedback. Interestingly, the reverse is not true for TF. This implies that if terms with large TF are interesting, they should not be given too much weight. The results displayed in Table 3 suggest that the mixture model [15] suffers from this problem.

4 Review of PRF Models

We review in this section different PRF models according to their behavior wrt the DF constraint we have defined. We start with language models, then

review the recent model introduced in [14] which borrows from both generative approaches *à la* language model and approaches related to the *Probability Ranking Principle* (PRP), prior to review Divergence from Randomness (DFR) and Information-based models.

Mixture Model: Zhai and Lafferty [15] propose a generative model for the set **F**. All documents are i.i.d and each document is generated from a mixture of the feedback query model and the corpus language model:

$$P(\mathbf{F}|\theta_F, \beta, \lambda) = \prod_{w=1}^{V} ((1-\lambda)P(w|\theta_F) + \lambda P(w|C))^{FTF(w)} \quad (2)$$

where λ is a "background" noise set to some constant. For this model, $FW(w) = P(w|\theta_F)$ and θ_F is learned by optimising the data log-likelihood with an Expectation-Maximization (EM) algorithm. The above formula shows that the mixture multinomial model behaves as if all documents were merged together. As a result, the mixture model is agnostic wrt to DF, and thus does not satisfy the DF constraint.

Divergence Minimization: For language models, a divergence minimization model was also proposed in [15] and leads to the following feedback model:

$$FW(w) = P(w|\theta_F) \propto \exp\Big(\frac{1}{(1-\lambda)}\frac{1}{n}\sum_{i=1}^{n}\log(p(w|\theta_{d_i})) - \frac{\lambda}{1-\lambda}\log(p(w|C))\Big)$$

Furthermore, this equation corresponds to the form given in equation 1 with a strictly concave function (log). Thus, by Theorem 1, this model satisfies the DF constraint. Despite this good theoretical behavior, our previous experiments, reported in Table 4, as well as those reported in [11], show that this model does not perform as well as other ones. Indeed, as shown in Table 3, the IDF effect is not sufficiently enforced, and the model fails to downweight common words.

Relevance Model: Another PRF model proposed in the framework of the language modeling approach is the so-called relevance model, proposed by Lavrenko *et al.* [8], and defined by:

$$FW(w) \propto \sum_{d \in \mathbf{F}} P_{LM}(w|\theta_d)P(d|q) \quad (3)$$

where P_{LM} denotes the standard language model. The above formulation corresponds to the form of equation 1 of Theorem 1, with a linear function, which is neither strictly concave nor strictly convex. This model is neutral wrt the DF constraint. The relevance model has recently been refined in the study presented in [13] through a geometric variant, referred to as GRM, and defined by:

$$FW(w) \propto \prod_{d \in \mathbf{F}} P_{LM}(w|\theta_d)^{P(d|q)}$$

As the log is on a concave function, the GRM model satisfies the DF constraint according to Theorem 1.

EDCM: Xu and Akella [14] propose a mixture of eDCM distributions to model the pseudo relevance feedback set. Terms are then generated according to two latent generative models based on the (e)DCM distribution and associated with two variables, relevant z_{FR} and non-relevant z_N. The variable z_N is intended to capture general words occurring in the whole collection, whereas z_{FR} is used to represent relevant terms occurring in the feedback documents. Disregarding the non-relevant component for the moment, the weight assigned to feedback terms by the relevant component is given by (M-step of the EM algorithm):

$$P(w|z_{FR}) \propto \sum_{d \in \mathbf{F}} I(c(w,d) > 0) P(z_{FR}|d,w) + \lambda c(w,q)$$

This formula, being based on the presence/absence of terms in the feedback documents, is thus compatible with the DF constraint.

DFR Bo: Standard PRF models in the DFR family are Bo models [1]:

$$FW(w) = \text{Info}(w, \mathbf{F}) = \log_2(1 + g_w) + FTF(w) \log_2(\frac{1 + g_w}{g_w}) \qquad (4)$$

where $g_w = \frac{N_w}{N}$ in $Bo1$ model and $g_w = P(w|C)(\sum_{d \in F} l_d)$ in $Bo2$ model. In other words, documents in $\mathbf{F}$ are merged together and a geometric probability model is used to measure the informative content of a word. As this model is DF agnostic, it does not satisfy the DF constraint.

Log-logistic: In information-based models [2], the average information brought by the feedback documents on given term w is used as a criterion to rank terms, which amounts to:

$$FW(w) = \text{Info}(w, \mathbf{F}) = \frac{1}{n} \sum_{d \in \mathbf{F}} -\log P(X_w > tdfr(w,d)|\lambda_w)$$

where $tdfr(w,d)$ is given in table 1, and λ_w a parameter associated to w and set to: $\lambda_w = \frac{N_w}{N}$. Two instanciations of the general information-based family are considered in [2], respectively based on the log-logistic distribution and a smoothed power law (SPL). The log-logistic model for pseudo relevance feedback is thus defined by:

$$FW(w) = \frac{1}{n} \sum_{d \in F} [\log(\frac{N_w}{N} + tdfr(w,d)) + \text{IDF}(w)] \qquad (5)$$

It is straightforward to show that both the log-logistic and the SPL models lead to concave functions. So, according to Theorem 1, these models satisfies the DF constraint.

5 Well-Founded, Simple PRF Reweighting

Let us introduce the family of feedback functions defined by:

$$FW(w) = \sum_{d \in F} tdfr(w,d)^k \text{IDF}(w) \qquad (6)$$

with $tdfr$ is given in table 1 and corresponds to the normalization used e.g. in DFR and information-based models. This equation amounts to a standard *tf-idf* weighting, with an exponent k which allows one to control the convexity/concavity of the feedback model. If $k > 1$ then the function is strictly convex and, according to Theorem 1, does not satisfy the DF constraint. On the contrary, if $k < 1$, then the function is strictly concave and satisfies the DF constraint. The linear case, being both concave and convex, is *in-between*. One can then build PRF models from equation 6 with varying k, and see whether the results agree with the theoretical findings implied by Theorem 1. We used the reweighting scheme of equation 6 and a log-logistic model to assess their performance. The new query $q_w^{'}$ was updated as in DFR and information-based models:

$$q_w^{'} = \frac{q_w}{\max_w \ q_w} + \beta \frac{FW(w)}{\max_w \ FW(w)} \tag{7}$$

Figure 2 a) displays the term statistics ($\mu(ftf)$, $\mu(fdf)$, mean IDF) for different values of k. As one can note, the smaller k, the bigger $\mu(fdf)$ is. In other words, the slowlier the function grows, the more terms with large DF are preferred. Figure 2 b) displays the MAP for different values of k. At least two important points arise from the results obtained. First, convex functions ($k > 1$) have lower performance than concave functions for all datasets, and the more a model violates the constraints, the worse it is. This confirms the validity of the DF constraint. Second, the square root function ($k = 0.5$) has the best performance on all collections: it also outperforms the standard log-logistic model. When the function grows slowly (k equals to 0.2), the DF statistics is somehow preferred compared to TF. The square root function achieves a different and better trade-off between the TF and DF information. This is an interesting finding as it shows that the TF information is still useful and should not be too downweighted wrt the DF one.

Power k	$\mu(ftf)$	$\mu(fdf)$	Mean IDF
0.2	70.46	7.4	5.21
0.5	85.70	7.1	5.09
0.8	88.56	6.82	5.14
1	89.7	6.6	5.1
1.2	91.0	6.35	5.1
1.5	90.3	6.1	5.0
2	89.2	5.8	4.9

(a) Statistics on TREC-12-A

Power k	robust-A	trec-12-A	robust-B	trec-12-B
0.2	29.3	28.7	28.7	30.0
0.5	**30.1**	**29.5**	**29.4**	**30.5**
0.8	29.6	29.3	29.4	30.3
1	29.2	28.9	29.1	29.9
1.2	28.9	28.6	28.6	29.6
1.5	28.6	28.1	28.3	28.9
2	28.1	27.2	27.4	28.0
log-logistic	29.4	28.7	28.5	29.9

(b) MAP (%) for different power function

Fig. 2. (a) Statistics on TREC-12-A. (b) MAP (%) for different power function. Suffix A means $n = 10$ and $tc = 10$ while suffix B means $n = 20$ and $tc = 20$.

6 Related Work

There are a certain number of additional elements that can be used in PRF settings. The document score hypothesis states that documents with a higher score (defined by $RSV(q, d)$) should be given more weight in the feedback function as in relevance models [8]. Moreover, the study presented in [10], for example, proposes a learning approach to determine the value of the parameter mixing the original query with the feedback terms. In addition, the study presented in [12] focuses on the use of positional and proximity information in the relevance model for PRF, where position and proximity are relative to query terms. Again, this information leads to improved performance. Furthermore, the study presented in [5] for example proposes an algorithm to identify query aspects and automatically expand queries in a way such that all aspects are well covered.

Another comprehensive, and related, study is the one presented in [3,6]. In this study, a unified optimization framework is retained for robust PRF. Lastly, several studies have recently put forward the problem of uncertainty when estimating PRF weights [4,9]. These studies show that resampling feedback documents is beneficial as it allows a better estimate of the weights of the terms to be considered for feedback.

The study we have conducted here differs from the above ones as it aims at explaining, through a specific constraint, why some PRF systems work and others do not. Our experimental validation has revealed that the DF constraint is an essential ingredient to be used while designing PRF models, and our theoretical development has shed light on those models which or which do not comply to this constraint.

7 Conclusion

The main contributions of this paper are the formulation of the Document Frequency constraint and its validation. The performance of PRF models varies from one study to another, as different collections and different ways of tuning model parameters are often used. It is thus very diffcult to draw conclusions on the characteristics of such or such models. What is lacking to do so is a theoretical framework which would allow one to directly compare PRF models, independently of any collection. The theoretical analysis we conduct provides explanations on several experimental findings reported for different PRF models, and thus paves the way towards a theoretical assessment of PRF models.

First, two widely used models in the language modeling family, the simple mixture and the divergence minimization models, are deficient as one does not satisfy the DF constraint while the other does not sufficiently enforce the IDF effect. Second, the mixture of eDCM distributions [14], the geometric relevance model [13], the log-logistic and the smoothed power law models [2] were shown to satisfy the DF constraint. Hence, we argue that the DF constraint do capture the behavior of these recent models and yield an explanation to the obtained improvements. Finally, we have introduced a simple family of reweighting functions which allow to further compare the different ingredients of PRF models.

The experiments conducted with this family bring additional confirmation of the well-foundedness of the DF constraint.

References

1. Amati, G., Carpineto, C., Romano, G., Bordoni, F.U.: Fondazione Ugo Bordoni at TREC 2003: robust and web track (2003)
2. Clinchant, S., Gaussier, E.: Information-based models for *ad hoc* IR. In: SIGIR 2010, Conference on Research and Development in Information Retrieval (2010)
3. Collins-Thompson, K.: Reducing the risk of query expansion via robust constrained optimization. In: CIKM 2009 Conference on Information and Knowledge Management (2009)
4. Collins-Thompson, K., Callan, J.: Estimation and use of uncertainty in pseudo-relevance feedback. In: SIGIR 2007 (2007)
5. Crabtree, D.W., Andreae, P., Gao, X.: Exploiting underrepresented query aspects for automatic query expansion. In: SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007 (2007)
6. Dillon, J.V., Collins-Thompson, K.: A unified optimization framework for robust pseudo-relevance feedback algorithms. In: CIKM, pp. 1069–1078 (2010)
7. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: SIGIR 2004: Conference on Research and Development in Information Retrieval (2004)
8. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR 2001:Conference on Research and Development in Information Retrieval (2001)
9. Lee, K.S., Croft, W.B., Allan, J.: A cluster-based resampling method for pseudo-relevance feedback. In: SIGIR 2008, Conference on Research and Development in Information Retrieval (2008)
10. Lv, Y., Zhai, C.: Adaptive relevance feedback in information retrieval. In: Conference on Information and Knowledge Management, CIKM 2009 (2009)
11. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: CIKM 2009: Conference on Information and Knowledge Management, pp. 1895–1898 (2009)
12. Lv, Y., Zhai, C.: Positional relevance model for pseudo-relevance feedback. In: SIGIR 2010, Conference on Research and Development in Information Retrieval (2010)
13. Seo, J., Croft, W.B.: Geometric representations for multiple documents. In: SIGIR 2010: Conference on Research and Development in Information Retrieval (2010)
14. Xu, Z., Akella, R.: A new probabilistic retrieval model based on the dirichlet compound multinomial distribution. In: SIGIR 2008: Conference on Research and Development in Information Retrieval (2008)
15. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: CIKM 2001 (2001)

Model-Based Inference about IR Systems

Ben Carterette

Dept. of Computer & Info Sciences, University of Delaware, Newark, DE, USA
carteret@cis.udel.edu

Abstract. Researchers and developers of IR systems generally want to make inferences about the effectiveness of their systems over a population of user needs, topics, or queries. The most common framework for this is statistical hypothesis testing, which involves computing the probability of measuring the observed effectiveness of two systems over a sample of topics under a null hypothesis that the difference in effectiveness is unremarkable. It is not commonly known that these tests involve *models* of effectiveness. In this work we first explicitly describe the modeling assumptions of the t-test, then develop a Bayesian modeling approach that makes modeling assumptions explicit and easy to change for specific challenges in IR evaluation.

1 Introduction

Arguably the fundamental problem in IR is modeling the relevance of information to users. Vast amounts of effort have gone into developing features, model families, and optimization methods that can model relevance in a way that produces systems that are useful to users. Nearly as important is modeling the actual utility of these systems to the users that are supposed to benefit from them. This is the *effectiveness evaluation* problem, and it is traditionally based on a combination of effectiveness measures to estimate utility and statistical hypothesis testing to make inferences about the relative utility of different systems. But while IR research on relevance modeling looks into all aspects of the problem, much of the IR literature on hypothesis testing defers to basic work on statistics that is written conservatively, with the goal of providing solutions that make the fewest and weakest assumptions so as to be applicable to the widest range of cases. We argue that better inference is possible if we tailor our tests to the particular challenges of evaluating IR.

Just as relevance and retrieval models are based on features (of queries, of documents, of query/document pairs, of users, etc), evaluation by statistical hypothesis test is based on features as well. But while relevance models can be extraordinarily complex, evaluation models have remained very simple. Every significance test in wide use models an evaluation measure with at most two "features" along with an intercept and residual error. The models and features used in evaluation models are almost always hidden from the practitioner, however. Unless one has a deep familiarity with the t-test—a familiarity beyond what is presented in introductory textbooks—one may not be aware that it is

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 101–112, 2011.

equivalent to performing inference in a simple linear model with categorical system ID and topic ID features. There is much room to tailor this model to specific evaluation scenarios in IR, but no statistics textbook will explain how to do that.

This paper proposes an explicitly model-based Bayesian framework for hypothesis testing and evaluation in general. Bayesian models have of course been used to model relevance, but to the best of our knowledge they have not been used in IR effectiveness evaluation. The advantage of the Bayesian framework is that it allows construction of tailored models for evaluation largely free from the details of how to perform inference in them (unlike the t-test). Bayesian inference is arguably more intuitive as well, as it comes down to the probability of a hypothesis being true rather than the probability of observing data given a null hypothesis (the p-value).

We begin in Section 2 by describing the use of models in traditional IR evaluation. We then present the Bayesian approach in Section 3, with Bayesian versions of the t-test along with a new Bayesian method for direct inferences about system effectiveness using relevance generated by a user model. In Section 4 we empirically analyze the Bayesian framework.

2 Traditional Model-Based Inference

In this section we summarize some previous work on models in evaluation and show how we use modeling assumptions in evaluation even though they may not be explicitly stated. Broadly speaking, models come into play at two points in evaluation: first, in effectiveness measures that model user utility or satisfaction, and second, in statistical tests of the significance of differences. A third way models come into play is in the use of data mined from interaction logs for evaluation, but that is outside the scope of this work.

2.1 Model-Based Evaluation Measures

There has always been interest in using effectiveness measures to model and approximate utility to a user. Recent work along these lines often involves constructing an explicit probabilistic user model, then combining it with relevance judgments to summarize utility [2]. Some examples are *rank-biased precision* (RBP) [9], *discounted cumulative gain* (DCG) [7], *expected reciprocal rank* [3], *expected browser utility* (EBU) [16], α-nDCG [4], and others [12,1]. In addition, traditional measures have had user models backfit to their mathematical expression [11,17], showing that most measures at least suggest a user model.

In this work we will focus on just two measures: precision at rank k, modeling a user that will stop at rank k and derive utility from the relevant documents appearing about that rank; and RBP, modeling a user that steps down a ranked list, deriving decreasing utility from each subsequent relevant document. RBP can be expressed as

$$RBP = \sum_{k=1}^{\infty} y_k \theta^{k-1}(1-\theta)$$

The reason for focusing on these two is that they have simple, clear user models, and that they have light computational requirements as compared to measures like average precision (AP). Computational requirements are an unfortunate drawback to some of the methods we propose, particularly those in Section 3.2.

2.2 Statistical Hypothesis Tests

Statistical hypothesis tests use the idea that a set of topics is a sample from some larger population to test the hypothesis that a difference between two systems is "real" and cannot be ascribed to random chance. Many different tests are used by IR researchers; the most common are the sign test, the Wilcoxon signed rank test, the t-test, ANOVA, the randomization (exact) test, and the bootstrap test [13,18]. Every test is in one way or another based on a model; every model has assumptions. The model and its assumptions are usually hidden to the practitioner, but understanding the model is key to understanding the test.

For this work we will focus on the t-test. We assume the basics of the t-test are well-known and not reiterate them here. Our interest is in its modeling assumptions: it is actually based on a linear model of a measure of the effectiveness y_{ij} of system j on topic i as a linear combination of an intercept μ, a "system effect" β_j, a "topic effect" α_i, and random error ϵ_{ij} that is assumed to be normally distributed with variance σ^2 [10]. The t-test model is therefore:

$$y_{ij} = \mu + \beta_j + \alpha_i + \epsilon_{ij}$$
$$\epsilon_{ij} \sim N(0, \sigma^2)$$

In this notation y_{ij} is equal to a sum of effects, one of which (the errors ϵ_{ij}) are drawn from a normal distribution with mean zero and variance σ^2 (as indicated by the $\sim N(0, \sigma^2)$ notation). We can equivalently express this model as:

$$\widehat{y_{ij}} = \mu + \beta_j + \alpha_i$$
$$y_{ij} \sim N(\widehat{y_{ij}}, \sigma^2)$$

and even more compactly as:

$$y_{ij} \sim N(\mu + \beta_j + \alpha_i, \sigma^2)$$

Note that this is exactly the same linear model that is the basis of linear regression, and in fact it is the same linear model that is the basis of ANOVA as well. ANOVA is a special case of linear regression, and the t-test is a special case of ANOVA. This is not well-known among IR practitioners; we refer to Monahan [10], Gelman et al. [6], and Venables & Ripley [14] for deeper treatment of linear models from different perspectives.

Performing a t-test therefore involves estimating parameters $\mu, \beta_j, \alpha_i, \sigma^2$. In practice, a paired t-test only requires estimates the magnitude of the difference between two system effects $(\beta_1 - \beta_2)$ and the error variance σ^2. The maximum likelihood estimates of $\beta_1 - \beta_2$ and σ^2 are the mean difference and variance of differences in measure values respectively. If topics are sampled independently

and identically (i.i.d.), the Central Limit Theorem says the estimate of $\beta_1 - \beta_2$ can be treated as having a normal distribution, and therefore $(\beta_1 - \beta_2)/\sqrt{\sigma^2/n}$ has a Student's t distribution with $n-1$ degrees of freedom.

This statement of the t-test as a linear model makes its assumptions explicit:

1. errors ϵ_{ij} are normally distributed with mean 0 and variance σ^2 (normality);
2. variance σ^2 is constant over systems and topics (homoskedasticity);
3. effects are additive and linearly related to y_{ij} (linearity);
4. topics are sampled i.i.d. (independence).

The first three of these assumptions are almost certainly *false* in typical IR experiments. The reason is that IR effectiveness measures are discrete-valued and bounded in the range $[0, 1]$. Consider each assumption in turn:

1. Normality: normal distributions are unbounded, so our error distribution will give non-zero probability to values outside the range of the measure.
2. Homoskedasticity: very bad and very good systems necessarily have lower variance than average systems, simply because as the measure approaches 0 or 1 there are fewer ways it can vary.
3. Linearity: $\widehat{y_{ij}}$ can be outside the range $[0, 1]$ because there is no bounding of the linear combination. Also, there is at least one measure that is surely non-linearly related to topic effect (recall) and one that is non-additive (GMAP).

In this work we are not terribly concerned with the effect of these violations—in fact, the t-test is quite robust to them. Our point is to state them clearly so that we can begin to form alternative models for evaluation that more precisely capture aspects of IR effectiveness that are not captured by the t-test, and so that we can state exactly how the models differ from each other.

Non-parametric tests rely on modeling assumptions as well, though they are typically weaker than those of the linear model. Even tests like the randomization and bootstrap tests rely on modeling assumptions that may be false: both of those tests relax homoskedasticity to a weaker assumption of *exchangeability*, and trade the Gaussian error distribution for an empirical error distribution. They still assume a linear model and independence of topic effects.

3 Bayesian Inference

Our aim is to find a framework for testing hypotheses about systems that can be adopted for the specific challenges of IR evaluation. Our first effort is towards explicit model-based hypothesis testing: we introduce a fully Bayesian version of the linear model we presented above. We then introduce greater and greater complexity to show what the Bayesian framework can do.

3.1 Bayesian Linear Model

As discussed above, the t-test assumes a linear model with three effects and normally-distributed errors. For our Bayesian version, we will start with the

same two assumptions. We will also introduce *prior distributions* for each model parameter. These prior distributions can be used to model any information we already have about the experiment. If we do not wish to make any strong assumptions, we can use non-informative priors. An example non-informative prior might be a uniform distribution over the entire real line. Better is a normal distribution with uncertain variance—i.e. the variance itself is a parameter with a prior distribution.

Thus our first attempt at a fully-Bayesian model is:

$$\begin{aligned} \widehat{y_{ij}} &= \mu + \beta_j + \alpha_i \\ y_{ij} &\sim N(\widehat{y_{ij}}, \sigma^2) & \sigma &\sim 1/\sigma \\ \mu &\sim N(0, \sigma^2_{\text{int}}) & \sigma_{\text{int}} &\sim 1/\sigma_{\text{int}} \\ \beta_j &\sim N(0, \sigma^2_{\text{run}}) & \sigma_{\text{run}} &\sim 1/\sigma_{\text{run}} \\ \alpha_i &\sim N(0, \sigma^2_{\text{topic}}) & \sigma_{\text{topic}} &\sim 1/\sigma_{\text{topic}} \end{aligned}$$

In words, each measure of system effectiveness on a topic is a sum of a population effect μ, a system effect β_j, and a topic effect α_i. Since we do not know anything about these effects *a priori*, we put prior normal distributions over them. Since we further do not know anything about the variances of those distributions, we use improper flat priors on the log scale (the non-informative Jeffreys prior).

To make inferences about systems, we need the posterior distribution of system effects: $P(\beta_j|y)$, where y is the effectiveness evaluation data. Obtaining the posterior distributions is best done by simulation, iteratively sampling parameters from their prior distributions, then updating posteriors based on the likelihood of the data. Monte Carlo Markov Chain simulation is a standard technique. We do not provide details here, as there are packages for general MCMC computation of posteriors available.

Once the posteriors have been computed, making inferences about the systems is relatively simple: we estimate the probability of a hypothesis such as $S_1 > S_2$ by estimating $P(\beta_1 > \beta_2)$ from our simulation data. Note that the Bayesian approach actually estimates the probability that a hypothesis is true (conditional on the model and the data) rather than the probability of observing the data under a null model (like the t-test and other traditional tests). We feel this has the additional advantage of being more intuitive than a p-value.

3.2 Direct Inference about Relevance

Effectiveness measures are themselves summaries of individual measurements on documents—relevance judgments. Instead of testing a hypothesis about a summarization of judgments by an effectiveness measure, the Bayesian framework allows us to model relevance *directly* according to some user model.

Let x_{ijk} be the judgment to the document retrieved by system j at rank k for topic i. We will model the judgments as coming from a Bernoulli distribution with parameter p_{ij}, essentially a coin flip biased by the system and topic. We will model p_{ij} using the linear model with a population effect, a system effect,

and a topic effect, filtered through a sigmoid function to ensure the result is bounded between 0 and 1.

$$
\begin{aligned}
x_{ijk} &\sim \text{Bernoulli}(p_{ij}) \\
p_{ij} &= \exp(y_{ij})/(1+\exp(y_{ij})) \\
y_{ij} &\sim N(\mu+\beta_j+\alpha_i, \sigma^2) & \sigma &\sim 1/\sigma \\
\mu &\sim N(0, \sigma^2_{\text{int}}) & \sigma_{\text{int}} &\sim 1/\sigma_{\text{int}} \\
\beta_j &\sim N(0, \sigma^2_{\text{run}}) & \sigma_{\text{run}} &\sim 1/\sigma_{\text{run}} \\
\alpha_i &\sim N(0, \sigma^2_{\text{topic}}) & \sigma_{\text{topic}} &\sim 1/\sigma_{\text{topic}}
\end{aligned}
$$

While we still have y_{ij} in the model, it should no longer be thought of as a measure of effectiveness. Now it is a hidden variable that influences the probability that a document appearing at a particular rank is relevant. p_{ij} is a convenience variable that converts the real-valued y_{ij} to a probability in $[0,1]$ by applying the sigmoid function.

As it turns out, however, y_{ij} can be congruent to an *estimate* of an effectiveness measure. If we restrict ranks to $k \leq K$ (for some constant K) the parameter p_{ij} is an estimate of the precision at rank K of system j on topic i. To see this, think of precision as the expectation that a randomly-chosen document in the set of K is relevant. That expectation is $\sum_{k=1}^{K} x_{ijk} p_{ij}$; the maximum-likelihood estimate of p_{ij} is $1/K \sum_{k=1}^{K} x_{ijk}$, which is precision. Thus our explicit model of the relevance judgments produces precision at rank K, and y is just the log-odds of that precision. Furthermore, we can do inference on precision using the β_j parameters just as we would in a t-test or ANOVA.

Other models give rise to other evaluation measures. Suppose we sample a rank k with probability p_k, and model x_{ijk} as

$$P(x_{ijk}) = p_{ij} p_k$$

Now p_{ij} is still a Bernoulli distribution parameter, but depending on how we define p_k, p_{ij} will be an estimate of utility. If $p_k = 1$ for $k \leq K$ and 0 for $k > K$, p_{ij} estimates precision at K. If p_k is a geometric distribution (that is, $p_k = \theta^{k-1}(1-\theta)$, then p_{ij} will be an estimate of RBP. This fits with our formulation in previous work, meaning many other measures can fit in this framework [2].

3.3 Modeling Other Evidence

Once the models are explicit, and computation/inference is divorced from model structure and assumptions, we can easily incorporate other sources of evidence without having to find new methods for inference. This is a limitation of traditional methods such as the t-test; the inference is strongly tied to the model structure and assumptions.

In this section we adopt a more "intuitive" notation for our models; rather than express a model as a sum of variables, we express it in words as a sum of effects. Our simple linear models above will be:

$$y_{ij} = \mu + \text{system}_j + \text{topic}_i + \epsilon_{ij}$$

This notation is meant to reduce the Greek letter assignment problem: as we add more sources of variance, we also add interactions between them, and there is a resulting combinatorial explosion in coefficients.

As an example of incorporating another source of variance, suppose we suspect that which assessor is assigned to which topics may explain something about the evaluation. We can incorporate assessor as another effect in the linear model:

$$y_{ijk} = \mu + \text{system}_j + \text{topic}_i + \text{assessor}_k + \text{topic}_i \times \text{system}_j + \text{topic}_i \times \text{assessor}_k + \text{system}_j \times \text{assessor}_k + \epsilon_{ijk}$$

where k is an integer identifying the assessor that worked on topic i. We also add interaction effects (denoted as $\mathit{effect}_1 \times \mathit{effect}_2$) to model any possible bias that an assessor might have for a system or topic. We use normal priors with log-normal priors on the variance parameters for all of these interaction effects as well as the assessor effect; these are omitted for space. The interaction between all three effects is subsumed by the errors, so it is not necessary to model it explicitly.

Note that assessor variance is not easy to model in a traditional ANOVA/t-test linear model: we have repeated measures of systems on topics (so we can do paired or within-group analysis), but we generally do not have repeated measures of assessors on topics (so we can only do unpaired or between-group analysis). Combining within-group and between-group analyses requires a whole other generalization of the linear model in classical statistics; in Bayesian statistics there is essentially no issue.

As another example, suppose we are interested in the effect of different corpus filters on results. We could incorporate that into the model easily:

$$y_{ijk} = \mu + \text{system}_j + \text{topic}_i + \text{corpus}_k + \text{topic}_i \times \text{system}_j + \text{topic}_i \times \text{corpus}_k + \text{system}_j \times \text{corpus}_k + \epsilon_{ijk}$$

In general, we can add any number of additional effects along with interactions between them. While this leads to a combinatorial explosion in the number of coefficients to estimate, which in turn requires an exponential amount of data in the traditional ANOVA/t-test model, the Bayesian approach does not suffer as the traditional approach would. Given little evidence for estimating a kth-order interaction effect, the Bayesian approach simply falls back to the prior and says that it cannot conclude with any confidence that the interaction effect is not a significant source of variance.

4 Empirical Analysis

Our main proposal in this work is that Bayesian models are a powerful alternative to traditional tools for evaluation and statistical analysis such as the t-test. We cannot prove that Bayesian versions are superior: empirically, there is no gold standard against which we can compare the inferences from different approaches

to show that one is more accurate on average; theoretically, both are valid. We can only show when the two approaches agree and when they disagree, and argue from principles about the relative cost of the disagreements.

4.1 Experimental Set-Up

Our data is retrieval systems submitted to TREC tracks over the years. Since we are concerned with significance testing, it does not really matter what data we use. We picked the 74 runs submitted to the TREC-6 ad hoc track for a large set of runs on a relatively small corpus that can be broken up into more homogeneous chunks for modeling (non-random) corpus effects. TREC-6 also has a second set of *qrels* from work done at the University of Waterloo [5]; we can use this to demonstrate modeling assessor effects.

In all of our analyses below we test a one-sided hypothesis about two systems. The null hypothesis is that the system S_1 is better than S_2 by some measure.

$$H_0 : S_1 \geq S_2$$
$$H_a : S_1 < S_2$$

The one-sided null hypothesis is generally the hypothesis practitioners are interested in, and we argue it is more "natural" than the two-sided point null hypothesis of $S_1 = S_2$. We test these hypotheses with two measures: precision at rank 10 and rank-biased precision (RBP) with $\theta = 0.8$. We compare the p-value for rejecting H_0 to the Bayesian posterior probability of H_a being true—this means that *higher* Bayesian probabilities correspond to *lower* p-values.

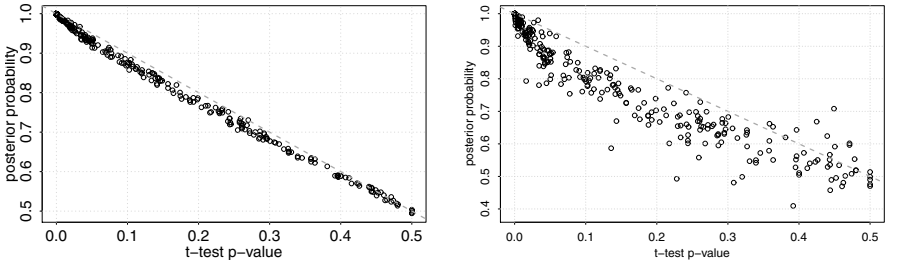
We use JAGS (Just Another Gibbs Sampler, an open-source implementation of BUGS for MCMC sampling to compute posteriors in Bayesian models) and its R interface `rjags` to implement the models we describe above. JAGS allows a user to write a "model file", a programmatic description of the model which is parsed into a full simulation program. `rjags` executes the simulation, computing posterior distributions conditional on data objects in R. All of our model files and R code can be downloaded from `ir.cis.udel.edu/~carteret/testing.html`.

Because it requires simulation, Bayesian testing is much more computationally-intensive than classical testing. Rather than test all 2,701 pairs of runs, we sampled a subset of 500 pairs to test. All results are based on the same sample.

4.2 Classical T-Test vs. Bayesian Linear Model vs. Direct Inference

Here we show that p-values from classical t-tests correlate well with posterior probabilities from Bayesian tests.

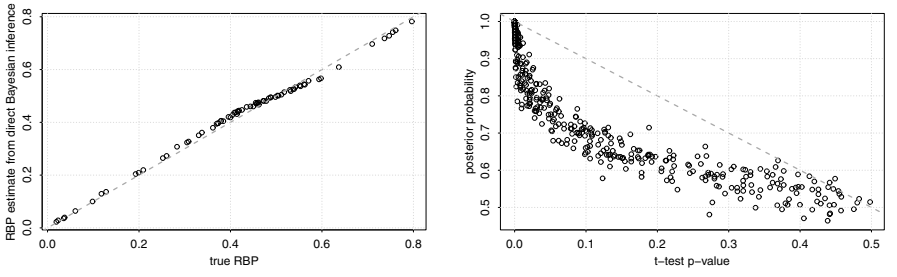
Figure 1(a) compares p-values from a one-sided paired t-test for a difference in precision to posterior probabilities from the Bayesian linear model we presented in Section 3.1. Note that they are highly correlated, though not quite identical. The Bayesian posterior probabilities are slightly more conservative than the t-test p-values, most obviously as they approach the boundary; this is due to the use of noninformative priors. In the Bayesian framework it takes extraordinary



(a) Comparison to linear model posterior probability

(b) Comparison to relevance model posterior probability

Fig. 1. Comparison of one-sided paired t-test p-values for a difference in precision@10 to Bayesian posterior probabilities from two different models: the traditional linear model (left) and the direct model of relevance (right)



(a) True RBP (with $\theta = 0.8$) agains RBP estimate p_{ij}

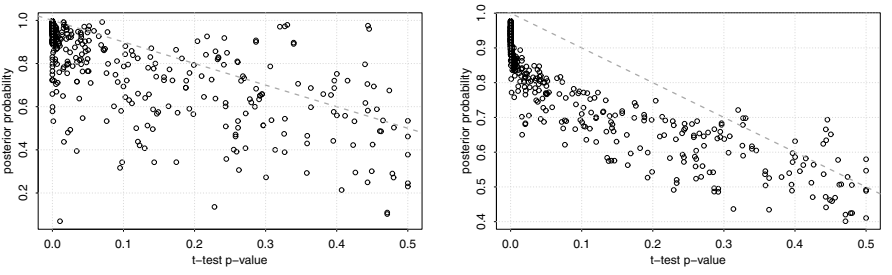
(b) t-test p-values and Bayesian posterior probabilities from direct model

Fig. 2. Using RBP's user model as part of a direct model of relevance results in accurate estimates of RBP (left), but less confidence in conclusions about H_a

evidence to draw the extraordinary conclusion that one system is better than the other with probability close to 1.

Figure 1(b) compares p-values from the one-sided paired t-test to posterior probabilities from the Bayesian direct inference model we presented in Section 3.2. We now see that tailoring the model to the particular data we have in IR and a particular user model has a stronger effect on inferences. The posterior probabilities are generally much more conservative than the t-test p-values.

In Section 3.2 we claimed that the parameter p_{ij} can estimate an effectiveness measure based on a user model. Figure 2(a) shows that in the direct inference approach with the RBP user model, p_{ij} indeed gives a good estimate of RBP: the estimates are almost perfectly correlated with true RBP. The posterior probabilities, however, are substantially more conservative than the p-values from a t-test (Figure 2(b)). This suggests that there may be many other reasons for documents to be ranked as they are apart from basic system and topic effects. It



(a) Linear model with corpus effects (b) Linear model with assessor effects

Fig. 3. Comparison of one-sided paired t-test p-values to Bayesian posterior probabilities from models that include additional effects

also suggests that by giving so much weight to the top-ranked documents, RBP makes it difficult to draw general conclusions about differences between systems.

4.3 Advanced Analysis

We simulated evaluation over multiple corpora by splitting each TREC-6 submitted run into separate runs by collection: for each of the Congressional Records, Federal Register, Financial Times, Foreign Broadcast Information Service, and LA Times collections, we created 74 new runs consisting of only documents in that collection (ranked in the same order as the original run). Thus with five collections we have a total of $74 \times 5 = 370$ systems[1]. Since some systems did not retrieve any documents from some collections for some topics, we have unbalanced data—this is a case that is hard for traditional methods to deal with, but the Bayesian approach can solve painlessly.

One-sided t-test p-values and Bayesian posterior probabilities from the model in Section 3.3 are shown in Figure 3(a). Although the relationship looks random by inspection, agreement is actually quite high—the linear correlation is -0.7, meaning the posterior probability of H_a is high when the chance of rejecting H_0 is high. But there are many cases in which taking corpus into account substantially changes the inference about systems. The most extreme case is the point in the lower left; the t-test concludes that S_2 is better, while Bayesian analysis taking corpus effects into account concludes that S_1 is better, and both inferences have high confidence. The first system actually has much better retrieval results on each corpus individually, but managed to interleave results in such a way that its final results are much worse. This is a formative conclusion that traditional statistical analysis could not tell us.

To test whether assessors had any effect, we evaluated all systems and topics using both sets of judgments available for the TREC-6 topics. Figure 3(b) shows

[1] We note the implicit assumption that the systems ranked documents for each collection using corpus statistics computed from all collections together. This is not very realistic, but we think the example is still illustrative.

the relationship between t-test p-values and posterior probabilities when assessor set is part of the model. As in Fig. 1(a), we still "believe" H_a is true in most cases—meaning assessors have little effect on whether we reject or accept H_a, confirming previous work [15]—but we have significantly less confidence. This is because there are more parameters to estimate in the model, and therefore less confidence in the hypothesis with the same amount of data.

5 Conclusion

We have introduced a Bayesian framework for inferences about IR systems. The advantage of this framework is that *all* models—from the user model in the effectiveness measure to the topic population model in the significance test—are made explicit, revealing all assumptions and opening them to refinement or correction. Since computation is largely divorced from model structure and assumptions, assumptions can be changed easily without developing new methods for inference. We showed how an evaluation model can be seamlessly combined with a user model for more user-centered system-based evaluation, and how many more factors affecting effectiveness can be incorporated into the evaluation model; both of these subjects are too big for a detailed treatment here, but we intend to follow up on both in future publications.

Because models are explicit, using this framework in a variety of evaluation scenarios is mostly a matter of building the model. For low-cost evaluation settings, we can model missing judgments. For settings with graded judgments, we can use multinomial distributions instead of Bernoulli trials, or user models that probabilistically map grades to binary judgements [12]. Tasks such as novelty/diversity [4] or sessions [8] simply involve creating new models of user utility. Furthermore, the models can be directly informed by logged user data by using that data to compute posterior distributions.

The tradeoff of increased transparency and power is decreased clarity. We concede that it can be difficult to look at the models in Section 3.2 and easily understand what is being modeled and how. Furthermore, computation is much more arduous (not to mention less transparent), and inferences are subject to simulation error, much like randomization and bootstrap tests.

Nevertheless, the framework is so powerful and flexible that we believe these tradeoffs are worthwhile. The inferences are close enough that the practitioner can still use t-tests for the basic paired experiments that are common in IR. But when more advanced analysis is required, our Bayesian model-based framework seems to be the solution.

References

1. Agrawal, R., Gollapudi, S., Halverson, H., Ieong, S.: Diversifying search results. In: Proceedings of WSDM 2009, pp. 5–14 (2009)
2. Carterette, B.: System effectiveness, user models, and user utility: A conceptual framework for investigation. In: Proceedings of SIGIR (to appear, 2011)

3. Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P.: Expceted reciprocal rank for graded relevance. In: Proceedings of the Annual International ACM Conference on Knowledge and Information Management, CIKM (2009)
4. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: Proceedings of SIGIR 2008, pp. 659–666 (2008)
5. Cormack, G.V., Palmer, C.R., Clarke, C.L.A.: Efficient construction of large test collections. In: Proceedings of SIGIR, pp. 282–289 (1998)
6. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. Chapman & Hall/CRC, Boca Raton (2004)
7. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (2002)
8. Kanoulas, E., Carterette, B., Clough, P.D., Sanderson, M.: Evaluation over multi-query sessions. In: Proceedings of SIGIR (to appear, 2011)
9. Moffat, A., Zobel, J.: Rank-biased precision for measurement of retrieval effectiveness. ACM Trans. Info. Sys. 27(1), 1–27 (2008)
10. Monahan, J.F.: A Primer on Linear Models, 1st edn. Chapman and Hall/CRC, Boca Raton (2008)
11. Robertson, S.E.: A new interpretation of average precision. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 689–690 (2008)
12. Robertson, S.E., Kanoulas, E., Yilmaz, E.: Extending average precision to graded relevance judgments. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 603–610 (2010)
13. Smucker, M., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of CIKM, pp. 623–632 (2007)
14. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S, 4th edn. Springer, Heidelberg (2002)
15. Voorhees, E.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: Proceedings of SIGIR, pp. 315–323 (1998)
16. Yilmaz, E., Shokouhi, M., Craswell, N., Robertson, S.: Expected browsing utility for web search evaluation. In: Proceedings of the ACM International Conference on Knowledge and Information Management (to appear, 2010)
17. Zhang, Y., Park, L.A., Moffat, A.: Click-based evidence for decaying weight distributions in search effectiveness metrics. Inf. Retr. 13, 46–69 (2010)
18. Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: Proceedings of SIGIR, pp. 307–314 (1998)

Selecting a Subset of Queries for Acquisition of Further Relevance Judgements

Mehdi Hosseini[1], Ingemar J. Cox[1], Natasa Milic-Frayling[2], Vishwa Vinay[2], and Trevor Sweeting[1]

[1] University College London
{m.hosseini,ingemar}@cs.ucl.ac.uk, {trevor}@stats.ucl.ac.uk
[2] Microsoft Research Cambridge
{natasamf,vvinay}@microsoft.com

Abstract. Assessing the relative performance of search systems requires the use of a test collection with a pre-defined set of queries and corresponding relevance assessments. The state-of-the-art process of constructing test collections involves using a large number of queries and selecting a set of documents, submitted by a group of participating systems, to be judged per query. However, the initial set of judgments may be insufficient to reliably evaluate the performance of future as yet unseen systems. In this paper, we propose a method that expands the set of relevance judgments as new systems are being evaluated. We assume that there is a limited budget to build additional relevance judgements. From the documents retrieved by the new systems we create a pool of unjudged documents. Rather than uniformly distributing the budget across all queries, we first select a subset of queries that are effective in evaluating systems and then uniformly allocate the budget only across these queries. Experimental results on TREC 2004 Robust track test collection demonstrate the superiority of this budget allocation strategy.

1 Introduction

In information retrieval (IR), a test collection is used to evaluate the performance of search systems. A test collection consists of (*i*) a document corpus, (*ii*) a set of topics with corresponding search queries, and (*iii*) a set of relevance judgments for each query. Relevance judgments indicate which documents in the corpus are relevant to a particular query. When the corpus and the number of queries are small, it is feasible to acquire relevance judgments by employing a number of human assessors and, possibly, judge the relevance of each document in the collection to all the queries. However, when the corpus and the number of test queries are large, this is no longer the case due to both the economic cost and the time involved.

In order to address this issue, the IR community has adopted a method of *pooling* candidate documents from the retrieval results of the participating systems [11]. Each participating system contributes a set of documents it retrieves for a query, e.g. the top-100 documents, and the set of unique documents is then

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 113–124, 2011.

judged for relevance by the human assessors. The total number of pooled documents is typically much smaller than the number of documents in the corpus. However, since documents are provided by systems that are being compared, the resulting document pool is expected to be effective in assessing their relative performance [14].

Gathering relevance assessments has an associated cost which, in its simplest form, depends on the number of queries and the number of documents per query that need to be assessed. However, the cost is not the only consideration when creating effective test collections. The accuracy and reusability of the test collections are also very important. A test collection is accurate if the participating systems' performance are precisely evaluated. In addition, a test collection is reusable if has no inherent bias that might affect evaluation of new as yet unseen systems.

In particular, a test collection may not be reusable if a new system, in response to queries in the test set, retrieves many documents that are not in the document pool. In this situation, (*i*) the previously unjudged documents must either be judged non-relevant [12], (*ii*) the new documents are assigned a probability of relevance and new systems' performance are measured by using metrics designed for incomplete relevance judgments, e.g. MTC [3], or (*iii*) additional user relevance judgments must be obtained for these documents. Assuming the documents are non-relevant potentially biases the test collection - only future systems that behave like the original participating systems will be evaluated accurately [5]. Assigning a probability of relevance may cause a high uncertainty in evaluation when there are a large number of unjudged documents for new systems [4], and acquiring additional user judgments can be expensive.

We assume a limited budget is available to build additional relevance judgements for previously unjudged documents retrieved by new systems. In this paper, we examine whether it is better to uniformly allocate the budget across all queries, or select a subset of queries and allocate the budget only to the selected queries to get deeper judgments per query at the same cost.

Selection of the subset is strongly related to the query selection problem. The query selection problem is motivated by empirical evidence that the ranking of systems based on many queries can be reproduced with a much reduced set of queries [8]. Thus, ideally, we would identify a minimal subset of queries that still enables a reliable evaluation of the existing and new systems. Furthermore, the gain from reducing the number of queries can be re-directed to increase the number of documents judged per query. Our hypothesis is that, given a fixed budget, a smaller but representative set of queries with a greater number of judged documents per query will increase the accuracy of ranking new systems.

In this paper, we introduce a query selection approach and present results of its application to obtain an accurate ranking of new systems' performance. In Section 3, we formalize the query selection problem and propose two query selection algorithms, a *greedy* algorithm and an optimization based on a *convex* objective. Section 4 then describes our experimental results, based on the Robust track of TREC 2004. Two different experiments are described. The first

experiment is concerned with how a subset of queries performs when evaluating new systems that did not participate in the original pooling (generalization). The second experiment examines the problem of allocating new relevance judgments to queries, and compares a uniform allocation across all queries to a (deeper) uniform allocation across a subset of queries. Finally, Section 5 concludes by summarizing the results of our research and outlining future directions. However, before proceeding we first discuss related work.

2 Related Work

Research presented in this paper starts with [13], which suggests that reliable IR evaluation requires at least 50 queries and that including a larger number of queries makes for a better test collection. Given these results, it is not surprising that recent studies have concentrated on IR evaluations with large query sets [6], and methods for reducing the number of relevance judgments per query in order to make relevance assessment feasible [3], as well as introducing evaluation metrics for partially judged result sets [2].

Following the belief that a larger query set is desirable, the Million Query track of TREC 2007 [1] was the first to include thousands of queries. The Million Query track used two document selection algorithms, proposed by [3] and [2], to acquire relevance judgments for more than 1,800 queries. The experiments on this test collection showed that a large number of queries with a few judgements (i) results in an accurate evaluation of participating systems, and (ii) is more cost-effective than evaluation conducted by fewer queries with more judgements. However, due to the small number of documents assessed per query, the reusability of such a test collection still remains questionable. Indeed, Carterette et al. [5] demonstrated that the Million Query track of TREC 2009 is not usable for assessing the performance of systems that did not participate in pooling documents.

Guiver et al. [8] showed that some queries or query subsets are better in predicting systems' overall performance than others. Finding a representative subset of queries is a combinatorial problem with NP-hard complexity. Little work is available on practical approaches that could be used to select a subset. Mizzaro and Robertson [9] suggested prioritizing queries based on a per-query measure, hubness, that indicates how well a query contributes in system evaluation. Guiver et al. [8] showed that a representative subset consists of not only queries that individually predict systems' performance with high precision but also queries that are weak in prediction on their own. They also suggested a greedy algorithm for query subset selection. However, Robertson [10] showed that a subset that is selected by the greedy algorithm suffers from overfitting and is not able to generalize to new systems. In this paper, we propose a convex method for query selection and show that its generalization is superior to the greedy algorithm.

3 Expanding Relevance Judgements

A test collection consists of a document corpus, a set of N queries $Q_N = \{q_1, q_2, ..., q_N\}$, and the associated relevance judgements that are gathered based on documents returned by a set of L participating systems, S_L. More precisely, each of the systems returns a number of results for each of the N queries. A pooling technique or a recently proposed document selection method, e.g. [3], is used to select a subset of documents returned by each of the L systems to build relevance judgements. The aim of the test collection is not only to accurately evaluate the performance of the participating systems but also to reliably estimate the performance of new systems that did not participate in pooling.

We begin with the assumption that a system can be reliably evaluated and compared with other systems if we manually assess a significant portion of the document corpus or, at least, a large number of documents retrieved by each individual system. Therefore, if new systems return many new (unjudged) documents, the current relevance judgements are insufficient to reliably assess their performance. In this situation, we assume that there is a limited budget to build relevance judgements for a subset of the new documents. How should we spend the limited budget to acquire additional relevance judgments? We could consider all queries and use a heuristic method to pool a few documents per query. Alternatively, we could select a representative subset of queries that closely approximates systems' overall performance, and allocate the budget only to the selected queries. The final solution is likely to include elements of both these approaches. In this paper, we assume the pooling method [11] is used to select documents at the query level and restrict our attention to the task of choosing queries. In the following, we formalize the query selection problem and describe three solutions.

3.1 The Query Selection Problem

Evaluating the L systems on the N queries forms a $L \times N$ performance matrix X. Each row represents a system, and each column a query. An entry, $x_{i,j}$, in X denotes the performance of the i^{th} system on the j^{th} query. We also consider the column vector M, as the average performance vector. The values of M indicate the average performance of individual systems over all queries. Thus, if the individual elements, $x_{i,j}$, measure average precision (AP), then the elements of M represent mean average precision (MAP) scores.

Now let $\Phi = \{j_1, ..., j_m\}$ be a subset of $\{1, 2, ..., N\}$ with $1 \leq m \leq N$ and consider the subset of queries $\{q_j : j \in \Phi\}$. We define M_Φ as the vector of systems' average performance measured on the subset of queries indexed in Φ. The aim of a query selection method is to find a subset of queries of a particular size, m, such that the corresponding vector M_Φ closely approximates the vector M. There are several measures to evaluate how well M_Φ approximates M. The IR community usually uses Kendall-τ rank correlation coefficient to measure the closeness between two vectors of real values. In this paper, we assume that the

objective of a query selection method is to maximize the Kendall-τ coefficient between the systems' rankings induced by M_Φ and M.

Finding the subset of queries of size m that maximizes this objective is NP-hard, and a brute force search is only practical for small N [8]. In the following, we introduce three computationally practical selection algorithms that approximate the optimal solution.

Random Sampling: One may use uniform random sampling to select a subset of queries. In this method, all queries are given the same chance to be selected. We use uniform random sampling as the baseline in our experiments. That is, for a given subset size, m, we randomly select a subset of m queries.

Greedy Algorithm: A forward selection scheme can be used to approximate the optimal subset of queries. That is, when m=1, the optimal solution is the query whose column score in matrix X leads to a systems' ranking that has the highest Kendall-τ rank correlation with the systems' ranking induced by M. For every $m > 1$ we use the best subset of size m-1 and select the m^{th} query from the queries indexed in Φ^c (the complement set of Φ) that maximizes the Kendall-τ between the two ranks induced by M and corresponding M_Φ. This greedy algorithm is fast and tractable but it is not guaranteed to find the best subset since the best subset of size m does not necessarily contain all the queries selected for the best subset of size m-1 [8]. In addition, applying a greedy algorithm may result in a subset that highly depends on the participating systems and does not accurately rank new systems [10].

Convex optimization can be used to find a globally optimum solution to an approximation of this problem, as outlined below.

Convex Optimization: For the i^{th} system, the average performance of this system, μ_i, based on queries in Q_N is

$$\mu_i = N^{-1} \sum_{j=1}^{N} x_{ij} = N^{-1} x_i e$$

where $e \in \{1\}^{N \times 1}$, is a column vector of N ones and $x_i \in R^{1 \times N}$ is the i^{th} row of matrix X. In addition, consider an activation vector $d \in \{0,1\}^{N \times 1}$ such that $d_j = 1$ if $j \in \Phi$ and $d_j = 0$ otherwise. The average performance of the i^{th} system on the subset Φ of size m is then

$$\mu_{i\Phi} = m^{-1} \sum_{j \in \Phi} x_{ij} = m^{-1} x_i d$$

where m is the number of selected queries. Also μ_i and $\mu_{i\Phi}$ are the i^{th} elements of vectors M and M_Φ respectively. While the greedy algorithm optimizes for Kendall-τ, which we also use as the evaluation measure in our experiments, we cannot use this measure in convex optimization. Instead, we use the *residual sum of squares* to minimize the sum of differences between pairs of μ_i and $\mu_{i\Phi}$.

$$\min_d \sum_{i=1}^{L} \left(N^{-1} x_i e - m^{-1} x_i d \right)^2; \quad \text{subject to: } \| d \|_0 \leq m \tag{1}$$

where $\| \, . \, \|_0$ is the L_0 norm that simply counts the number of non-zero elements in d and controls the size of the subset, m.

To minimize Equation 1, we use *convex relaxation* that replaces the above minimization function with a convex function that admits tractable solutions. Note that the optimization function in Equation 1 is not convex due to the L_0 norm constraint. We alter this constraint and convert Equation 1 to a convex form by removing the restriction for having only binary values, i.e., we replace d by $\beta \in [0,1]^{N \times 1}$ which contains real values bounded between 0 and 1 such that if j^{th} query is selected, $\beta_j > 0$, otherwise $\beta_j = 0$. In addition, we control the number of selected queries, m, based on the budget available to build additional relevance judgements. We denote Ω as the budget needed to build relevance judgements for all previously unjudged documents that are returned by new systems. Also B denotes the limited budget ($0 \leq B \leq \Omega$) that is available to build additional relevance judgements. We replace the L_0 constraint with the linear constraint $\sum_{j=1}^{N} \beta_j \leq \frac{B}{\Omega}$. Therefore, choosing a subset is now based on solving the following convex optimization function:

$$\min_{\beta} \sum_{i=1}^{L} \left(N^{-1} x_i e - x_i \beta \right)^2; \quad \text{subject to:} \sum_{j=1}^{N} \beta_j \leq \frac{B}{\Omega} \tag{2}$$

To solve this convex optimization we use Least Angle Regression (LARS) [7] in our experiments. LARS generates the optimal subsets of size $1, 2, ..., N$ as the budget B varies from 0 to Ω. Hence, we select all queries for which β_j is non-zero, and by varying the budget B, we can control the number, m, of queries in the subset.

4 Experimental Evaluation

Our experimental investigations were performed using the Robust track of TREC 2004 consisting of 249 topics (queries), and 14 sites with a total of 110 runs. Normally, organizations participating in a TREC experiment register as *sites* and submit a number of experimental *runs* for evaluation. These runs often represent variations on a retrieval model's settings. For our purposes we considered runs as search systems, taking special care when considering runs from the same site.

In order to evaluate properties of our query selection method, we partitioned the set of all experimental runs into *participating* and *new* systems. In addition, in order to ensure that new systems were truly different from the participating ones, we held out as new systems not only individual runs but also the entire set of runs from the same site. Furthermore, during computation of performance metrics, we removed the documents that were uniquely retrieved by the new (held-out) systems from the pool.

We describe the results of two experiments. In the first experiment, we select subsets of varying size m, using each of the three query selection methods. We then compare how well these subsets rank new systems based on the relevance judgements provided by participating systems. This experiment is intended to

determine which of the three query selection methods selects the best subsets for ranking new systems, i.e. which algorithm provides subsets that generalize to new systems. The second experiment compares the performance of two budget allocation methods, uniform allocation across all queries versus uniform allocation across a subset of queries, that are used to expand relevance judgements based on documents retrieved by new systems.

4.1 Generalization

We assess generalization of query subsets selected by the three algorithms, namely *random*, *greedy*, and *convex*. For the random query sampling method we report the average of 1000 random trials.

Experimental Setup: We first partitioned systems into participating and new systems. Using the participating systems we constructed initial pools for the full set of queries. We then constructed the performance matrix X, comprising effectiveness scores (AP) of the participating system-query pairs. We selected a subset of queries of size m using one of the three query selection algorithms. Next, for all queries in the subset we used corresponding relevance judgements collected from the initial pool to measure the performance of new systems and construct the corresponding vector M_Φ. Note that in the convex method the vector M_Φ weights each query equally, i.e. the value of β coefficients, calculated as part of the solution to Equation 2 are ignored. We do this because the sample mean of AP scores of selected queries is an unbiased estimator of MAP calculated over the full set of queries, and amongst all the unbiased estimators, it has the smallest variance. We also computed the vector M of new systems based on all the queries and their original set of relevance judgements collected in the Robust track. The generalization was expressed through the Kendall-τ coefficient for the new systems' rankings induced by M_Φ and M. If the two rankings were similar, the reduced set of queries was deemed useful for evaluating new systems.

We used a repeated random sub-sampling technique to split systems into participating and new systems. At each trial, we randomly selected 40% of sites, and labeled their runs as new systems. The remaining runs were treated as participating systems and used to build documents pools. For each of the query selection methods, we selected a subset of queries for a given size m. We then measured their generalization as explained above.

We repeated the sampling process over 10 trials and averaged their results to produce single estimates. We note that 10 trials of sampling ensured that the runs of each site were at least assigned once to the participating set and once to the new set. The advantage of this technique over k-fold cross validation is that the proportion of the training/test split is not dependent on the number of iterations (folds). Consequently, a considerable subset of runs, about 45 out of 110 on average, were held out at each trial as new systems.

Experimental Results: The generalization of the selected subsets by each of the query selection methods is shown in Figure 1 for subset sizes between 1 and 70. As seen, the convex optimization outperforms the greedy method

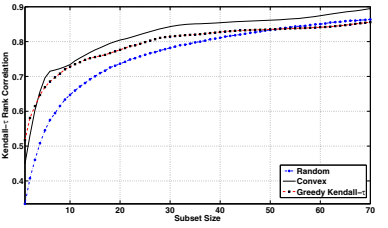


Fig. 1. The Kendall-τ of the three query selection methods as a function of query subset size. The total number of queries is 249.

and random sampling for almost all subset sizes. The difference between the Kendall-τ scores for convex and greedy is 0.04 on average, which is equivalent to correctly ordering 19 additional pairs of systems. The performance of the greedy method degrades toward random sampling for subset sizes bigger than 30. This suggests that as the size of the subset increases, the greedy method over-learns and consequently lacks generalizability. We also note that the differences between the three methods become negligible as the size of subset approaches the full set of queries. This happened in our experiment after selecting 174 queries from a total of 249.

4.2 Comparing Two Relevance Judgment Allocation Methods

In this section, we assume a fixed budget is available to collect new relevance judgments. We examine two methods for allocating the budget across queries. In the first method, the resources are equally spread across all queries. For example, if the budget can cover only 200 new judgments and there are 100 queries, we judge two new documents per query. In the second method, we select a subset of queries and then allocate the budget equally across them.

Experimental Setup: We first randomly selected a subset of sites and used their experimental runs as participating (held-in) systems. We then analyzed the held-out sites and distinguished between those sites that performed similarly to the held-in sites, i.e. there was considerable overlap in the documents retrieved by these sites and the held-in sites, and those sites that were very different from the held-in sites. To do this we applied the reusability measure proposed in [4] to measure the extent to which the corresponding pooled documents covered the documents retrieved by the held-out systems. For each held-out system and each query we considered the ranked list of documents and computed the average reuse (AR),

$$AR(q) = \frac{1}{judged(q)} \sum_{i} \frac{judged@i(q)}{i}$$

where $judged@i(q)$ was the number of judged documents in the top-i results of the held-out system for query q, and $judged(q)$ was the total number of documents judged for query q. In addition, we defined the mean average reuse (MAR) as the average of AR values for a system over the full set of queries.

We separated held-out sites into two groups based on the average of the MAR scores of their runs: those with high MAR across runs that could be evaluated using the existing relevance judgments, and the second group with runs that had low MAR and thus required additional relevance judgments in order to be evaluated. The first group of runs formed the *auxiliary set* of systems, and the others were considered as *new systems*. We used the new systems to evaluate the different resource allocation methods. The full experiment is as below:

1. Pick s_1 sites at random. The runs of these sites are treated as participating systems.
2. For each query, construct the *initial* pool of top-k_0 documents retrieved by participating systems and build associated relevance judgments. Compute the performance matrix X.
3. Compute the MAR for the runs that did not participate in the pooling. Average the MAR scores across the runs from the same site and produce average reuse score for each site.
4. Pick s_2 sites with the smallest scores and treat their runs as *new systems*. The remaining runs are *auxiliary systems* that can be evaluated with the existing relevance judgments. Their performance values are added to the performance matrix X. Note, however, that the auxiliary systems do not contribute to the document pool.
5. Given a budget B, select a subset of m queries using the convex optimization method.
6. Acquire additional relevance judgments in one of two ways:
   (a) **Subset:** For each of the m selected queries assess an additional k_1 documents contributed by the new systems where k_1 is adjusted based on B.
   (b) **Uniform:** For each of the N queries assess an additional k_2 documents contributed by the new systems where $m \times k_1 = N \times k_2$.
7. Add the newly judged documents to the initial pool and compute the effectiveness scores for the new systems.

Experimental Results: We applied the above steps across 10 trials. In each trial, we randomly chose a different set of participating sites, s_1=1, 3 or 5. The runs of the remaining sites were partitioned into auxiliary and new systems based on their reusability scores. We considered the s_2 lowest scoring sites and chose their runs to be new systems, where $s_2 = 3$, 6 or 8. The auxiliary sets comprised 5, 6 or 7 sites. To construct the initial pools we considered the top-k_0 documents from each participating system, where $k_0 = 10$ or $k_0 = 30$. Assuming a fixed budget, $B = \{1, 3 \text{ or } 5\} \times 10^4$ and a performance matrix X composed of participating and auxiliary systems, we used the convex optimization method to select a subset of queries. As B increased, the number of selected queries also increased. In our experiments, the size of the subsets varied between 14 to 237 with a median of 69.

Table 1 compares the performance statistics for the Robust 2004 track test collection before and after acquiring new relevance judgments in 12 different experimental configurations. The values given in the table are Kendall-τ scores

– averaged over 10 trials – between the ranking of new systems induced by the initial pool (containing top-k_0 documents returned by participating systems) or one of the two resource allocation methods ("uniform" and "subset") and the ranking induced by MAP scores that are measured over the full set of queries and by using the original pools (TREC *qrels*). Also, p^+ counts additional pairs of systems that are correctly ordered by the subset method when compared to the number of pairs correctly ordered by the uniform method. In addition, Ω is the number of judgements needed to build relevance judgements for all previously unjudged documents that are returned by new systems in a pool of depth 100.

We note that if the difference in average performance scores of two systems is not statistically significant, it is completely reasonable that they may be ordered differently when evaluated over a subset of queries. Having such tied systems in a test set increases the probability of a swap and consequently decreases Kendall-τ. This is because the Kendall-τ is not able to distinguish between pairs of systems with and without significant differences. This is the case in Robust track test collection in which about 30% of pairs are ties, when measured by a paired t-test at significance level 0.05. In Table 1, the Kendall-τ scores in parentheses are calculated by only considering the pairs of systems with a statistically significant difference in MAP.

The positive effect of increasing the number of sites s_1 that contribute to the document pool, can be observed from the experiments 1, 7 and 10 for which s_1 is varying from 1 to 5, with $B = 1 \times 10^4$. In addition, increasing s_1 or k_0 increases the average reuse scores of held-out sites and, consequently, reduces the number of new systems and the amount of Ω. This can be seen from the experiments 1-9. Diversifying the set of participating systems by increasing s_1 while keeping k_0 constant causes a bigger improvement in Kendall-τ than the opposite, i.e., increasing k_0 and keeping s_1 constant. This is demonstrated by

Table 1. Results for TREC 2004 Robust runs evaluated by MAP. The first six columns report experimental parameters. The next three columns report the Kendall-τ between the rankings induced by M and M_Φ of new systems for the initial pool and each of the two budget allocation methods. The last column (p^+) counts additional pairs of systems that are correctly ordered by the subset method against the uniform method. The values in parentheses are measured by only considering pairs of new systems with a statistically significant difference.

exp.#	s_1	s_2	k_0	Ω	B	$\frac{B}{\Omega}$	Kendall-τ						p^+	
							initial pool		uniform		subset			
1					10,000	0.06			0.54	(0.63)	0.6	(0.66)	95	(50)
2	1	8	10	163,842	30,000	0.18	0.42	(0.49)	0.57	(0.66)	0.64	(0.70)	110	(63)
3					50,000	0.31			0.61	(0.69)	0.68	(0.74)	111	(79)
4					10,000	0.09			0.66	(0.74)	0.73	(0.79)	53	(44)
5	1	6	30	107,817	30,000	0.28	0.54	(0.61)	0.70	(0.77)	0.77	(0.81)	62	(42)
6					50,000	0.46			0.75	(0.80)	0.82	(0.85)	62	(46)
7					10,000	0.1			0.70	(0.76)	0.76	(0.83)	53	(51)
8	3	6	10	104,580	30,000	0.29	0.60	(0.65)	0.75	(0.81)	0.82	(0.87)	62	(53)
9					50,000	0.48			0.82	(0.80)	0.89	(0.89)	71	(79)
10					10,000	0.19			0.87	(0.91)	0.90	(0.95)	7	(11)
11	5	3	10	53,535	30,000	0.56	0.70	(0.77)	0.92	(0.94)	0.96	(0.98)	11	(8)
12					50,000	0.93			0.99	(1.0)	0.98	(1.0)	-2	(0)

the experiments 4 and 7 where $s_2 = 6$ and $B = 1 \times 10^4$. This result is consistent with observations by Carterette et al. [4] that a higher diversity of participating systems results in a better ranking of new systems. In experiments 1-9 where the total cost, Ω, is considerably bigger than the available budget, B, the subset method significantly outperforms the uniform method. As B approaches Ω, the amount of improvement decreases such that in the last experiment ($s_1 = 5$ and $\frac{B}{\Omega} = 0.93$) the Kendall-τ obtained by the uniform method is bigger than the Kendall-τ for subset. We note that, as B approaches Ω, the number of selected queries gets closer to the total number of queries in the test collection. Therefore, when $\Omega \approxeq B$, the difference between the performance of subset and uniform method is negligible.

5 Discussion and Conclusion

In this paper, we considered the problem of expanding the relevance judgements of a test collections in order to better evaluate the performance of new systems. Given a fixed budget, we investigated whether it is better to uniformly allocate the budget across all the queries in the test collection, or only to a subset of queries. Our hypothesis was that a smaller but representative set of queries with a greater number of judged documents per query increases the accuracy of ranking new systems.

The hypothesis was tested using the Robust Track of TREC 2004. Three methods for determining a subset of queries were considered, referred to as random, greedy and convex. Experimental results demonstrated that query selection based on a convex optimization provided better generalization. The difference between the Kendall-τ scores for convex and greedy was 0.04 on average, which is equivalent to correctly ordering 19 additional pairs of systems. For a fixed budget, we then compared how well new systems were ranked, based on a uniform allocation across (i) all queries and (ii) a subset of queries chosen using the convex method. A variety of different experimental configurations were tested, which (i) varied the number of participating sites (1, 3 or 5), (ii) the number of new sites (3, 6 or 8), (iii) the size of the top-k_0 documents contributing to the initial pool, and (iv) the budget available ($B = \{1, 3 \text{ or } 5\} \times 10^4$ additional relevance judgments). When B was much smaller than the required budget, Ω, to build complete relevance judgements, allocating the budget uniformly across a subset of queries performed better than uniform allocation across all queries. As B approached Ω the difference between two methods became negligible.

There are a variety of avenues for future work. First, while the convex optimization was shown to exhibit better generalizability, its objective function does not explicitly consider generalization. However, in practice we could, perhaps, indirectly measure generalizability based on predicting the number of unseen relevant document for each query, based on work in [14], and a corresponding cost term could be incorporated into our objective function.

Ultimately, a budget allocation strategy should be prioritizing not just queries, but individual query-document pairs. Considerable work has been done to minimize the number of documents pooled for a given query [3,2]. These techniques

are largely complementary to the experiments in this paper, and future work is needed to combine these different approaches.

References

1. Allan, J., Carterette, B., Aslam, J.A., Pavlu, V., Dachev, B., Kanoulas, E.: TREC 2007 million query track. Notebook Proceedings of TREC 2007. TREC (2007)
2. Aslam, J.A., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 541–548. ACM, New York (2006)
3. Carterette, B., Allan, J., Sitaraman, R.: Minimal test collections for retrieval evaluation. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 268–275. ACM, New York (2006)
4. Carterette, B., Gabrilovich, E., Josifovski, V., Metzler, D.: Measuring the reusability of test collections. In: WSDM 2010: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 231–240. ACM, New York (2010)
5. Carterette, B., Kanoulas, E., Pavlu, V., Fang, H.: Reusable test collections through experimental design. In: SIGIR 2010: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 547–554. ACM, New York (2010)
6. Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J.A., Allan, J.: Evaluation over thousands of queries. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 651–658. ACM, New York (2008)
7. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2004)
8. Guiver, J., Mizzaro, S., Robertson, S.: A few good topics: Experiments in topic set reduction for retrieval evaluation. ACM Trans. Inf. Syst. 27(4) (2009)
9. Mizzaro, S., Robertson, S.: Hits hits trec: exploring ir evaluation results with network analysis. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2007, pp. 479–486. ACM, New York (2007)
10. Robertson, S.: On the contributions of topics to system evaluation. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 129–140. Springer, Heidelberg (2011)
11. Sparck Jones, K., van Rijsbergen, K.: Information retrieval test collections. Journal of Documentation 32(1), 59–75 (1976)
12. Voorhees, E.M.: The philosophy of information retrieval evaluation. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 355–370. Springer, Heidelberg (2002)
13. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 316–323. ACM, New York (2002)
14. Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–314. ACM, New York (1998)

On the Feasibility of Unstructured Peer-to-Peer Information Retrieval

H. Asthana, Ruoxun Fu, and Ingemar J. Cox

Department of Computer Science, University College London, Gower St., London WC1E 6BT, UK
{h.asthana,r.fu,ingemar}@cs.ucl.ac.uk

Abstract. We consider the feasibility of web-scale search in an unstructured peer-to-peer network. Since the network is unstructured, any such search is probabilistic in nature. We therefore adopt a probably approximately correct (PAC) search framework. The accuracy of such a search is defined by the overlap between the set of documents retrieved by a PAC search and the set of documents retrieved by an exhaustive (deterministic) search of the network. For an accuracy of 90%, we theoretically determine the number of nodes each query must be sent to for three distributions of documents in the network, namely uniform, proportional and square root. We assume that the query distribution follows a power law and investigate how performance is affected by the scale factor. For various configurations, we estimate the global and local network traffic induced by the search. For a network of 1 million nodes, a query rate of 1000 queries per second, and assuming each node is capable of indexing 0.1% of the collection, our analysis indicates that the network traffic is less that 0.07% of global internet traffic.

1 Introduction

P2P networks can be generally categorized into two classes, namely structured and unstructured networks. Structured networks, typically based on distributed hash tables (DHTs), bind data to designated locations within the network. The advantage of a structured architecture is that the query latency, proportional to the number of nodes a query must visit, is $O(\log n)$ where n is the number of nodes in the network. However, multi-term queries can consume considerable bandwidth as nodes need to exchange information regarding the sets of documents containing each term [1]. Additional bandwidth is needed to maintain the binding, and can grow very quickly in the face of dynamic membership (churn), which can in turn saturate the network. Further concerns have been raised in [2]. In particular, distributed hash tables are particularly susceptible to adversarial attack [3].

Unstructured networks exhibit no such binding between data and nodes. As such, they are much less affected by churn, and are generally more resistant to adversarial attack. However, since a particular document being sought by a user can be anywhere in the network, the only way to guarantee searching

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 125–138, 2011.

the entire collection is to exhaustively query all nodes in the network. This is, of course, impractical. To be practical, any search must only query a relatively small subset of nodes in the network. Thus, search in an unstructured P2P network is necessarily probabilistic.

One of the earliest attempts to estimate the feasibility of web search in peer-to-peer networks was undertaken by Li *et al.* [1] which examined the practicality of web search based on a structured P2P network indexing 3 billion documents and concluded that the bandwidth required was still an order of magnitude greater than was practical at the time. Even after applying various optimization techniques, the estimated query size (i.e. the total traffic generated in issuing and answering a query) was still found to be 6 MB per query. Whilst this (relatively high) communication cost would now be feasible, as we discuss later, the analysis in [1] does not account for expected churn rates in observed peer-to-peer systems which can significantly increase the required bandwidth in structured networks for keeping the DHT up-to-date. Zhong *et al.* [4] investigated the effectiveness of various indexing strategies in structured peer-to-peer networks using 3.7 million queries. However, the authors did not investigate unstructured peer-to-peer networks and did not address the concerns, previously mentioned, regarding structured peer-to-peer networks. Yang *et al.* [5] compared the performance of keyword search in structured, super-peer, and unstructured peer-to-peer network by downloading the documents from 1,000 web sites and allocating each peer to "host" one web site. The conclusion of the study is that the performance of the three network types is similar. However, since the documents are only present at one node, there is no *replication of documents* - a key concept which we explore in the next section.

To our knowledge, there have been no studies into whether it is possible to perform web search in unstructured peer-to-peer networks which takes into account document replication with high probabilities of finding the relevant document(s).

The main contributions of this paper are

- a theoretical analysis that predicts the number of nodes that must be queried in order to guarantee an expected accuracy of 90% for three different document replication policies
- estimation of the corresponding communication bandwidth required, concluding that probabilistic search in an unstructured peer-to-peer network where nodes issue queries in volume comparable to commercial search engines will consume no more than 0.07% of the global internet traffic.

In this paper, the feasibility of web scale search in unstructured peer-to-peer networks is based on communication cost. However, we acknowledge that other factors are also of concern, e.g. latency and security. Latency is directly proportional to the number of nodes queried. Although it is part of our future research, we do not address latency in this paper, and refer the reader to [6,7], which discusses the optimal network topology to reduce latency. The security of an unstructured peer-to-peer network is outside the scope of the paper. However, initial investigations, not reported here, suggest that security is better than that for structured architectures based on distributed hash tables.

In Section 2 we review prior work on probabilistic search in an unstructured P2P network. In Section 3 we extend the probably approximately correct search architecture, which is a recently proposed unstructured P2P search framework, to incorporate non-uniform replication strategies. In Section 4 we calculate the communication cost of web search in unstructured P2P networks based on the theoretical results of Sections 2 and 3. We conclude in Section 5.

2 Probabilistic Search

Probabilistic storage and search in an unstructured P2P network can be modeled as follows. Given a set of n nodes in the network, we assume that the object of interest is stored on a random subset of r nodes. A query is issued to a random subset of z nodes. We are interested in the probability that the two subsets have a non-empty intersection, as this implies a successful search for that object. This theoretical foundation is directly adopted in prior work which focuses on retrieval of files stored in the network based on queries that contain terms that only appear in the file names. Information retrieval is broader than this, as the index, and associated queries, contain terms present not just in the file name, but also terms present within the file (document) itself. As such, matching of queries to documents is more ambiguous, and it is therefore necessary to provide a set of documents, usually ranked by relevance. Nevertheless, this model is appropriate for the class of information retrieval problems referred to as *known-item* search. And, the probabilistic model can be extended to encompass other information retrieval requirements, as discussed in Section 2.3.

2.1 Ferreira et al's Model

Early work on probabilistic search in unstructured P2P networks has its origins in the study of probabilistic quorum systems [8] to improve the availability and efficiency of replicated systems. Ferreira *et al.* [9] proposed the use the probabilistic quorum model to describe search in an unstructured P2P network. Given n nodes in the network, an object is replicated $\gamma\sqrt{n}$ times onto a random subset of nodes. A query is also sent to a random subset of $\gamma\sqrt{n}$ nodes. It can then be shown that the probability of finding the desired object of the query is at least $1 - e^{-\gamma^2}$. Clearly as γ increases, the object is replicated over more nodes and the probability of finding the object therefore increases.

2.2 Cohen and Shenkers' Model

The previous analysis assumed that an object/document is uniformly randomly replicated across nodes in the network. Other replication strategies are also possible. Cohen and Shenker [10] provided both a theoretical and empirical analysis of such. Here it is assumed that the n nodes in the P2P network each have capacity ρ, i.e. ρ is the number of files each node can store. Let $R = n\rho$ denote the total capacity of the system. It is further assumed that there are m unique

files stored in the P2P system and that each file i is replicated on r_i random nodes. Obviously, $\sum_i r_i = R$, and $mr = R$ if $r_i \equiv r$ is a constant. Let $p_i = \frac{r_i}{R}$ be the fraction of the total system capacity allocated to file i. Finally, let q_i be the normalized query popularity for the ith file. Thus

$$\sum_{i=1}^{m} q_i = 1$$

The search size, z_i, of file i, is defined as the number of nodes searched in response to a query q_i to find file i. Of course, the search size will depend very much on the search strategy used. In [10] a random probing model is assumed, i.e. each of many probes randomly selects a node in the network. Thus, each probe has a probability $\frac{r_i}{n}$ of finding the requested file i, and a probability $1 - \frac{r_i}{n}$ of not finding the file. The search size z_i is simply a random variable drawn from a geometric distribution

$$P(z_i) = (1 - \frac{r_i}{n})^{z_i - 1} \frac{r_i}{n}$$

The average search size for file i is

$$\mu_z(i) = \frac{n}{r_i}$$

and the expected search size, μ_z, of all m files is

$$\mu_z = \sum_i q_i \times \mu_z(i) = n \sum_i \frac{q_i}{r_i} = n \sum_i \frac{q_i}{R p_i} = \frac{n}{R} \sum_i \frac{q_i}{p_i} \tag{1}$$

For a uniform replication strategy, $r_i \equiv r$ is a constant and $mr = R$. Thus, the expected search size with uniform replication, μ_z^u, is

$$\mu_z^u = n \sum_i \frac{q_i}{r} = \frac{n}{r} \sum_i q_i = \frac{m}{\rho} \tag{2}$$

Two alternatives to a uniform replication strategy are also considered. A proportional replication strategy replicates content based on its popularity, i.e. proportional to the number of queries requesting it. Perhaps surprisingly, such a replication strategy results in the same expected search size as the uniform replication. For proportional replication strategy, $r_i = R q_i$ and the expected search size μ_z^p is

$$\mu_z^p = n \sum_i \frac{q_i}{R q_i} = \frac{nm}{R} = \frac{m}{\rho}$$

In effect, while popular documents will be found by querying fewer nodes than for a uniform replication strategy, this is balanced by the need to visit far more nodes in order to find unpopular documents.

To minimize the expected search size, we would like to minimize $\sum_i \frac{q_i}{p_i}$ in Equation (1). Solving this optimization problem [10], we have $\frac{p_i}{p_m} = \frac{r_i}{r_m} = \frac{\sqrt{q_i}}{\sqrt{q_m}}$.

Thus, $r_i = \lambda\sqrt{q_i} = \frac{R}{\sum_i \sqrt{q_i}}\sqrt{q_i}$. This is the square root replication strategy which produces the optimal expected search size given by

$$\mu_z^s = n \sum_i \frac{q_i}{\lambda\sqrt{q_i}} = \frac{1}{\rho}(\sum_i \sqrt{q_i})^2 \tag{3}$$

Typically, the query distribution follows a power law [11], i.e. $q_i = \frac{1}{c} i^{-\alpha}$ where c is the normalization constant. In this case, Equation (3) becomes

$$\mu_z^s = \frac{1}{\rho c}(\sum_i \frac{1}{i^{\alpha/2}})^2 \tag{4}$$

Analysis of publicly available logs from AOL, as well as logs of a commercial search engine made available to us, indicate that the value of α ranges from 0.8 to 1.0.

2.3 Probably Approximately Correct Search

The previous work on randomized search looked at the expected search length necessary to find a specific document. Assuming a query is sent to a constant number of nodes, z, we can also ask what the probability of finding a document is. This, and related questions, are addressed in recent papers on probably approximately correct (PAC) search [12,13]. The PAC architecture considers both an acquisition and a search stage. In this paper we only consider the search stage.

During the search stage, a query is sent to z machines, and the results returned by the different machines are consolidated and then displayed to the user. If we are searching for a single, specific document d_i, then the probability of retrieving this document is given by

$$P(d_i) = 1 - (1 - \frac{\rho}{m})^z \tag{5}$$

In information retrieval, it is more common to be interested in the top-k retrieved documents. In this case, the correctness of a PAC search is measured by *retrieval accuracy*. If $\mathcal{D}$ denotes the set of top-k documents retrieved when searching the full index, i.e. an exhaustive search, and $\mathcal{D}'$ the set of top-k documents retrieved when querying z nodes, then the retrieval accuracy, a, is defined as

$$a = \frac{|\mathcal{D} \cap \mathcal{D}'|}{|\mathcal{D}|} = \frac{k'}{k}$$

where k' denotes the size of the overlap of the two sets, i.e. $|\mathcal{D} \cap \mathcal{D}'|$.

The size of the overlap in the result sets, k' is a random variable drawn from a binomial distribution, and is given by

$$P(k') = \binom{k}{k'} P(d_i)^{k'} (1 - P(d_i))^{k-k'} \tag{6}$$

Since Equation (6) is a binomial distribution, the expected value of k' is $E(k') = kP(d_i)$ and the expected retrieval accuracy μ_e is

$$\mu_e = \frac{\mu_{k'}}{k} = \frac{k \times P(d_i)}{k} = 1 - \left(1 - \frac{\rho}{m}\right)^z, \tag{7}$$

If we assume that a document is, on average, replicated r times onto different nodes in the network, then the total storage of the network, R, satisfies $R = m \times r$. And the Equation (7) can be transformed into

$$\mu_e = 1 - \left(1 - \frac{\rho}{m}\right)^z = 1 - \left(1 - \frac{r}{n}\right)^z \tag{8}$$

3 Non-uniform Replication in PAC

The original work on PAC, described above, assumed a uniform replication strategy for documents. Here, we extend the probabilistic analysis to the cases where the documents are replicated (i) in proportion to their popularity, and (ii) in proportion to the square root of their popularity, as discussed in [10].

In general, given a query distribution, we are interested in what replication strategy can yield the highest expected retrieval accuracy for all queries. Let $\mathcal{Q}$ denote the set of all queries, and let q_j denote the query rate for query j, such that $\sum q_j = 1$. The replication rate for document i is r_i. From Equations (5) and (8), we can get the probability of retrieving document i as

$$P(d_i) = 1 - \left(1 - \frac{r_i}{n}\right)^z \tag{9}$$

Let $\mathcal{D}_k(j)$ denote the set of top-k documents retrieved for a query q_j. Thus, the expected retrieval accuracy for the top-k documents, A_k, averaged over all queries is given by

$$A_k = \sum q_j \frac{\sum_{d_i \in \mathcal{D}_k(j)} (1 - (1 - \frac{r_i}{n})^z)}{k} \tag{10}$$

Now consider the case where we are only interested in the top document, i.e. top-1 ($k = 1$), as in [10]. Thus, from Equation (10) we have

$$A_1 = \sum q_j (1 - (1 - \frac{r_j}{n})^z) \tag{11}$$

For a proportional replication strategy, where $r_j = Rq_j$, the expected accuracy A_1^p is then given by

$$A_1^p = \sum q_j (1 - (1 - \rho q_j)^z) \tag{12}$$

and for square root replication strategy, where $r_j = R\frac{\sqrt{q_j}}{\sum \sqrt{q_j}}$, the expected accuracy A_1^s is given by

$$A_1^s = \sum q_j (1 - (1 - \rho \frac{\sqrt{q_j}}{\sum \sqrt{q_j}})^z) \tag{13}$$

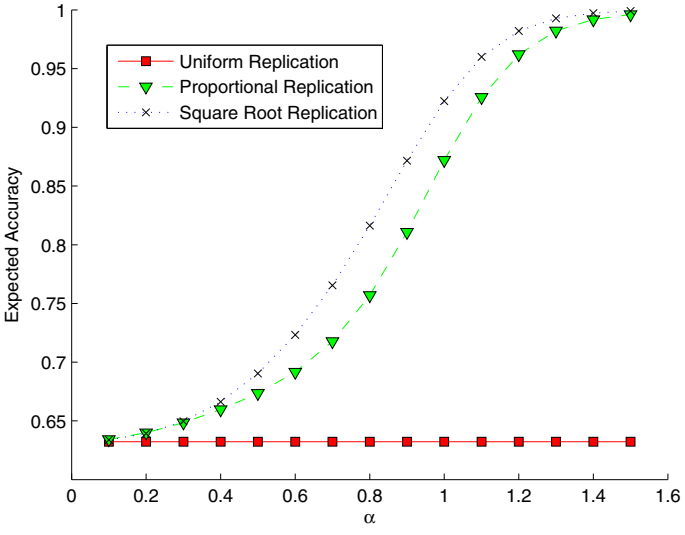


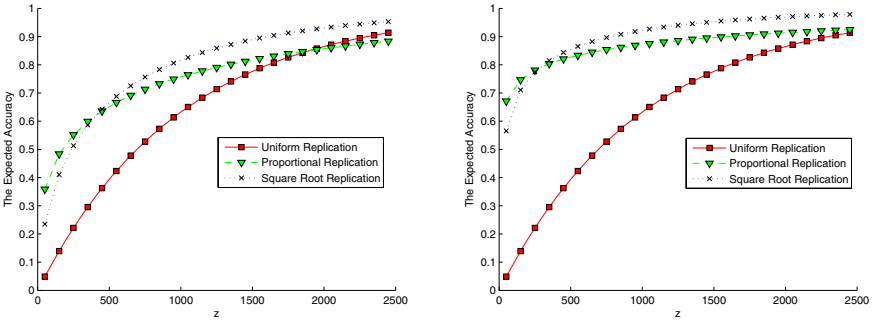
Fig. 1. Expected accuracy for retrieving the top-1 document, A_1, as a function of the power law exponent, α, for different replication strategies, when the number of nodes queried, $z = 1000$

To examine the effect of the different replication strategies, we considered a PAC configuration in which it is assumed that there are 1 million documents in the collection ($m = 10^6$), and 10,000 nodes in the network ($n = 10^4$). Each node is able to index 1000 documents ($\rho = 1000$).

Figure 1 shows the expected accuracy, A_1, when retrieving the top-1 document as a function of the power law exponent for different replication strategies. Here, we have assumed that the query distribution follows a power law, and have fixed the search size to 1000 nodes, ($z = 1000$). The square root replication strategy performs better than the proportional replication strategy, and grows more rapidly as α increases.

Figures 2a and 2b show the expected accuracy, A_1, when retrieving the top-1 document, as a function of the search size, z, for different replication strategies, and for $\alpha = 0.8$ and $\alpha = 1$ respectively. We observe that square root replication is inferior to proportional replication, when the search size is small. Note however, that as the search size increases, proportional replication improves more slowly, and square root replication performs better.

We can extend our analysis to the case where we are interested in the top-k retrieved documents, rather than only the top-1. Theoretically, an infinite number of queries can be issued by users, and many queries can retrieve the same documents. However, to simply our analysis of expected accuracy we assume a finite number of queries, $|Q|$, which is certainly true for a finite period of time. The top-k documents retrieved by each query, $\mathcal{D}_k(j)$, are likely to be non-disjoint, i.e. two queries might retrieve some documents in common. Thus, replication



(a) $\alpha = 0.8$ (b) $\alpha = 1.0$

Fig. 2. Expected accuracy for retrieving the top-1 document, A_1, as a function of the search size, z, for different replication strategies

should be based on the distribution of retrieval frequency of the documents, rather than the query distribution directly.

To solve top-k retrieval problem, let us define a document retrieval frequency set $\mathcal{Q}'$ which holds the distribution of retrieval frequency of the documents in the collection. Thus, for each $q'_i \in \mathcal{Q}'$, we have

$$q'_i = \sum_{j=1}^{|\mathcal{Q}|} q_j \zeta(j,i)$$

where

$$\zeta(j,i) = \begin{cases} 1 \text{ if document } i \text{ is in query } j\text{'s top-}k \text{ result list.} \\ 0 \text{ otherwise.} \end{cases}$$

We can then transform Equation (10) to

$$A_k = \sum q'_i (1 - (1 - \frac{r_i}{n})^z) \tag{14}$$

where the replication rate for document i, r_i, is computed based on the corresponding q'_i. Since the expected accuracy is essentially a weighted mean, we can exploit the overlap of retrieved documents of queries and are able to simplify the top-k retrieval into an equation which is akin to the top-1 retrieval.

4 Communication Cost

In this Section we consider the communication cost associated with P2P search. We assume a probabilistic search architecture based on the PAC model. We make the following assumptions with regard to the system:

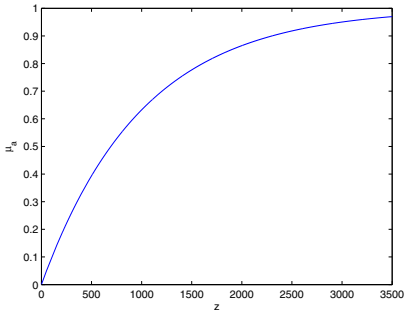
- a network size of $n = 1,000,000$ nodes.
  Several P2P services already exceed this number, e.g. Gnutella and BitTorrent, and the commercial P2P information retrieval system Faroo[1] currently claims 1 million users.
- a query rate of 1,000 queries a second.
  The estimated query rate of Google is 38,000 queries per second[2]. However, Google's query rate is based on a user community of about 150M unique users[3]. A query rate of 38,000 queries per second is equivalent to each of 1 million nodes issuing over 2 queries a minute! A rate of 1000 queries per second corresponds to each node issuing almost 4 queries an hour, 24 hours per day, which would seem like an upper bound on any realistic query rate.
- a collection size of 10 billion documents to be indexed.
  Currently, it is estimated that Google indexes approximately 20 billion documents, while Bing and Yahoo index approximately 12 billion documents[4].
- a required expected retrieval accuracy of 90%.
  If we are only interested in a single document, then the accuracy is given by Equation (9). Thus, we must choose a combination of the number of nodes the query is sent to, z, and the local storage capacity, ρ. Let $\kappa = \frac{\rho}{m}$, denote the fraction of the global collection indexed at a node.
- a minimum storage of 5 GB, available at each node and a maximum of 10 GB.
  This allows a node to index $\kappa = \frac{1}{1000}$ of the global document collection as discussed shortly.

For uniform replication, Figure 3a illustrates the expected accuracy as a function of the number of nodes queried when each node randomly samples $\kappa = \frac{1}{1000}$ of the global document collection. For 90% accuracy we need to query approximately 2,300 nodes. Figure 3b shows the number of nodes that need to be queried to obtain 90% accuracy as function of κ.

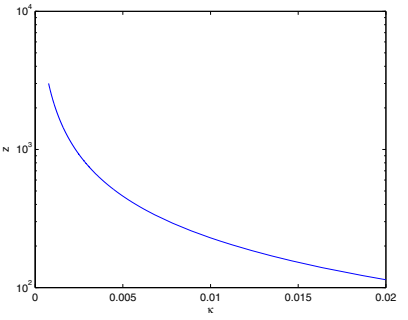
In order to estimate the communications load we further assume:

- an average of 2 bytes per character.
  This is based on UTF-8 encoding, where each character takes between 1 - 4 bytes depending on the language used[5].
- a query message size of 300 bytes.
  Analysis of query logs [11] has shown that the average query size is 2.5 terms or 30 characters. This corresponds to about 60 bytes per query message. However, we must also assume some overhead associated with the underlying TCP/IPv6 protocol. We therefore conservatively assume a query message size of 300 bytes. Therefore if this message must be sent to $z = 1,000$ peers, the communication cost associated with *sending* a query is 300 KB.

[1] http://www.faroo.com/hp/p2p/p2p.html
[2] http://searchengineland.com/by-the-numbers-twitter-vs-facebook-vs-googlebuzz-36709
[3] http://siteanalytics.compete.com/google.com+facebook.com+yahoo.com/
[4] http://www.worldwidewebsize.com
[5] http://tools.ietf.org/html/rfc3629



(a) Expected accuracy as a function of the number of nodes queried when $\kappa = 0.001$



(b) The number of nodes queried for 90% accuracy as a function of κ. Note that this is a log-linear plot.

Fig. 3. Relationship between number of nodes queried (z), the fraction of the global collection indexed at a node (κ), and the expected accuracy (μ_e) for uniform replication

Finally, in order to estimate the communication bandwidth needed to respond to a query, we assume the following:

- we are only interested in the top-10 documents.
  Analysis of commercial search engine query logs show that users rarely look beyond the top-10 documents. Thus, when a user issues a query, each node only needs to return its top-10 URLs. If, however, the user requests to see results 11-20, we could ask the same nodes to return their top 11-20, which would again be merged and re-ranked at the node originating the query.
- a query response size of 1KB.
  We estimate that each result (result name, hyper-link, snippet, minimal surrounding XML etc) requires no more than 400 characters or 800 bytes. Since the query result is entirely alphanumeric, it can usually be compressed to 10% of its original size. This is common practice with modern web servers[6].
  Thus the total bandwidth required to answer a query is simply 800 bytes per result, multiplied by 10 results per query, times 0.1 compression factor, i.e. 800 bytes. We round this to 1 KB to account for TCP/IPv6 overheads.

4.1 Communications Load for Uniform Replication

Based on the previous assumptions, we first consider the uniform replication strategy. From Equation (8), the expected accuracy of 90% can be obtained by sending the query to 2,300 nodes, assuming each node indexes 0.1% of the global collection.

We are now in a position to calculate the total communication load of such a system. For broadcasting the query and receiving the response from 2,300 peers,

[6] http://httpd.apache.org/docs/2.0/mod/mod deflate.html

the total cost per query is approximately 3 MB. For 1,000 queries per second, the total traffic generated is 3 GB/s.

Note that this traffic is spread throughout the internet. The total internet traffic in 2009 was approximately 4,630 GB/s [14,15] and is forecast to grow by 50% each year, primarily due to video traffic. Using the 2009 figures, the traffic generated by a PAC web IR service would only constitute 0.065% of the global internet traffic. Thus, web search using an unstructured P2P network will not impose a significant global communication cost.

As well as the global communication cost, it is useful to consider the requirements placed on each node, both in terms of storage and bandwidth.

We now estimate the resource requirements on each peer participating in the search. We have assumed that each node randomly samples 1/1000 of the 10 billion documents which need to indexed. This implies that each peer must index 10 million documents, which must first be crawled. The Akamai internet report[7] states that the global average internet connection speed is approximately 200 KB/s. In the developed nations it is considerably higher, but we do not account for this here. If we assume that 25% of this bandwidth can be utilized (say, during the peer's idle time), it will take approximately 58 days to complete the crawl, assuming that the average size of a document on the Web is 25KB[8].

The crawled documents, representing 250GB of data, can be indexed using approximately 10 GB of disk space which would record term frequencies and positions as well as other statistical measures. This is typical of popular information retrieval packages such as Lucene[9]. We are aware that some machines may not have 10GB of disk storage available for this service. However, lossless compression [16] can reduce the size of the index by utilizing efficient data structures, and lossy index compression techniques [17] have been shown to reduce the size of the index by 50 to 70% with minimal loss in precision.

Using efficient Trie structures, only small percentages of the index need to be read and loaded into memory, and the system can answer queries using no more than 500 MB of the peer's memory, as has been demonstrated by systems such as Lucene.

For a PAC web IR system of 1 million nodes answering 1,000 queries per second, each peer on average would have to answer 2.3 queries per second. The corresponding bandwidth needed is 0.69 KB/s in the download direction and 2.3 KB/s in the upload.

To summarize, each peer would need to contribute 5-10 GB of disk space, 500 MB of memory, and approximately 0.69 KB/s download as well as 2.3 KB/s upload from the peer's bandwidth for query answering as well as 50KB/s during idle time for crawling. Both the local communication, and disk and memory requirements appear reasonable.

[7] "Akamai report: The state of the internet, 3rd quarter, 2010", http://www.akamai.com/stateoftheinternet/

[8] http://www.optimizationweek.com/reviews/average-web-page/

[9] http://lucene.apache.org/

Table 1. The number of nodes queried and the corresponding communication cost, for uniform, proportional and square root replication strategies, for $\alpha = 0.8$ and 1 to obtain an expected accuracy of 90%.

	$\alpha = 0.8$		$\alpha = 1.0$	
	Nodes Queried	Cost/Query (MB)	Nodes Queried	Cost/Query (MB)
Uniform Replication	2300	3.000	2300	3.000
Proportional Replication	2750	3.575	1180	1.534
Square Root Replication	1650	1.534	780	1.014

4.2 Communication Load for Non-uniform Replications

As mentioned previously, the value of α ranges from 0.8 to 1.0. We can use Equations (12) and (13) to calculate the communication costs for an expected accuracy of 90% for non-uniform replications. The results are summarized in Table 1. We observe that for both values of α, the square root replication strategy needs to query fewer nodes than for the uniform distribution. For $\alpha = 0.8$ this reduces the communication bandwidth by about 50%, while for $\alpha = 1$, the bandwidth is reduced by about two thirds. It is interesting to note that for $\alpha = 0.8$, proportional replication performs worse than a uniform replication.

For a system servicing 1,000 queries per second, the communication costs correspond to between 0.03% and 0.07% of the global internet traffic.

5 Conclusion

This paper investigated the feasibility, with respect to communication bandwidth, of performing web-scale search on an unstructured, distributed peer-to-peer network. The unstructured nature of the network necessitates that the search is probabilistic in nature. While this has been previously recognized, prior work has not considered the accuracy of the search, nor the probability of attaining said accuracy.

Communication cost is often cited as the limiting factor in the deployment and scalability of P2P search. For a uniform replication policy, that ignores the query distribution and the popularity of documents, it was shown that the communication load produced by the P2P system was only 0.07% of global internet traffic, in order to guarantee an expected accuracy of 90%. In addition, the local communication load placed on each peer is approximately 2.3KB/s in the upload direction and 0.69KB/s for download. Thus, the communication overhead is well below any level that would preclude P2P search.

The communication cost can be reduced by replicating documents based on their popularity. Two popular replication policies are proportional and square root, and we extended the theoretical analysis of expected accuracy for PAC search to these two non-uniform replication policies. For $\alpha = 0.8$, the proportional policy is actually worse than uniform, but is better than uniform when

$\alpha = 1$. The square root policy is superior to uniform for both values of α. However, we note that the square root policy is not optimum for maximizing accuracy. An optimum replication policy is left for future work.

In Section 4.1 we found that a node would take approximately 58 days to complete each iteration of a web crawl. This could degrade the freshness of results and decrease the relevance of documents. A solution to this issue, could be the use of cloud computing resources such as Amazon EC2. These servers, financed perhaps with small individual donations made to a non-profit organization, can be scaled up or down based on available funding, and can be used to continuously crawl and index the web. The nodes in the P2P network could then refresh their index in fragments continuously using BitTorrent from these cloud computing based index servers. Apart from drastically reducing the number of days required for a crawl, this would have an added advantage of reducing the peer's workload of crawling and indexing.

Of course, communication cost is not the only factor that might prevent wide scale P2P web search. Latency, i.e. the time to respond to a query, is also a factor. Latency is usually considered to be proportional to the number of nodes queried. However, we note that for an unstructured P2P search architecture, we do not have to wait for all peers to respond before displaying a *partial* result list. The heterogeneity of peers in any P2P network makes this an inevitability. The analysis, modelling, and minimization of latency in web search within an unstructured P2P architecture is part of our future research.

Acknowledgment. We are thankful for the comments and suggestions provided by the referees, which has considerably helped in improving this paper. Ruoxun Fu gratefully acknowledges the support of BT.

References

1. Li, J., Loo, B.T., Hellerstein, J., Kaashoek, F., Karger, D.R., Morris, R.: On the Feasibility of Peer-to-Peer Web Indexing and Search. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, Springer, Heidelberg (2003)
2. Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N., Shenker, S.: Making gnutella-like p2p systems scalable. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 407–418. ACM, New York (2003)
3. Sit, E., Morris, R.: Security considerations for peer-to-peer distributed hash tables. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 261–269. Springer, Heidelberg (2002)
4. Zhong, M., Moore, J., Shen, K., Murphy, A.L.: An evaluation and comparison of current peer-to-peer full-text keyword search techniques. In: Proc. of the International Workshop on the Web Databases (WEBDB) (2005)
5. Yang, Y., Dunlap, R., Rexroad, M., Cooper, B.F.: Performance of Full Text Search in Structured and Unstructured Peer-to-Peer Systems. In: Proceedings of the 25th IEEE Internation Conference on Computer Communications, Barcelona, Spain. IEEE, Los Alamitos (2006)

6. Cooper, B.F.: An optimal overlay topology for routing peer-to-peer searches. In: Alonso, G. (ed.) Middleware 2005. LNCS, vol. 3790, pp. 82–101. Springer, Heidelberg (2005)
7. Terpstra, W.W., Kangasharju, J., Leng, C., Buchmann, A.P.: Bubblestorm: resilient, probabilistic, and exhaustive peer-to-peer search. In: SIGCOMM, pp. 49–60 (2007)
8. Malkhi, D., Reiter, M., Wright, R.: Probabilistic quorum systems. In: Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing, p. 273. ACM, New York (1997)
9. Ferreira, R., Ramanathan, M., Awan, A., Grama, A., Jagannathan, S.: Search with probabilistic guarantees in unstructured peer-to-peer networks. In: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (2005)
10. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 177–190. ACM, New York (2002)
11. Baeza-Yates, R., Gionis, A., Junqueira, F., Murdock, V., Silvestri, F.: The impact of caching on search engines. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27. ACM, New York (2007)
12. Cox, I.J., Fu, R., Hansen, L.K.: Probably approximately correct search. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 2–16. Springer, Heidelberg (2009)
13. Cox, I.J., Zhu, J., Fu, R., Hansen, L.K.: Improving query correctness using centralized probably approximately correct (pac) search. 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. 265–280. Springer, Heidelberg (2010)
14. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: Proceedings of the ACM SIGCOMM 2010 Conference on SIGCOMM, New Delhi, India, August 30-September 03. ACM, New York (2010)
15. http://dtc.umn.edu/mints/
16. Witten, I.J., Moffat, A., Bell, T.C.: Managing Gigabvtes: Compressing and Indexing Documents and Images, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (1999)
17. de Moura, E.S., dos Santos, C.F., Fernandes, D.R., Silva, A.S., Calado, P., Nascimento, M.A.: Improving Web search efficiency via a locality based static pruning method. In: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, May 10-14 (2005)

Can Information Retrieval Systems Be Improved Using Quantum Probability?

Massimo Melucci

University of Padua
massimo.melucci@unipd.it

Abstract. In this paper we reformulate the retrieval decision problem within a quantum probability framework in terms of vector subspaces rather than in terms of subsets as it is customary to state in classical probabilistic Information Retrieval. Hence we show that ranking by quantum probability of relevance in principle yields higher expected recall than ranking by classical probability at every level of expected fallout and when the parameters are estimated as accurately as possible on the basis of the available data.

1 Introduction

Data management systems, such as database, information retrieval (IR), information extraction or learning systems, store, organize, index, retrieve and rank information units, like tuples, objects, documents and items. A wide range of applications of these systems have emerged that require the management of uncertain or imprecise data. Important examples of data are sensor data, webpages, newswires, imprecise attribute values. What is common to all these applications is uncertainty and, that they have to deal with decision and statistical inference. In this paper, we concentrate on IR, yet what is illustrated can be generalized to other domains.

Ranking by probability of relevance is perhaps the most crucial task performed by IR systems. To perform this task, the region of acceptance of a hypothesis (e.g. the document is relevant) must be calculated. It is then possible to detect whether the observed data confirm the hypothesis (detection), calculate a probability of detection (also known as expected recall or power) and calculate a probability of false alarm (also known as expected fallout or size). When a threshold is tuned, the system ranks the document by probability of relevance, namely, the probability of detection.

Probabilistic IR systems are based on classical probability theory which views events as subsets – for example, document collections can be partitioned into two subsets of documents that correspond to the presence/absence of a term in such a way that a document belongs to a subset if and only if the term occurs/does not occur. Although IR systems reach good results thanks to classical probability theory, ranking is far perfect because irrelevant documents are often ranked at the top or relevant documents are missed.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 139–150, 2011.

In contrast, quantum probability theory views events as subspaces. The key difference between subsets and subspaces is that a subspace is a subset of vectors tied up with linear functions. This implies that the membership to a subspace is implemented by projectors, which tell if a vector belongs to a subspace, while membership to a subset is implemented by an (indicator) function which tells if an element belongs to the subset or not. The move from subset to subspace is crucial because it entails the use of probability measures of a different nature which have no counterpart in the classical probability theory.

The main question asked in this paper is whether further improvement in retrieval effectiveness may be obtained if the classical probability theory is replaced by an alternative probability theory, where by "further" we do not mean "incremental". The answer given in this paper is affirmative. We show that ranking documents by probabilility of detection where regions of acceptance are based on *subspaces* is in principle more effective than rankings by classical probability given the same data available for parameter estimation and at every given probability of false alarm.

2 Related Work

This paper links to [9] with regard to the foundations of density matrices and projectors. The notions of Quantum Theory and IR are used in this paper as they are in [4]. The results of this paper are inspired by [3] which provides the foundations and the main results in quantum detection; an example of the exploitation of the results in quantum detection is reported within communication theory [2].

We parallel the Probability Ranking Principle (PRP) proposed in the context of classical probability in [7]. The PRP states that if the parameters are as accurately estimated as possible, the probability of relevance yields the best ranking, that is, the best region of acceptance in terms of expected recall and expected fallout provided that the regions of acceptance are subsets. In contrast, we keep the same parameter estimation, but claim that there are more effective regions of acceptance and rejection defined in terms of subspaces. As the PRP leverages Bayes' postulate and then the distributive law, it is incompatible with subspaces which do not admit the distributive law as Bayes' postulate does [1].

In [11] the authors propose the Quantum PRP (QPRP) to rank documents by quantum probability and suggest that interference (which must be estimated) might model dependencies in relevance judgements such that documents ranked until position $n-1$ interfere with the degree of relevance of the document ranked at position n. The optimal order of documents under the PRP differs from that of the QPRP. However, higher effectiveness of quantum probability may stem only from the correct estimation of interference. Moreover, it is impossible to say whether the QPRP is superior to the PRP or to any other ranking principle (e.g. vector- or logic-based principles). Moreover, the QPRP estimates probabilities from statistical features of the document collection, thus using the Bayes postulate and then the distributive law of subsets.

Similarly, in [6] the authors discuss how to employ quantum formalisms for encompassing various IR tasks within a single framework. From an experimental point of view, what that paper demonstrates is that ranking functions based on quantum formalism are computationally feasible.

In contrast to these two papers, in this paper we do not need to address interference because quantum probability can be estimated using the same data used to estimate classical probability, nor do we use just a formalism. We rather show that not only does ranking by quantum probability provide a different optimal ranking, it is also more effective than classical probability because ranking optimality only depends on the region of acceptance defined upon subspaces and not on estimation (which could be well based on BM25).

3 Quantum Probability and Decision

Document representation and ranking are described in terms of decision[1] and are affected by two errors: missed detection and false alarm. Thus, the probability of detection and the probability of false alarm related to a decision must be calculated.

A certain document (e.g. a webpage or a store item) is observed in such a way as to obtain numbers (e.g. the PageRank or the number of positive reviews) on the basis of which a decision has to be made about its state. The *state* might be, for example, the relevance of the webpage to the search engine user's interests or the customer's willingness to buy the store item. The use of the term "state" is not coincidental, because the numbers are observed depending upon the density matrix, which is indeed the mathematical notion implementing the state of a system. Thus, quantum probability ascribes the decision about the state of a document to test the hypothesis that the density matrix has generated the observed numbers.

Consider the density matrix (state) ρ_1 and the alternative density matrix (state) ρ_0. In data management, ρ_0 asserts, for example, that a customer does not buy an item or that a webpage shall be irrelevant to the search engine user, whereas ρ_1 asserts that an item shall be bought by a customer or that a webpage shall be relevant to the user. Therefore, the probability that, say, a feature occurs in an item which shall not be bought by a customer or that a keyword occurs in a webpage which shall be irrelevant to the search engine user depends on the state (i.e. the density matrix).

Statistical decision theory is usually based on classical probability. Neyman-Pearson's lemma [5] is by now one out of the most important results because it provides a criterion for deciding upon states. The lemma provides the rule to govern the behavior of a decider (e.g. an IR system) as for the true state without hoping to know whether it is true. Given a document and a state about the document, such a rule calculates a specified number (e.g. a feature weight) and, if the number is greater than a threshold, the decider rejects the state, otherwise, it accepts it. Such a rule tells us nothing about whether, say, the

[1] Estimation is a special case of decision [3].

document shall be deemed relevant by the user, but the lemma proves that if the rule is always followed, then in the long run the state shall be accepted at the highest probability of detection (or power, or expected recall) possible at a fixed probability of false alarm (or size, or expected fallout) [5]. The set of the pairs given by size and power is the power curve, which is also known as the Receiver Operating Characteristic (ROC) curve. Neyman-Pearson's lemma implies that the set of the observable numbers (e.g. feature weights) can be partitioned into two distinct regions; one region includes all the numbers for which the state shall be accepted and is termed acceptance region; the other region includes all the numbers for which the state shall be rejected and is termed rejection region. For example, if a keyword is observed from webpages and only presence/absence is observed, the set of the observable numbers is $\{0, 1\}$ and each region is one out of possible subsets, i.e. $\emptyset, \{0\}, \{1\}, \{0, 1\}$. The lemma is at the basis of probabilistic IR.

This paper reformulates Neyman-Pearson's lemma in terms of subspaces instead of subsets to utilize quantum probability. Therefore, the region of acceptance and the region of rejection must be defined in terms of subspaces. The definition and the proof of the following result is in [3].

Theorem 1. *Let ρ_1, ρ_0 be the density matrices. The region of acceptance at the highest power at every size is given by the projectors of the spectrum of*

$$\rho_1 - \lambda\rho_0 \qquad \lambda > 0 \tag{1}$$

whose eigenvalues are positive.

Definition 1 (Optimal Projector). *This is a projector which separates the region of acceptance from the region of rejection found according to Theorem 1.*

Definition 2 (Discriminant Function). *This is*

$$\mathrm{tr}((\rho_1 - \lambda\rho_0)\mathbf{E}) \tag{2}$$

where $\mathbf{E}$ *is the projector corresponding to an event and* tr *is the trace function.*

If the discriminant function is positive, the observed event represented by $\mathbf{E}$ is placed in the region of acceptance. The discriminant function generalizes the maximum likelihood ratio that is at the basis of probabilistic IR. The latter implements the density matrices by using the mixed case explained below.

Suppose, for example, that the event is term occurrence, p_1 is the probability that a term occurs in a relevant document and p_0 is the probability that a term occurs in a irrelevant document. The density matrix of a state i represents the following mixed distribution [4]:

$$\mu_i = p_i\mathbf{P}_1 + (1 - p_i)\mathbf{P}_0 \qquad \mathbf{P}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad \mathbf{P}_0 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{3}$$

Following Theorem 1, the projectors of the spectrum are $\mathbf{P}_0, \mathbf{P}_1$ and the region of acceptance is given by the optimal projectors whose eigenvalues are positive.

Table 1. Twenty documents have been used for training a data management system. Each document has been indexed using one binary feature and has been marked as relevant (1) or irrelevant (0).

document	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
feature	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0
relevance	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

The optimal projectors represent the absence and the presence, respectively, of a term. Thus, the decision on, say, webpage classification, topic categorization or item suggestion, can be made upon the occurrence of one or more features corresponding to $\mathbf{P}_0, \mathbf{P}_1$, which hence represent "physical" events; "physical" means that we can build a device (e.g. a parser) to detect the occurrence of a feature. The eigenvalues of the optimal projectors are, respectively:

$$(1 - p_1) - \lambda(1 - p_0) \qquad p_1 - \lambda p_0 \tag{4}$$

Therefore: if they are both positive, the region of acceptance is $\mathbf{P}_0 + \mathbf{P}_1 = \mathbf{I}$, that is, always accept; if either $(1 - p_1) - \lambda(1 - p_0)$ or $p_1 - \lambda p_0$ is positive, the region of acceptance is either $\mathbf{P}_0$ or $\mathbf{P}_1$, respectively, that is, accept only if the term either occurs or does not, respectively; if they are both not positive, always reject.

Consider the numerical example of Table 1. We have that, $p_1 = \frac{7}{10}, p_0 = \frac{3}{10}$. When $\lambda < \frac{3}{7}$, both eigenvalues are positive; when $\frac{3}{7} < \lambda < \frac{7}{3}$ an eigenvalue is negative whereas the other is positive.

Note that when $\mathbf{E}$ represents the region of acceptance, the discriminant function is positive if and only if the likelihood ratio of the classical probabilistic model [8] is higher than λ, thus showing that the classical probabilistic model is a special case. When the region of acceptance is represented by the projector $\mathbf{E}$, the power and the size are, respectively,

$$P_d = \mathrm{tr}(\mu_1 \mathbf{E}) \qquad P_0 = \mathrm{tr}(\mu_0 \mathbf{E}) \tag{5}$$

In particular, $P_d = p_1$, $P_0 = p_0$ when $\mathbf{E} = \mathbf{P}_1$.

The ROC curve can be built as illustrated in the appendix; if the previous example is considered, the curve is depicted in Fig. 1. The proof of the following corollary easily follows:

Corollary 1. *The optimal decision to accept the state represented by μ_1 is taken if $p_1 > \lambda p_0$ and the event represented by $\mathbf{P}_1$ is true, or $p_1 < \lambda p_0 + (1 - \lambda)$ and the event is false (i.e. $\mathbf{P}_0$).*

The key point is that a mixture is not the unique way to implement the probability distributions. The superposed vectors

$$|\varphi_1\rangle = \begin{pmatrix} \sqrt{p_1} \\ \sqrt{1 - p_1} \end{pmatrix} \qquad |\varphi_0\rangle = \begin{pmatrix} \sqrt{p_0} \\ \sqrt{1 - p_0} \end{pmatrix} \tag{6}$$

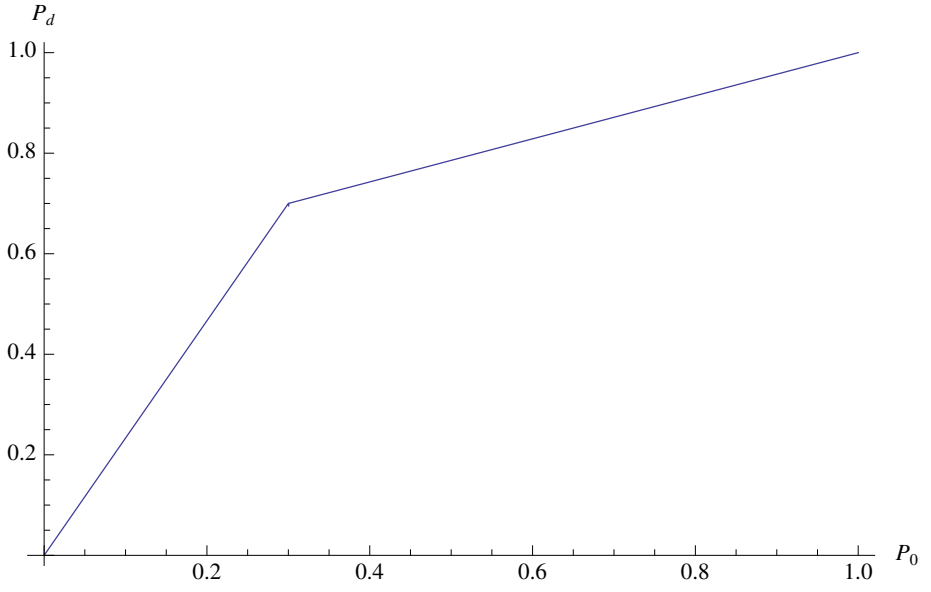


Fig. 1. A graphical representation of the ROC curve in the mixed case. The figure depicts the polygonal curve resulting from the mixed case.

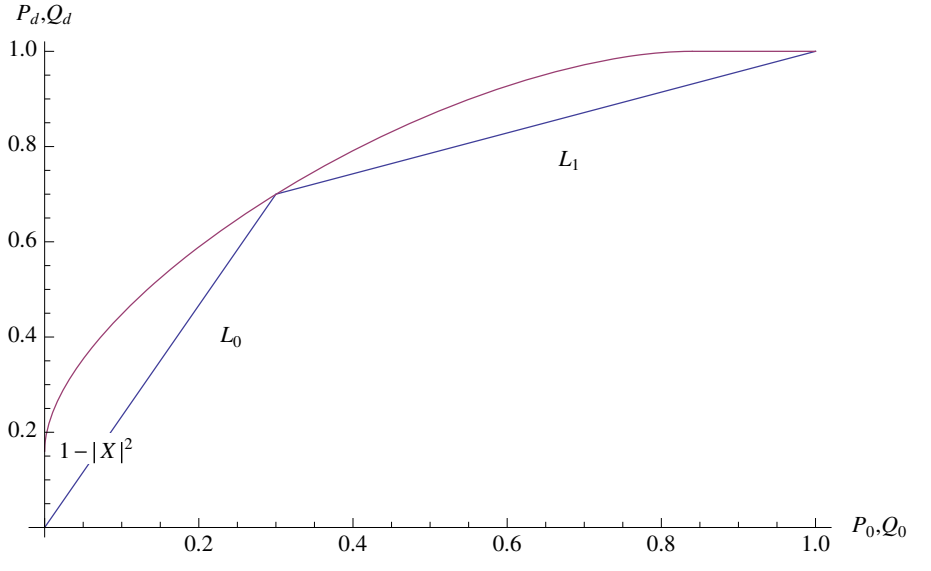


Fig. 2. A graphical representation of the ROC curves in the pure and mixed case. The figure depicts the polygonal curve resulting from the mixed case and the curve resulting from the pure case.

yield the pure densities

$$\rho_1 = |\varphi_1\rangle\langle\varphi_1| \qquad \rho_0 = |\varphi_0\rangle\langle\varphi_0| \tag{7}$$

and are an alternative to the mixed densities. The derivation of the probability of detection Q_d in the pure case is illustrated in Appendix B. Q_d is function of the probability of false alarm Q_0 in the pure case. Both Q_d and Q_0 ultimately depend on $|X|^2$, which is the squared cosine of the angle between the subspaces corresponding to the density vectors. The justification of viewing $|X|^2$ as a distance comes from the fact that "the angle in a Hilbert space is the only measure between subspaces, up to a constant factor, which is invariant under all unitary transformations, that is, under all possible time evolutions" [10].

Consider the example of Table 1; we have that $|X|^2 = \frac{\sqrt{21}}{5}$; the computation of Q_d, Q_0 follows from (21). Fig. 2 plots Q_d against Q_0 when p_0, p_1 are estimated using the example data.

Expressions (7) have no counterpart in classical probability, that is, it is not possible to express the quantum optimal projectors in terms of classical optimal projectors $\mathbf{P}_0, \mathbf{P}_1$ through classical logical operations [9]. This result is at the basis of this paper because it allows us to improve ranking while using the same amount of evidence as the evidence used in the classical probability distribution (3).

4 Optimal Projectors in the Quantum Space

We prove the following

Lemma 1. *$Q_d \geq P_d$ at every given probability of false alarm.*

Proof. The equality holds only if

$$\mathbf{P}_i = \mathbf{Q}_i \qquad i = 0, 1. \tag{8}$$

Indeed

$$\mathrm{tr}(\mu_i \mathbf{P}_1) = \mathrm{tr}(\rho_i \mathbf{P}_1) = p_i \qquad \mathrm{tr}(\mu_i \mathbf{P}_0) = \mathrm{tr}(\rho_i \mathbf{P}_0) = 1 - p_i \qquad i = 0, 1 \tag{9}$$

is an easy calculation.

Let x be a certain false alarm probability and let $Q_d(x), P_d(x)$ be the real, continuous functions yielding the detection probabilities at x. Q_d, which is defined by (21), admits the first and the second derivatives in the range $[0, 1]$. In particular, $Q_d'' < 0$ in $[0, 1]$.

In the mixed case, the optimal decision is provided by Corollary 1. It follows that the ROC of the mixed case is the set of points $(x, P_d(x))$ depicted by the polygonal curve of Fig. 2. Each segment of the polygonal corresponds to the polynomial of order 1 where x determines the diagonal values of μ_1 and $\mathbf{E} \in \{\mathbf{0}, \mathbf{P}_0, \mathbf{P}_1, \mathbf{I}\}$. Thus, P_d is a continuous function.

Consider the polynomial $L_0(x)$ of order 1 passing through the points $(0, 1 - |X|^2)$ and (p_0, p_1) at which L_0 intersects Q_d. Then, the Lagrange interpolation

theorem can be used so that $Q_d(x) - L_0(x) = Q''_d(c)\frac{x(x-p_0)}{2}$ the latter being non negative because $Q''_d < 0$ and $0 \leq x \leq p_0$. The number $c \in [0, p_0]$ exists due to the Rolle theorem. As $L_0(x) \geq P_d(x), x \in [0, p_0]$, hence, $Q_d(x) \geq P_d(x), x \in [0, p_0]$. Similarly, consider the polynomials $L_1(x)$ and $L_2(x)$ of order 1 passing through the points (p_0, p_1), $(1-p_0, 1-p_1)$ and $(1-p_0, 1-p_1)$, $(1, 1)$ at which L_1 and L_2 intersect Q_d, respectively. Then, the Lagrange interpolation theorem can again be used so that

$$Q_d(x) - L_1(x) = Q''_d(c)\frac{(x-p_0)(x-1+p_0)}{2}$$
$$Q_d(x) - L_2(x) = Q''_d(c)\frac{(x-1+p_0)(x-1)}{2}$$

Then, $Q_d(x) \geq P_d(x)$ for all $x \in [0, 1]$.

5 Quantum Probability Gives a Different Ranking

Lemma 1 shows that the power (i.e. recall) of the decision rule in quantum probability is greater than, or equal to, the power of the decision rule in classical probability with the same amount of information available from the training set to estimate p_0, p_1 (e.g. the probability that a keyword occurs in a (non-) relevant document) and at every probability of false alarm (i.e. fallout).

In this section we show that the improvement is not due to a different estimation of p_0, p_1, but it is due to the pure case which leads to a different ranking possible only in quantum probability.

The density vectors $|\varphi_1\rangle, |\varphi_0\rangle$ are linear combinations of two different bases, i.e. $|0\rangle, |1\rangle$ and $|\eta_0\rangle, |\eta_1\rangle$, at the same time; the former are equivalent to the optimal projectors in the mixed case, the latter to optimal projectors in the pure case. When $|0\rangle, |1\rangle$ is the basis, the coordinates of $|\varphi_i\rangle$ are $\sqrt{p_i}, \sqrt{1-p_i}$. When $|\eta_0\rangle, |\eta_1\rangle$ is the basis, the coordinates of $|\varphi_i\rangle$ are $x_{00}, x_{01}, x_{10}, x_{11}$ such that

$$|\varphi_0\rangle = x_{00}|\eta_0\rangle + x_{01}|\eta_1\rangle \qquad |\varphi_1\rangle = x_{10}|\eta_0\rangle + x_{11}|\eta_1\rangle \tag{10}$$

$$x^2_{00} = \frac{|X|^2}{(1-\eta_1)^2 + |X|^2} \qquad x^2_{01} = \frac{(1-\eta_1)^2}{(1-\eta_1)^2 + |X|^2} \tag{11}$$

$$x^2_{10} = \frac{|X|^2}{(1+\eta_1)^2 + |X|^2} \qquad x^2_{11} = \frac{(1+\eta_1)^2}{(1+\eta_1)^2 + |X|^2} \tag{12}$$

To answer the question whether the pure case leads to a different ranking, we wonder if there are p_0, p_1, λ such that the region of acceptance in the pure case differs from that in the mixed case. Consider Theorem 1 to answer the question. The region of acceptance in the mixed case is defined through Table 2 whereas the region of acceptance in the pure case is defined through Table 3. Furthermore, the discriminant function is

$$\mathrm{tr}((\sigma_1 - \lambda\sigma_0)\mathbf{E}) \qquad \mathbf{E} \in \{\mathbf{0}, \mathbf{Q}_0, \mathbf{Q}_1, \mathbf{I}\} \tag{13}$$

Table 2. The regions of acceptance corresponding to the sign of the eigenvalues of the spectrum of the discriminant function in the mixed case. As for zero eigenvalues, see [3].

	$p_1 - \lambda p_0$	
$1 - p_1 - \lambda(1 - p_0)$	< 0	> 0
< 0	$\mathbf{0}$	$\mathbf{P}_1$
> 0	$\mathbf{P}_0$	$\mathbf{I}$

Table 3. The regions of acceptance corresponding to the sign of the eigenvalues of the spectrum of the discriminant function in the pure case. As for zero eigenvalues, see [3].

	$\lvert x_{11}\rvert^2 - \lambda\lvert x_{01}\rvert^2$	
$1 - \lvert x_{11}\rvert^2 - \lambda(1 - \lvert x_{01}\rvert^2)$	< 0	> 0
< 0	$\mathbf{0}$	$\mathbf{Q}_1$
> 0	$\mathbf{Q}_0$	$\mathbf{I}$

where

$$\sigma_i = \begin{pmatrix} |x_{i1}|^2 & |x_{i1}|\sqrt{1 - |x_{i1}|^2} \\ |x_{i1}|\sqrt{1 - |x_{i1}|^2} & 1 - |x_{i1}|^2 \end{pmatrix} \qquad i = 0, 1 \tag{14}$$

Corollary 2. *The discriminant function of the mixed case ranks documents in a different way from the discriminant function of the pure case.*

Proof. Consider the example data of Table 1. It follows that $X^2 = \frac{21}{25}$ and $R = \sqrt{\frac{1}{4}(1 - \lambda)^2 + \frac{4\lambda}{25}}$. Fig. 3 depicts the values of the four eigenvalues (two eigenvalues of the mixed case, two eigenvalues of the pure case) as λ varies between 0 and 2. The plot shows that there is at least one value of λ such that the decisions contrast each other; for example, when $\lambda < \frac{3}{7}$, the mixed case always suggests acceptance, while the pure case always suggests to accept only if the event corresponding to $\mathbf{Q}_1$ is observed.

Let's see how a system can use these results in practice. It reads the feature occurrence symbol (i.e. either 0 or 1); check whether the feature is included by the region of acceptance. If the feature is not included, relevance is rejected.

Another view of the preceding decision rule is the ranking of the information units. When ranking documents, the system returns the units whose features lead to the highest probability of detection, then those whose features lead to the second highest probability of detection, and so on.

When the previous example is considered and $\lambda > \frac{3}{7}$, the ranking ends up to placing the documents that include the feature on the top and those that do not include it on the bottom of the list. The performance is described by Fig. 2.

Suppose that the system can recognize the events corresponding to $\mathbf{Q}_0, \mathbf{Q}_1$. Then, for any λ, the ranking ends up to placing the documents that make the

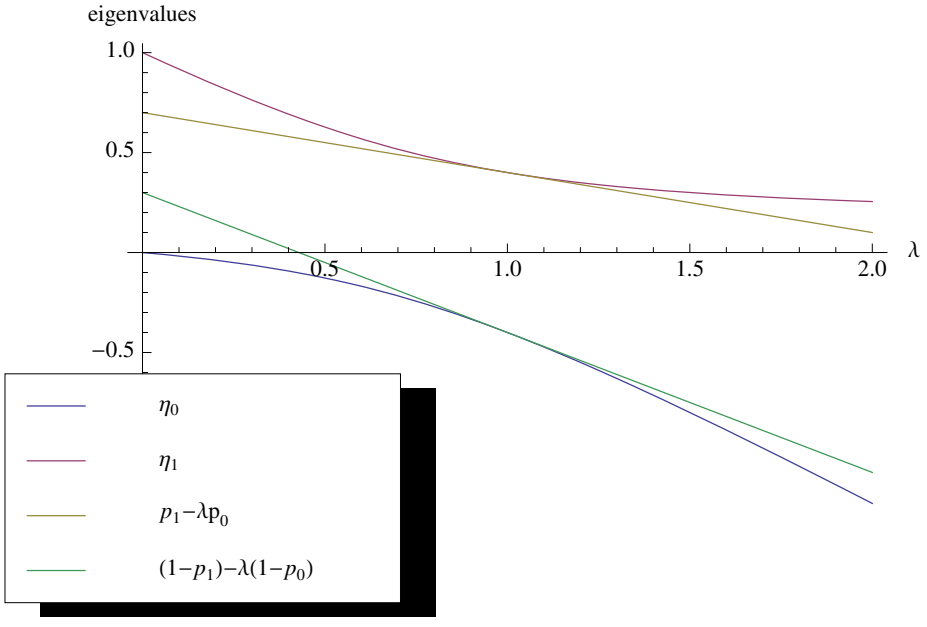


Fig. 3. The four eigenvalues plotted against λ

event true on the top and those that do not make it true on the bottom of the list. The performance is described by Fig. 3.

The observation of the features corresponding to $\mathbf{P}_0, \mathbf{P}_1$ cannot give any information about the observation of the events corresponding to $\mathbf{Q}_0, \mathbf{Q}_1$ due to the incompatibility between these pairs of events [4]. Thus, the design of an algorithm that implements the decision rule so that the observation of a feature can be translated into the observation of the events corresponding to $\mathbf{Q}_0, \mathbf{Q}_1$ is still an open problem.

6 Future Developments and Conclusions

The improvements obtained through estimators or ranking functions in the past cannot be more effective than those stated by the PRP under the assumptions stated in [7] and the fact that a region of acceptance is based on subsets. We conjecture that asking a different question (i.e. what if subspaces are used?) is more effective than looking for better answers (i.e. better subsets, better parameter estimations) to old questions.

We have proved that a significant improvement can in principle be attained if subspaces are used, but we need a device that produces "yes" when the event represented by $\mathbf{Q}_1$ is true in a document for implementing this improvement. The design of such a device is not trivial at all because those events do not correspond to the "physical" events (e.g. feature occurrence) with which IR systems deal.

The future developments are threefold. First, we will work on the interpretation of the optimal projectors in the pure case because the detection of them in a document is problematic, but this may open further insights. Second, multivariate features and quantum entanglement will be investigated. Third, empirical evaluation is crucial to understanding whether the results of the paper can be confirmed by the experiments.

References

1. Accardi, L.: On the probabilistic roots of the quantum mechanical paradoxes. In: Diner, S., de Broglie, L. (eds.) The Wave-Particle Dualism, pp. 297–330. D. Reidel pub. co., Dordrecht (1984)
2. Cariolaro, G., Pierobon, G.: Performance of quantum data transmission systems in the presence of thermal noise. IEEE Transactions on Communications 58, 623–630 (2010)
3. Helstrom, C.: Quantum detection and estimation theory. Academic Press, London (1976)
4. Melucci, M., van Rijsbergen, K.: Quantum mechanics and information retrieval. In: Advanced Topics in Information Retrieval. Springer, Heidelberg (2011)
5. Neyman, J., Pearson, E.: On the problem of the most efficient tests of statistical hypotheses. Philosophical Transactions of the Royal Society, Series A 231, 289–337 (1933)
6. Piwowarski, B., Frommholz, I., Lalmas, M., van Rijsbergen, K.: What can quantum theory bring to information retrieval? In: Proc. 19th International Conference on Information and Knowledge Management, pp. 59–68 (2010)
7. Robertson, S.: The probability ranking principle in information retrieval. Journal of Documentation 33(4), 294–304 (1977)
8. van Rijsbergen, C.: Information Retrieval, 2nd edn., Butterworths, London (1979)
9. van Rijsbergen, C.: The geometry of information retrieval. Cambridge University Press, UK (2004)
10. Wootters, W.K.: Statistical distance and Hilbert space. Phys. Rev. D 23(2), 357–362 (1981)
11. Zuccon, G., Azzopardi, L., van Rijsbergen, C.: The quantum probability ranking principle for information retrieval. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 232–240. Springer, Heidelberg (2009)

A The ROC Curve in the Mixed Case

This section follows [3, pages 15–16]. Suppose, as an example, that a webpage either includes or does not include a term. Using the quantum probability formalism, these two events respectively correspond to the projectors $\mathbf{P}_1$, $\mathbf{P}_0$. In constrast, these two events respectively correspond to the propositions $x = 1$, $x = 0$ using classical probability. The likelihood functions under the hypothesis of relevance (irrelevance) are given by the Bernoulli distributions with parameter p_1 (p_0), respectively. Hence, the likelihood ratio will be $p_1^x(1-p_1)^{1-x}/p_0^x(1-p_0)^{1-x}$. The system shall decide for relevance when x exceeds a certain number v and

irrelevance when $x < v$. When $x = v$, however, the system shall choose relevance with such a probability r that

$$P_0 = \sum_{x>v} p_0^x (1-p_0)^{1-x} + r p_0^v (1-p_0)^{1-v} \tag{15}$$

equals the preassigned false alarm probability (or size). As $P_d = \sum_{x>v} p_1^x (1-p_1)^{1-x} + r p_1^v (1-p_1)^{1-v}$ and r is calculated from (15), it follows that

$$P_d = \sum_{x>v} p_1^x (1-p_1)^{1-x} + \frac{P_0 - \sum_{x>v} p_0^x (1-p_0)^{1-x}}{p_0^v (1-p_0)^{1-v}} p_1^v (1-p_1)^{1-v} \tag{16}$$

The decision level v is either 0 or 1, thus we have that for each v, P_d is a linear function of P_0. If $p_0 = \frac{3}{10}, p_1 = \frac{7}{10}$ the ROC curve is depicted in Fig. 1. Therefore,

$$P_d = \begin{cases} \frac{1-p_1}{1-p_0} P_0 + \frac{p_1 - p_0}{1-p_0} & v = 0 \\ \frac{p_1}{p_0} P_0 & v = 1 \end{cases} \tag{17}$$

B The ROC Curve in the Pure Case

This section follows [3, pages 112–113]. Theorem 1 instructs us to define the optimal projectors as those of the spectrum of (11) whose eigenvalues are positive, the spectrum being

$$\eta_0 \mathbf{Q}_0 + \eta_1 \mathbf{Q}_1 \qquad \mathbf{Q}_0 = |\eta_0\rangle\langle\eta_0| \qquad \mathbf{Q}_1 = |\eta_1\rangle\langle\eta_1| \tag{18}$$

where the η's are eigenvalues,

$$\eta_0 = -R + \frac{1}{2}(1-\lambda) < 0 \qquad \eta_1 = +R + \frac{1}{2}(1-\lambda) > 0 \tag{19}$$

and

$$R = \sqrt{\frac{1}{4}(1-\lambda)^2 + \lambda(1-|X|^2)} \qquad X = \sqrt{p_0}\sqrt{p_1} + \sqrt{1-p_0}\sqrt{1-p_1} \tag{20}$$

The eigenvalue η_1 is a measure of the extent to which the states are separated and then well detectable. Both eigenvalues are positive when $\lambda < 1 - 2R$.

The probability of detection (i.e. the power) Q_d and the probability of false alarm (i.e. the size) Q_0 in the pure case are defined as follows:

$$Q_d = \frac{\eta_1 + \lambda(1-|X|^2)}{2R} \qquad Q_0 = \frac{\eta_1 - (1-|X|^2)}{2R} \tag{21}$$

Finally, Q_d can be defined as a function of Q_0:

$$Q_d = \begin{cases} \left(\sqrt{Q_0}\sqrt{|X|^2} + \sqrt{1-Q_0}\sqrt{1-|X|^2}\right)^2 & 0 \le Q_0 \le |X|^2 \\ 1 & |X|^2 < Q_0 \le 1 \end{cases} \tag{22}$$

so that the power curve is obtained.

An Analysis of Ranking Principles and Retrieval Strategies

Guido Zuccon*, Leif Azzopardi, and C.J. Keith Van Rijsbergen

School of Computing Science
University of Glasgow
Scotland, UK
{guido,leif,keith}@dcs.gla.ac.uk

Abstract. The assumptions underlying the Probability Ranking Principle (PRP) have led to a number of alternative approaches that cater or compensate for the PRP's limitations. All alternatives deviate from the PRP by incorporating dependencies. This results in a re-ranking that promotes or demotes documents depending upon their relationship with the documents that have been already ranked. In this paper, we compare and contrast the behaviour of state-of-the-art ranking strategies and principles. To do so, we tease out analytical relationships between the ranking approaches and we investigate the document kinematics to visualise the effects of the different approaches on document ranking.

1 Introduction

The Probability Ranking Principle (PRP) has played a central role in the development of Information Retrieval (IR). The PRP has largely stood the test of time for adhoc retrieval, but for emerging retrieval tasks, such as novelty and diversity, the assumptions made by the PRP have been shown to lead to non-optimal performance [2,5,7]. Alternative ranking approaches have been proposed; these include two ranking strategies, Maximal Marginal Relevance (MMR) [1] and Portfolio Theory (PT) [7], along with the Quantum PRP (qPRP) [8], and the Interactive PRP (iPRP) [3]. Each approach can be regarded as a revision of the PRP, where the point of departure is the introduction of document dependent evidence within the revised ranking. The function used for revising a ranking may be formulated differently, depending upon the ranking approach. However, the net effect of the revision boils down to the promotion of diversity, i.e. documents which are different from those previously seen in the ranking are promoted up in the ranking, or of similarity, i.e. documents that are similar to the previous one, obtaining a sort of pseudo-relevance feedback effect.

While there has been a lot of interest in this area and a number of empirical comparisons, there has been no formal analysis of these approaches. Given that these new approaches attempt to address the same problem, it is important

* Supported by EPSRC Grant number EP/F014384/ and Zirak s.r.l. (http://www.zirak.it/)

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 151–163, 2011.

to identify specifically and formally relationships, similarities and differences between methods, in order to contextualise existing methods and to develop improved theory.

To this end, we perform a comprehensive theoretical analysis and comparison of ranking principles and strategies. We first introduce each approach in section 2, establishing a common framework, which allows us to further contrast them from an analytical perspective. Indeed, in section 3 we tease out relationships among approaches by analysing their ranking behaviour within a small scale controlled scenario. The analysis is completed in section 4 where we investigate the document kinematics that different approaches impose on the rankings.

2 Principles and Strategies

Approaches to ranking can be divided into two categories:

strategies that are empirically driven and devised to cater for the limitations of the PRP, i.e. Maximal Marginal Relevance [1] and Portfolio Theory [7], and,

principles that are theoretically driven and implicitly cater for the limitations of the PRP, i.e. the interactive PRP [3] and quantum PRP [8].

Regardless of the approach, strategy or principle, the recently proposed alternatives to the PRP mathematically deviate through the inclusion of a function that captures dependencies between documents. This function expresses the relationship between documents: depending upon how the function is set, the ranking approach promotes either document diversity or similarity. As we shall see, alternatives differ in the way dependencies are incorporated, and the extent of parameterisation of the ranking formula. Specifically, PT and qPRP are characterised by an additive ranking function, MMR by an interpolated and iPRP by a multiplicative, where PT and MMR are by definition parameterised. On the contrary, in their original formulations iPRP and qPRP do not have parameters. However, parametric instantiations may be formulated as well for qPRP and iPRP.

Next we will provide the formal analysis to justify the previous statements by providing a common framework to describe each of the principles and strategies, so that we can compare them analytically in a straightforward manner.

Probability Ranking Principle

The PRP states that documents should be retrieved in decreasing order of their estimated probability of relevance given the query [6]. By adhering to the PRP, at each rank position i the IR system should select a document d_i such that:

$$d_i = \underset{d \in \mathcal{RE} \setminus RA}{\arg\max}\ P(d) \tag{1}$$

where $P(d)$ is the probability of a document being relevant given the query, RA is the list of documents that have been ranked, and d is a document belonging

to the set of retrieved documents ($\mathcal{RE}$). Ranking according to this criteria has been shown to provide the optimal ranking [6]. This, however, depends upon a number of assumptions; of those the most criticised are:

(i) the independent assessment of document relevance (i.e. independence assumption); and
(ii) the certainty of the estimation of relevance.

Goffman noticed that by assuming independence between document's relevance assessments, the "relationship between query and the document is both necessary and sufficient to establish relevance" [4]. It has been argued [2,5] that this is not strictly the case in real search scenarios, where document's relevance depends upon information acquired during the course of the retrieval process. Goffman formalised this intuition as follows: the relevance of a document must depend upon what is already known at the time the document is examined by the user. If a document d has been judged relevant to a particular information need, the relevance of other documents might be affected by the relevant information already known. Gordon and Lenk have demonstrated the sub-optimality of the PRP when the independence assumption does not hold [5]. While, Chen and Karger showed that the PRP is not always optimal for different information needs [2]. These limitations and a number of empirical observations regarding the PRP have motivated a number of alternative ranking strategies and principles.

2.1 Alternatives to the PRP

In the following we consider ranking approaches alternative to the PRP. A common trend between these alternatives is the presence in the ranking function of two main elements: (1) the probability of relevance, or score of the document; and (2) a function that estimates the similarity between the representations of two documents. To facilitate comparison, we reformulate the approaches in a common framework, so that their ranking formulas are written with respect to a common estimation of the probability of relevance for a document d (represented by $P(d)$), and a common similarity function between documents. In the following we select the Pearson's correlation coefficient[1] $\rho_{d,d'}$ as measure of similarity between d and d'.

Maximal Marginal Relevance

In Maximal Marginal Relevance (MMR) [1], an hyper-parameter λ is used to balance the similarity between document and query, and the similarity between the candidate document and documents ranked at earlier positions. A document at rank i is selected using the following objective function:

[1] This choice is motivated by the fact that Pearson's correlation is used within PT and in previous instantiations of the qPRP. The choice of similarity function across all ranking approaches is however rather arbitrary: we have kept them all the same so that the quintessential differences between approaches can be teased out.

$$d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(\lambda s(d,q) - (1-\lambda) \max_{d' \in RA} sim(d,d') \right)$$

where $s(d,q)$ is a similarity function between document and query, while $sim(d,d')$ is a function that determines the similarity between documents d and d'. If two candidate documents have the same probability of relevance (or $s(d,q)$), MMR will rank first the one that is least similar to any of the documents that have been ranked at previous positions. The hyper-parameter can be inferred by the user's model: $\lambda < 0.5$ characterises users with a preference for rankings where document dependencies are more important than relevance. Greater values of λ would capture the converse situation. For consistency, we re-state MMR in terms of $P(d)$ and $\rho_{d,d'}$ in place of $s(d,q)$ and $sim(d,d')$, respectively:

$$\textbf{MMR: } d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(\lambda P(d) - (1-\lambda) \max_{d' \in RA} \rho_{d,d'} \right) \tag{2}$$

Portfolio Theory

Portfolio Theory applied to IR [7] attempts to minimise the risk associated with ranking documents under uncertainty in their relevance estimates by balancing the expected relevance value (mean) and its variance. The ranking criteria combines the estimated document relevance with (i) an additive term which synthesises the risk inclination of the user, (ii) the uncertainty (variance) associated with the probability estimation, and (iii) the sum of the correlations between the candidate document and documents ranked in previous positions. For each rank position i, documents are selected according to:

$$\textbf{PT: } d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(P(d) - b w_d \sigma_d^2 - 2b \sum_{d' \in RA} w_{d'} \sigma_d \sigma_{d'} \rho_{d,d'} \right) \tag{3}$$

where b encodes the risk propensity of the user, σ_d^2 is the variance associated to $P(d)$, and w_d is a weight that expresses the importance of the rank position of d and d'. When PT has been employed in practice, σ_d^2 has been treated as a model parameter (see [7,8]), because a single point-wise relevance estimation is used: in the rest of the paper we follow the same route.

Interactive PRP

In [3], Fuhr proposes a theoretical framework for extending the PRP to the context of interactive IR where the independence assumption is rejected. This is because in interactive searches relevance depends on documents the user has previously examined. Search is therefore modelled as situation, i.e. a list of choices the user is presented with: users move between situations by accepting one of the choices they are provided with. Once a choice is accepted, the retrieval system produces a new list of choices dependent from the previous choice. The ranking

principle strives to provide the optimal ordering of the choices presented in each situation. For each rank i, documents under the iPRP are ranked as follows:

$$d_i = \underset{d \in \mathcal{RE} \setminus RA}{\arg\max} \left[e + P(d) \left(b_{d,i} Q(d) + g(1 - Q(d)) \right) \right] \text{ , where}$$

- $Q(.)$ is the probability that the user does not revise their choice of selecting document d (i.e. the probability that the user does not change their mind about the relevance of the document d after examining it);
- e is the effort of examining document d;
- g is the additional effort required for correction if the user judges a viewed document as irrelevant;
- $b_{d,i}$ is the benefit of ranking document d at rank i if the document is relevant.

In this study, we provide a possible instantiation of the iPRP for the first pass of retrieval (i.e. before any actual user interaction has transpired): in this context we do *not* consider any further interaction or re-ranking. This instantiation is in line with the assumptions of [3], and had been first proposed in [10]. Since we are examining the case of the first pass of retrieval, we assume e, g and $Q(.)$ as constants. These can then be dropped for rank equivalence reasons. We then consider the benefit of ranking d at rank i. A reasonable approximation would be to determine how similar the current candidate document is with all previous documents. This is because $b_{d,i}$ is dependent upon previously ranked documents. We achieve this through a summation over all previously ranked documents of the negative correlation[2] between previously ranked documents and d. If document d is similar to previous documents, then the correlation will be low, and possibly negative: the total benefit achieved will thus be low. Similar documents are demoted in the ranking, while diverse documents are promoted, giving rise to the following objective function:

$$\textbf{iPRP: } d_i = \underset{d \in \mathcal{RE} \setminus RA}{\arg\max} \left(P(d) b_{d,i} \right) = \underset{d \in \mathcal{RE} \setminus RA}{\arg\max} \left(-P(d) \frac{\sum_{d' \in RA} \rho_{d,d'}}{|RA|} \right) \quad (4)$$

Under the iPRP dependencies between documents are incorporated through *multiplication*, providing a completely different approach to the other alternatives.

Quantum PRP

The qPRP develops from quantum probability theory (as opposed to traditional Kolmogorovian probability theory), and naturally incorporates dependencies between documents through the notion of *quantum interference* [8]. In order to obtain the most valuable document ranking for a user the total probability of relevance of the ranking needs to be maximised. The interference $I_{d,d'}$ between

[2] A negative value implies a cost to the user. This might occur when examining relevant but redundant information.

two documents influences the total probability of relevance (see [8]). The qPRP then selects a document d to be ranked at position i such that:

$$d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(P(d) + \sum_{d' \in RA} I_{d,d'} \right)$$

The underlying intuition is that documents in a ranking share relationships at relevance level, i.e. they interfere with each other, and the interference has to be taken into account when ranking documents. According to [8], interference can be approximated via a function such as the correlation $\rho_{d,d'}$ between documents[3], where $I_{d,d'} = -2\sqrt{P(d)}\sqrt{P(d')}\rho_{d,d'}$. Therefore, the ranking rule becomes:

$$\textbf{qPRP: } d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(P(d) - 2 \sum_{d' \in RA} \sqrt{P(d)}\sqrt{P(d')}\rho_{d,d'} \right) \tag{5}$$

2.2 Parametric Instantiations of iPRP and qPRP

While MMR and PT are by definition characterised by the settings of their parameters, the instantiations of iPRP and qPRP of Eqs 4 and 5 are not parametric. However, parametric instantiations of these principles can be given, where parameters control the impact of correlation on the ranking process. The parameter is formally introduced within the approximations of benefit and interference.

When instantiating the iPRP, the benefit of ranking a document d at rank i (i.e. $b_{d,i}$) has been approximated as $-\frac{\sum_{d' \in RA} \rho_{d,d'}}{|RA|}$. A possible parametric instantiation of the iPRP is obtainable by setting $b_{d,i} = -\beta \frac{\sum_{d' \in RA} \rho_{d,d'}}{|RA|}$, with β being a free parameter (and $\beta \in \mathbb{R}$). Therefore, the ranking formula of iPRP becomes:

$$\textbf{iPRP(parametric): } d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(-\beta P(d) \frac{\sum_{d' \in RA} \rho_{d,d'}}{|RA|} \right) \tag{6}$$

Similarly, when operationalising the qPRP, interferences have been approximated as $I_{d,d'} = -2\sqrt{P(d)}\sqrt{P(d')}\rho_{d,d'}$. Alternative approximations have been investigated in [9]: these considered similarity functions other than Pearson's correlation for estimating interferences and no parameter was introduced. We can however consider a parametric instantiation of the qPRP, by introducing the parameter β in the approximation of the interference term, obtaining:

$$\textbf{qPRP(parametric): } d_i = \arg\max_{d \in \mathcal{RE} \setminus RA} \left(P(d) - 2\beta \sum_{d' \in RA} \sqrt{P(d)}\sqrt{P(d')}\rho_{d,d'} \right) \tag{7}$$

The first contribution of this paper is the common framework for describing ranking approaches. Using this framework we can now perform an analysis of their ranking behaviour and of the kinematics imposed on relevant documents.

[3] While $\sqrt{P(d)}\sqrt{P(d')}$ is the magnitude of the complex probability amplitudes associated to documents d and d'.

3 Analysis of Ranking Behaviours

Each approach handles document dependencies in a characteristically different way. The question is: *How do different approaches affect document ranking?*

To answer this question, we shall consider two aspects: (1) what document is ranked first?, and (2) what documents are then subsequently ranked next?

For all approaches, the document ranked at fist position (i.e. $i = 1$) is the same. This is the document which has the highest probability of relevance. Differences between alternatives and the PRP manifest at ranks greater than one. At $i > 1$, each alternative approach will tend to revise the original ranking such that documents which are different to those ranked previously will be promoted. To obtained deeper intuition of this phenomena for each ranking alternative, we analytically compare each method at the functional level to determine more precisely how the ranking of documents would be affected.

To this aim, we shall consider the following example scenario, where we have two documents, d and d', with the same probability of relevance, i.e. $P(d) = P(d')$, and d has been ranked first. We are interested to determine what is likely to happen to d' given the PRP, MMR, PT, iPRP, and qPRP: i.e. is it likely to be demoted or promoted? We consider three further cases, where documents d, d' are:

case 1: virtually identical[4] and thus positively correlated, i.e. $\rho_{d,d'} = 1$;
case 2: with nothing in common, and thus not correlated at all, i.e. $\rho_{d,d'} = 0$;
case 3: sharing the same terms, but with complete different use and frequencies, and thus anti-correlated[5], i.e. $\rho_{d,d'} = -1$.

Probability Ranking Principle. The behaviour of the PRP does not depend on the correlation, so the PRP always ranks documents d and d' consecutively, and actually both $(d, d', ...)$ and $(d', d, ...)$ are valid rankings.

Maximal Marginal Relevance. When documents are correlated (case 1), MMR assigns to d' the score $\lambda P(d') - (1 - \lambda)$, which might assume negative values. If $\lambda = 1$ then MMR reduces to PRP, while if $\lambda = 0$ document d' gets a score of 1. For $0 < \lambda < 1$, the original score of $P(d')$ is remodulated by λ and then decreased of $(1 - \lambda)$. In case 2, MMR rescales the document's probability by the hyper-parameter, assigning to d' the score $\lambda P(d')$. The document score increases in the third case, i.e. when the correlation has negative value, adding to the (re-scaled) probability of the document a value proportional to $1 - \lambda$: if $\rho_{d,d'} = -1$, then the score of d' is $\lambda P(d') + 1 - \lambda$.

[4] We consider the document term vectors to compute correlations (and thus dependencies): term-position does not influence correlation, while term's (weighted) presence does. Two documents containing the same exact text, but shuffled in different orders, will appear identical to the correlation function.

[5] While in practice correlations of -1 are unlikely, there might be cases where correlations are negative because of the weighting schema used to compute document term vectors. However, for the purpose of our example, we imagine the two documents to be completely anti-correlated.

Portfolio Theory. The score PT assigns to a document differs to the one provided by the PRP of $-bw_d\sigma_d^2 - 2bw_{d'}\sigma_d\sigma_{d'}\rho_{d,d'}$. The sign of PT's variation in scores, i.e. increment or decrement, are then not only dependent upon the correlation's sign, but also upon the user's model parameter b. We focus our analysis on the situation where $b > 0$: under this circumstance PT promotes diversity in the document ranking. The initial document probability of relevance is revised of $-|b|w_d\sigma_d^2 - 2|b|w_{d'}\sigma_d\sigma_{d'}\rho_{d,d'}$. In case 1, i.e. $\rho_{d,d'} = 1$, the score of d' is decreased by $-|b|w_d\sigma_d^2 - 2|b|w_{d'}\sigma_d\sigma_{d'}$. If documents are not correlated (case 2), the initial score undergoes a limited decrement of $|b|w_d\sigma_d^2$. Finally, in case 3 (anti-correlated documents), the initial score of d' is modified by PTs's ranking formula of $-|b|w_d\sigma_d^2 + 2|b|w_{d'}\sigma_d\sigma_{d'} \approx |b|\sigma_d^2(2w_{d'} - w_d)$. The discount factor w_d is estimated through a monotonically decreasing function of the document's rank position, thus $2w_{d'} - w_d$ can be either positive or negative. If positive, d''s score gets incremented; vice versa, d' gets demoted in the document ranking. Finally, when $b = 0$ PT's ranking function reduces to the one of the PRP.

Interactive PRP. The iPRP is characterised by a multiplicative ranking function. When d and d' are completely correlated (case 1), iPRPs assigns to d' the score $-P(d')$, and thus the document is demoted: documents that are more relevant than others would suffer a stronger demotion. In the situation of zero-correlated documents (case 2), d' gets assigned a score of zero and is demoted in the ranking. In case 3, iPRPs assigns to d' the same score obtained with the PRP, i.e. $P(d')$, and thus d' is ranked immediately after d (as in the PRP).

Quantum PRP. When documents correlate, as in case 1, the probability assigned to d' is revised and is modified to the value $-P(d')$: this is due to the interference term becoming $I_{d,d'} = -2\sqrt{P(d)}\sqrt{P(d')} = -2P(d')$. In this situation, as for other models, also according to the qPRP d''s chances to get ranked at second position are decreased, possibly demoting it to lower positions. When d and d' are not correlated at all as in case 2, i.e. $\rho_{d,d'} = 0$, qPRP does not change PRP's estimate since the interference term is zero: there is no dependence between the actual candidate and the previous ranked document. In case 3, qPRP boost the original probability of d' to the quantity $3 \cdot P(d')$. In fact, the interference term results $I_{d,d'} = 2\sqrt{P(d)}\sqrt{P(d')} = 2P(d')$.

Summary. The approaches revealed a common pattern. When promoting diversity, the initial probability estimation associated to d', i.e. $P(d')$, is revised by a quantity proportional to the correlation of d' with those documents that have been already ranked. The revision increments the initial probability estimation if documents are anti-correlated. Vice versa if documents are correlated, the document score is decreased. The case of no correlation (case 2) is handled differently by each ranking approach: for example iPRP assigns to the document a zero score, while qPRP returns the same probability estimation of PRP.

Finally, the amount of revision that the score of a document is subject to depends upon the parametrisation of the ranking function. Specifically:

Table 1. Overview of the characteristics of the ranking principles and strategies

Model	Dependence	Parameters	$\rho = 1$	$\rho = 0$	$\rho = -1$
PRP	-	-	∘	∘	∘
MMR	Interpolated	λ: hyperparameter	↓	∼PRP	↑
PT	Additive	b: user risk propensity σ: variance estimation relevance w: discount rank position	↓ (if $b > 0$)	∼PRP	↑ (if $b > 0$)
iPRP	Multiplicative	-	↓	0	↑
qPRP	Additive	-	↓	=PRP	↑

- MMR weights the contribution of the correlation depending on λ; high values of λ (i.e. $\lambda \to 1$) return rankings similar to those of PRP;
- PT modulates the contribution of the correlation by the product of the parameters b and σ_d^2, and considering the importance of the rank position;
- iPRP reduces the influence of the correlation by a quantity inversely proportional to the number of documents retrieved at previous ranks;
- qPRP modulates the contribution of the correlation by the square root of the probabilities of the documents involved in the comparison.

4 Kinematics of Documents

To provide a deeper understanding of the revision process, in the following we empirically explore the movement of the relevant documents.

To do so, we employ the Clueweb09 collection (part B only) and the TREC 2009-2010 Web Diversity topics and relevance judgements. Documents and queries were stemmed and stop-words were removed: thereafter documents were indexed using the Lemur 4.10 toolkit[6]. Documents were retrieved according to a unigram language model with Dirichlet smoothing ($\mu = 2,500$): for each query, the 100 documents with higher score were considered for ranking. The PRP ranking was formed arranging documents in decrease order of scores. Approaches alternative to the PRP were used to re-rank documents. For PT, we regarded both the variance of the probability estimations (σ^2) and b as parameters, and we let them varying in the ranges $[10^{-7}, 10^{-2}]$ (with decimal increments) and $[-10, +10]$ (with unitary increments), respectively. MMR's hyper-parameter was varied in the range $[0, 1]$ with steps of 0.1. We considered the parametric versions of iPRP and qPRP (Eqs. 6 and 7), studying values of β varying in the range $[-1, 1]$ with steps of 0.1. Pearson's correlation between (normalised) term frequency representations of documents was employed in all re-ranking approaches.

For each ranking approach, we built a retrieval run by tuning the parameters with respect to α-NDCG@10[7] on a query-by-query basis: that is, for each query, we rank documents using the best parameter values for the query.

[6] http://lemurproject.org/

[7] With $\alpha = 0.5$, set according to the TREC 2009 and 2010 Web Track guidelines.

While our focus is on the kinematics of documents, we report the performance of the runs, to show how the re-ranking affects performance. Specifically, the approaches obtained the following values of α-NDCG@10[8]:

$$\text{PRP: } 0.137 < \text{qPRP: } 0.172^* < \text{PT: } 0.182^* < \text{iPRP: } 0.197^* < \text{MMR: } 0.205^*$$

To illuminate the differences in the re-ranking strategies, we focus on the kinematics of only the relevant documents. In particular, for each ranking approach, we recorded the change in the position of each relevant[9] document between the alternative ranking approach and the PRP. We thus count the number of times and the extent of the promotion or demotion of relevant documents with respect to the PRP. In Figure 1 we plot the distributions of the (relevant) document kinematics, where on the x-axis zero indicates no movement of documents, greater than zero indicates that the documents have been promoted, while lesser than zero indicates the documents have been demoted. The y-axis shows the frequency of the movement. To assess the symmetry of the kinematics shapes with respect to the zero-movement abscissa (i.e. the zero on the x-axis) we consider the area under the curve (AUC), that is given by the sum of the frequencies of promotions or demotions for a given approach. Specifically, we define as AUC left (AUCL) the sum of the frequencies for $x \in [-100, -1]$, while AUC right (AUCR) is defined as the sum of the frequencies for $x \in [+1, +100]$. We further extend the notion of AUC to a weighted version (WAUC) which weights each movement amplitude (each x value on the x-axis) by its frequency $f(x)$ and normalises this by the number of movements amplitudes different from zero contained in the considered movement range (note that for some values of x there is no movement). Formally, WAUC for a range $\mathcal{R}$ is defined as:

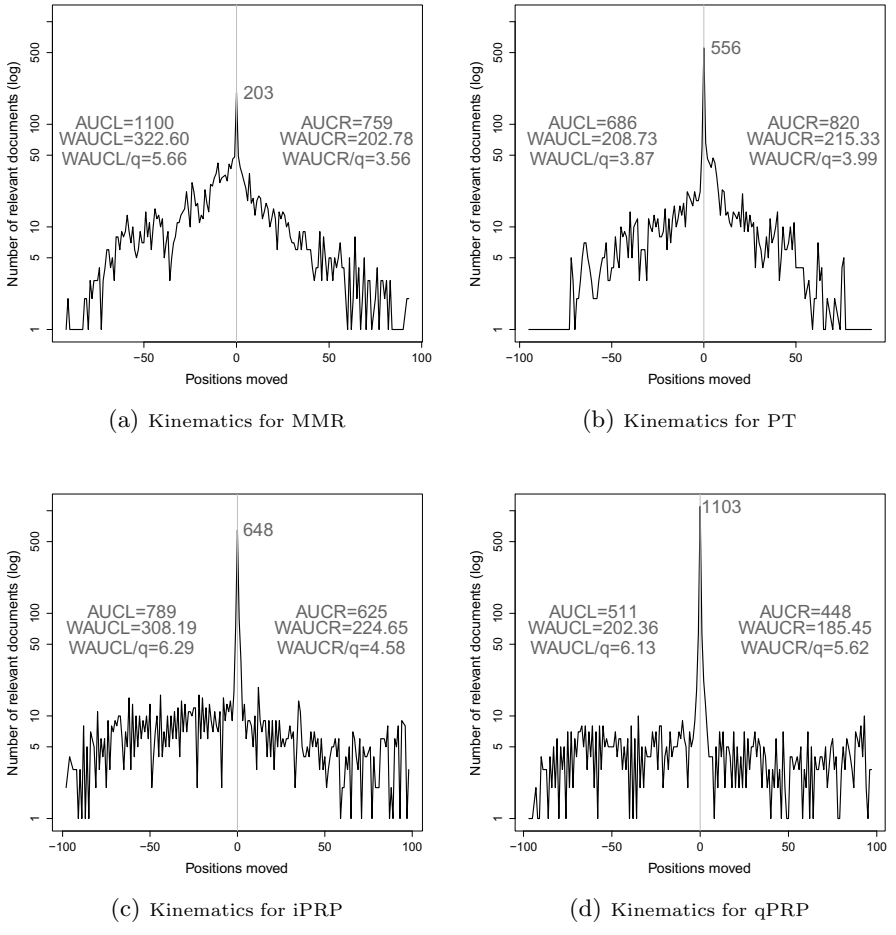
$$WAUC(R) = \frac{\sum_{x \in \mathcal{R}} |f(x) \cdot x|}{\sum_{x \in \mathcal{R}} \upsilon(x)} \text{ , where } \upsilon(x) = \begin{cases} 1 & \text{if } f(x) > 0 \\ 0 & \text{otherwise} \end{cases}$$

In particular, in the following we consider WAUCL for $x \in [-100, -1]$ (the area on the left of the zero-movement abscissa) and WAUCR for $x \in [+1, +100]$ (the area on the right of the zero-movement abscissa). Values of (W)AUCL and (W)AUCR for each approach are reported in Figure 1, together with the frequency of the zero-movement (i.e. $f(x = 0)$).

Retrieval strategies (i.e. PT and MMR, Figures 1(a) and 1(b)) are characterised by wider kinematics shapes that the ones of the principles (i.e. iPRP and qPRP, Figures 1(c) and 1(d)). MMR appears to be the approach that most revises the position of relevant documents, as it is characterised by the lowest frequency of zero-movements among all approaches. This might be mainly due to the fact that for 57 out of the 98 queries of the TREC 2009-2010 dataset the best performing value of the parameter λ is different from 1: that is, MMR's ranking

[8] Where * indicates statistical significant differences with respect to the PRP as measured by a two tailed paired t-test with $p \ll 0.01$. Note that no statistical significant differences were found between the performances of PT, MMR, iPRP and qPRP.

[9] We considered a document relevant if it is relevant to at least one facet/intent.



(a) Kinematics for MMR

(b) Kinematics for PT

(c) Kinematics for iPRP

(d) Kinematics for qPRP

Fig. 1. Kinematics, with respect to the PRP, imposed to the relevant documents by ranking strategies that cater for document dependencies. We also report the values of AUC, WAUC and the WAUC-to-query ratio (WAUC/q). Finally, in correspondence to $x = 0$, we report the frequency of zero-movements, i.e. $f(x = 0)$.

function effectively provides a ranking different than that of PRP, while for the remaining 41 queries MMR's ranking function reduces to PRP's one (since $\lambda = 1$ for these queries). The movement of relevant documents that is witnessed in Figure 1(a) is therefore generated by a high number of queries. While, movements that form the kinematics shapes of other approaches involve a lower number of queries. Specifically, the number of queries for which the best performing parameters do not reduce the ranking functions to that of PRP are 54 for PT, 49 for iPRP, 33 for qPRP.

The shape of MMR's kinematics is asymmetric and unbalanced towards the left side of the x-axis. The AUC of MMR confirms this impression: AUCL amounts to 1100, while the AUCR amounts to 759. This suggests that relevant documents are demoted more times than what are promoted. If compared to the kinematics shapes of other approaches, that of MMR can be regarded as being the most unbalanced towards the left side of the x-axis. Nevertheless, MMR achieves the highest value of α-NDCG@10 in our experiments: this might be because the relevant documents that are most demoted are those that are also most redundant, while the relevant documents that get promoted are novel with respect to the ones ranked at previous positions.

The shape of PT's kinematics is similar to the one of MMR's, although PT moves less relevant documents than MMR (higher zero-movement frequency) and its kinematics "ends" sooner than MMR's: no relevant documents are moved of more than 90 positions up or down the ranking. Furthermore, the kinematics of PT seems to favour the promotion of relevant documents over their demotion, as the kinematics shape is slightly unbalanced towards the right of the x-axis. This is confirmed by the difference between AUCR and AUCL; note that PT is the only approach for which AUCR $>$ AUCL. However, the difference between the area under the curve for the left and the right range decreases if WAUC is considered (i.e. WAUCL= 208.73, WAUCR= 215.33): this means that PT promotes relevant documents of fewer positions more than the ones it demotes.

The kinematics of the ranking principles (i.e. iPRP and qPRP) have a common shape. The kinematics are characterised by a high spike in correspondence of the zero-movement coordinate and a fast flattering out shape when movements involve more than half a dozen rank positions (note that the y-axis is in log-scale). The central spike represents no movement of relevant documents with respect to PRP: more relevant documents are moved by iPRP than qPRP. As for MMR, this observation is in line with the number of queries for which iPRP and qPRP provide a ranking different than PRP's one: this happens 49 times (out of 98 queries – i.e. for the 50% of the cases) for iPRP, while only 33 times for qPRP. For both principles the shapes are asymmetric and slightly unbalanced towards left (AUCL $>$ AUCR).

By comparing the WAUC of the approaches' kinematics, we can understand which strategy promotes or demotes relevant documents of more positions. Note however that a higher WAUC might not be due only to a propensity to promote or demote relevant documents of more positions, but might be as well biased by the number of queries that generated the kinematics. A better indication might be provided by the WAUC-to-query ratio (reported in Figure 1), where WAUC is divided by the number of queries for which there has been an effective movement of relevant documents with respect to the PRP. For example, while WAUCR of PT (215.33) is higher than the one of qPRP (185.45), WAUCR-to-query ratio of PT (3.99) is lower than the correspondent value for qPRP (5.62).

Notably, the lowest WAUC-to-query ratio is achieved by MMR with respect to documents that are promoted up the ranking (see WAUCR/q ratio of MMR), suggesting that overall MMR is the approach that less promotes relevant

documents. However, MMR is not the approach that most demotes relevant documents, as the WAUCL-to-query ratios of iPRP (6.29) and qPRP (6.13) are higher than that of MMR. The highest promotion of relevant documents is achieved by qPRP (WAUCR/q = 5.62): however this positive characteristic does not seem to find a parallel in the retrieval performances (at least in terms of α-NDCG@10). This might be due to the fact that (i) promoted relevant documents are redundant with respect to those ranked at previous positions, and/or (ii) promotions of relevant documents do not take place within the first 10 rank positions.

The previous analysis clearly shows how each ranking approach moves relevant documents within the ranking. As a further note, we can observe that if little movement transpires then the retrieval results are similar to the PRP, while more movement results in greater or lower performance.

5 Summary and Future Work

In this paper, we have investigated a number of ranking strategies and principles that have been proposed in the literature. Our analysis focused both on the analytical relationships between the approaches and on their ranking behaviours. We have shown the links that exist between ranking approaches. Moreover we have described the behaviours of the approaches when having to decide whether promote or demote a document given previously ranked evidence. Finally, we have examined the relevant document kinematics with respect to the PRP that the re-ranking approaches impose on the ranking: to the best of our knowledge, this is the first work that investigates this aspect of ranking approaches.

References

1. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998, pp. 335–336 (1998)
2. Chen, H., Karger, D.R.: Less is more: probabilistic models for retrieving fewer relevant documents. In: SIGIR 2006, pp. 429–436 (2006)
3. Fuhr, N.: A probability ranking principle for iir. JIR 12(3), 251–265 (2008)
4. Goffman, W.: On relevance as a measure. Info. Stor. & Ret. 2(3), 201–203 (1964)
5. Gordon, M.D., Lenk, P.: When is the prp suboptimal. JASIS 43(1), 1–14 (1999)
6. Robertson, S.E.: The probability ranking principle in IR. J. Doc. 33, 294–304 (1977)
7. Wang, J., Zhu, J.: Portfolio theory of information retrieval. In: SIGIR 2009, pp. 115–122 (2009)
8. Zuccon, G., Azzopardi, L.: Using the quantum probability ranking principle to rank interdependent 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. 357–369. Springer, Heidelberg (2010)
9. Zuccon, G., Azzopardi, L., Hauff, C., van Rijsbergen, C.J.: Estimating interference in the QPRP for subtopic retrieval. In: SIGIR 2010, pp. 741–742 (2010)
10. Zuccon, G., Azzopardi, L., van Rijsbergen, C.J.: The interactive PRP for diversifying document rankings. In: SIGIR (to appear, 2011)

Towards a Better Understanding of the Relationship between Probabilistic Models in IR

Robin Aly[1] and Thomas Demeester[2]

[1] University of Twente
r.aly@ewi.utwente.nl
[2] Ghent University
thomas.demeester@intec.ugent.be

Abstract. Probability of relevance (PR) models are generally assumed to implement the Probability Ranking Principle (PRP) of IR, and recent publications claim that PR models and language models are similar. However, a careful analysis reveals two gaps in the chain of reasoning behind this statement. First, the PRP considers the relevance of particular documents, whereas PR models consider the relevance of any query-document pair. Second, unlike PR models, language models consider draws of terms and documents. We bridge the first gap by showing how the probability measure of PR models can be used to define the probabilistic model of the PRP. Furthermore, we argue that given the differences between PR models and language models, the second gap cannot be bridged at the probabilistic model level. We instead define a new PR model based on logistic regression, which has a similar score function to the one of the query likelihood model. The performance of both models is strongly correlated, hence providing a bridge for the second gap at the functional and ranking level. Understanding language models in relation with logistic regression models opens ample new research directions which we propose as future work.

1 Introduction

The Probability Ranking Principle (PRP) of IR [10] is one of the widest acknowledged ranking principles in IR, and the fact that probability of relevance (PR) models [13] implement the PRP is commonly accepted without arguing [1]. Furthermore, to explain the empirically strong performance of language models, recent publications reason that language models are similar to PR models and therefore also implement the PRP [5, 14]. We identify two gaps in this chain of reasoning: (Gap1) The PRP considers the relevance of particular documents, which cannot be directly related to the relevance of query-document pairs considered by the PR models, and (Gap2) the relevance of query-document pairs cannot be directly related to the term and document draws considered by language models. In this paper, we investigate the above mentioned gaps and examine how they can be bridged. Figure 1 shows an overview of the content of this paper.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 164–175, 2011.

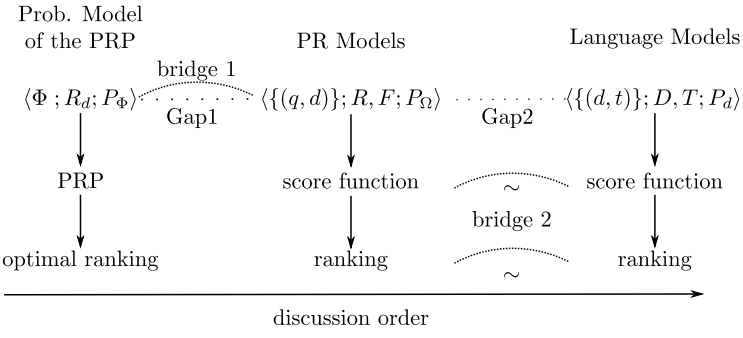


Fig. 1. Graphical overview over this paper's contents. Gap1's bridge translates models. Gap2's bridge relates score functions and rankings. The notation $\langle X; Y; Z \rangle$ denotes a probabilistic model where X are samples, Y are events, and Z is a probability measure. The detailed definition of the symbols used in this figure will be given in further sections.

The PRP shows that ranking a document d by the probability of its relevance, for example, maximizes the expected precision of a ranking. On the other hand, PR models rank by the probability of any query-document pair (q, d) being relevant given the pair has certain features F, see Sect. 3.2. Therefore, Gap1 is the difference among the considered relevance events and among their probabilities. We argue that Gap1 has so far gone unnoticed because the probabilistic model considered by the PRP has not been defined on a mathematical basis yet. To bridge Gap1, we define the PRP's *probabilistic model*, and show how PR models can be related to this definition.

Language models consider variations of drawing terms and documents as samples. First, the *query likelihood model* [9] considers drawing query terms, second, *Hiemstra's model* [3] additionally considers drawing documents, and finally, the *risk-minimization model* [20] as well as the *relevance language model* [6] consider drawing a single term. The difference between the drawing of query-document *pairs* in PR models and the drawing of terms and documents in language models forms Gap2, whose existence is controversially discussed in literature [16, 11, 7, 18]. Similar to [11], we argue that this controversy originates from the fact that the concept of sample spaces in language models has received little attention so far. Therefore, we first define the sample spaces of the above language models on a mathematical basis. Given these definitions, we claim that PR models and the above language models are too dissimilar for Gap2 to be bridged at the probabilistic model level.

If Gap2 cannot be bridged at the probabilistic model level, it is interesting to investigate to what extent language models are related to PR model in terms of score functions and rankings. Roelleke and Wang [14] are the first to find a relation on an analytical level between the score functions of the Binary Independence Model (BIM) [12] and Hiemstra's Language model. However, this relation only holds for documents with the same term occurrences (apparent from

Theorem 2 in [14]). To overcome this limitation, we define a new PR model based on logistic regression, the score function of which is similar to the score functions of the query likelihood model in terms of structure, weights, and ranking results. Although we are not able to bridge Gap2 at the probabilistic model level, we can therefore bridge Gap2 at the functional and ranking level.

This paper is structured as follows: Section 2 introduces the notation and basic definitions. Section 3 describes Gap1 and the probabilistic model we propose for the PRP to bridge it. Section 4 discusses Gap2, and why we cannot bridge it at the probabilistic model level. Section 5 defines a new PR model which ranks similarly to language model score functions and bridges Gap2 at the functional and ranking level. Finally, Section 6 concludes the paper.

2 Notation and Definitions

In this section, we introduce basic notations and central concepts from information retrieval and probability theory.

We denote queries and documents by lower case q's and d's, respectively. The considered set of queries is denoted by $\mathcal{Q}$ and the considered set of documents (the collection) by $\mathcal{D}$. Lower case t's are used for terms, and $\mathcal{T}$ indicates the considered set of terms (the vocabulary). The query terms of a query are modeled as the vector $\boldsymbol{qt} = (qt_1, ..., qt_{ql})$ where ql is the query length. Furthermore, the random variable R, relevance, is defined as

$$R(q,d) = \begin{cases} 1 & \text{if document } d \text{ is relevant to query } q, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Note that a query on its own should not be confounded with its properties. For example, the reader may think of a query as an object in an object-oriented programming language, the symbol q as a reference, and the query terms $\boldsymbol{qt}$ as some of the object's properties. The same holds for documents.

Following Manning and Schuetze [8, chap. 2], we define the basic concepts of probability theory as follows: a *sample* is a possible outcome of a process. The corresponding *sample space* is the set of all possible samples. An *event* is a subset of the sample space. An *event space* is a set of events. A *probability measure* is a function which maps events to probabilities. We use a subscript to the probability measure P to indicate the process on which the measure is defined, for example $P_X : \mathcal{E} \rightarrow [0 : 1]$ is a probability measure defined on the event space $\mathcal{E}$ for process X. A *random variable* is a function mapping samples to the function's range. Note that a value of a random variable defines an event: the subset of samples in the sample space for which the random variable yields this value.

3 Gap1 – Between the PRP and PR Models

In this section, we bridge Gap1, the difference between the PRP and PR models. First, we describe the PRP and the unified framework of PR models. After that, we show a way to relate the two probabilistic models.

3.1 The PRP

In the following we sketch the PRP and propose a definition for the underlying sample space and events, which has not yet, on a mathematical basis, been proposed in literature. Note that the proposed sample space is not necessarily the one Robertson [10] had in mind, however we consider it likely that this is indeed the case.

For a given query, the PRP considers the expected precision (together with the expected recall and reading costs) for a reader who stops reading at any rank n. This expected precision can be defined as follows:

$$E[Prec^n_{\boldsymbol{d}}] = \frac{1}{n} \sum_{j=1}^{n} P_{\Phi}(R_{d_j}{=}1)$$

Here, $\boldsymbol{d}$ is a ranking of documents which is read until rank n. The PRP then shows that a ranking of documents

$$(d_1, ..., d_{|\mathcal{D}|}) \text{ for which } P_{\Phi}\left(R_{d_1}\right) \geq ... \geq P_{\Phi}\left(R_{d_{|\mathcal{D}|}}\right), \tag{2}$$

maximizes the expected precision for *any* rank n:

$$(d_1, ..., d_{|\mathcal{D}|}) = \operatorname*{argmax}_{\boldsymbol{d}} E[Prec^n_{\boldsymbol{d}}] \tag{3}$$

Here, $\boldsymbol{d}$ varies over all possible rankings of the documents in the collection $\mathcal{D}$ and each document d can either be labeled relevant $R_d = 1$, or irrelevant $R_d = 0$. Therefore, we propose that the PRP's sample space Φ consists of all possible relevance labeling combinations of the documents in the collection, for the current query:

$$\Phi = \underbrace{\{0,1\} \times ... \times \{0,1\}}_{|\mathcal{D}| \text{ times}}$$

Here, each component corresponds to a document in the collection. For a particular sample (a specific relevance labeling of all documents) $\phi \in \Phi$, we denote the relevance label of document d as ϕ_d, and we define a (trivial) relevance random variable for each document $d \in \mathcal{D}$ as the relevance of that document within the sample, shortly, $R_d(\phi \in \Phi) = \phi_d$. The event $R_d{=}1$ is the set of all samples ϕ with $R_d(\phi){=}1$. The sample space Φ requires a Bayesian perspective on probabilities, because in a Frequentist's perspective a document can never be relevant or irrelevant to the same query, according to our assumptions in Sect. 2. As a result, the probability measure $P_{\Phi}(R_d{=}1)$ expresses the degree of belief that document d is relevant. A retrieval model has to define these probabilities for each document $d \in \mathcal{D}$ in order to implement the PRP.

3.2 PR Models

Robertson et al. [13] propose a unified framework of PR models which rank by the probability that any query-document pair from the sample space $\Omega = \mathcal{Q} \times \mathcal{D}$

is relevant[1]. The unified framework of PR models comprises four (meta-) models (Model 1–4), which consider variations to partition the sample space Ω by abstract query features and document features (or random variables)[2].

$$\boldsymbol{QF} = (QF_1, ..., QF_m) \tag{4}$$

$$\boldsymbol{QF}(q) = (QF_1(q), ..., QF_m(q)) \tag{5}$$

$$\boldsymbol{F} = (F_1, ..., F_n) \tag{6}$$

$$\boldsymbol{F}(d) = (F_1(d), ..., F_n(d)) \tag{7}$$

Here, QF_i is a query feature (a function of $q \in \mathcal{Q}$), $\boldsymbol{QF}$ is a vector of m considered query features, and $\boldsymbol{QF}(q)$ are the query features of query q. Furthermore, F_i is a document feature (a function of $d \in \mathcal{D}$), $\boldsymbol{F}$ is the vector of n document features, and $\boldsymbol{F}(d)$ are the document features of document d. For example, a query feature could be "query q contains term t", defined as $W_t : \mathcal{Q} \rightarrow \{0, 1\}$. The sets of considered features $\boldsymbol{QF}$ and $\boldsymbol{F}$ are usually selected by considering the query terms $\boldsymbol{qt}$ or terms from query expansion [2]. For later use, we introduce two trivial features: let $Q(q) = q$ be the query of a query document pair, and let $D(d) = d$ be the document of the query-document pair.

Because of space limitations, we focus our discussion to the BIM, an instance of Model 2. The BIM considers ql indexing document features, $I_i : \mathcal{D} \rightarrow \{0, 1\}$, indicating whether or not a document is indexed with query term qt_i. Documents are then ranked by the probability that any query-document pair is relevant, which we display for instructive reasons from a Frequentist's perspective, similar to [13]:

$$P_\Omega(R \mid Q(q){=}q^*, \boldsymbol{F}{=}\boldsymbol{F}(d^*)) = \frac{|\{(q,d) \in \Omega \mid R(q,d){=}1, Q(q){=}q^*, \boldsymbol{F}(d){=}\boldsymbol{F}(d^*)\}|}{|\{(q,d) \in \Omega \mid Q(q){=}q^*, \boldsymbol{F}(d){=}\boldsymbol{F}(d^*)\}|} \tag{8}$$

Here, q^* is the current query, and d^* is the current document. Now, Gap1 exists between the probabilistic model of the PRP, which considers relevance of particular documents to particular queries, and PR models which consider the relevance of any query-document pairs.

3.3 A Bridge for Gap1

In this section, we bridge Gap1 by showing how PR models can be used in the definition of the probability measure used by the PRP. Considering Model 2, if we assume that the only knowledge we have about documents are their features $\boldsymbol{F}$,

[1] Note that Robertson [11] refers to Ω as an event space. However, Ω is a set of pairs whereas an event space is a set of sets according to our definitions in Sect. 2. We assume Ω to be a sample space.

[2] In PR model literature, document features are also referred to as document representations, and descriptors, and they are often denoted by D. We denote features by $\boldsymbol{F}$ to avoid confusion with a document d.

we can decide to treat documents with the same features as indistinguishable. Under this assumption, it is reasonable to define the degree of belief $P_\Phi(R_d)$ that document d is relevant, as the probability that a document of any random query-document pair is relevant, given that the query is the current query and the document has the same features $\boldsymbol{F}(d)$ as the current document d:

$$P_\Phi(R_d) = P_\Omega(R|Q{=}q^*, \boldsymbol{F}{=}\boldsymbol{F}(d)) \tag{9}$$

Because of this equality of the two probability measures, PR models which rank by the probability $P_\Omega(R|Q{=}q^*, \boldsymbol{F}{=}\boldsymbol{F}(d))$ produce the same ranking as the PRP, see Equation 2. Therefore, Equation 9 bridges Gap1 between the PRP and PR models derived from Model 2. Note that Fuhr [2] discusses the influence of the chosen features $\boldsymbol{F}$ on the probability $P_\Omega(R|Q{=}q^*, \boldsymbol{F}{=}\boldsymbol{F}(d))$. However, although the choice of $\boldsymbol{F}$ influences the strength of the bridge (the more selective the features, the more realistic the assumption in Equation 9), this did not lead to the discovery of or answer to Gap1.

Furthermore, for example, Model 1 of the unified framework of PR models ranks a document d by the probability $P_\Omega(R|\boldsymbol{QF}{=}\boldsymbol{QF}(q^*), D{=}d)$, where $\boldsymbol{QF}$ are query features. Therefore, for each document d, Model 1 considers different queries with the same features. It is however less intuitive, why this probability would express our degree of belief $P_\Phi(R_d)$ that document d would be relevant to the *current* query. We postpone the investigation of this issue to future work.

4 Gap2 – Between PR Models and Language Models

In this section, we analyze Gap2, the difference between PR models and language models. First, we define the corresponding probabilistic model for four popular language models and then point out the differences to PR models described in Sect. 3.2.

4.1 Language Models

Language models have in common that they consider draws of terms, for which we define the (partial) sample space and the considered random variables:

$$\mathcal{T}_n = \overbrace{\mathcal{T} \times ... \times \mathcal{T}}^{n \text{ times}} \tag{10}$$

$$T_i(\boldsymbol{t} \in \mathcal{T}_n) = \text{the } i\text{th term in } \boldsymbol{t} \tag{11}$$

$$\boldsymbol{T}(\boldsymbol{t} \in \mathcal{T}_n) = \boldsymbol{t} \tag{12}$$

Here, $\mathcal{T}_n$ is the (partial) sample space of drawing n terms (the set of all possible term combinations resulting from n term draws), the random variable T_i states the ith term, and $\boldsymbol{T}$ does the same for sequences of term draws. Furthermore, in a uni-gram model, to which we limit the discussion, the random variable T_i represents the results of the ith independent trial from a multinomial distribution,

and we have $P_d(\boldsymbol{T}{=}\boldsymbol{qt}) = \prod_{i=1}^{ql} P_d(T_i{=}qt_i) = \prod_{i=1}^{ql} \theta_{d,qt_i}$. Here, $P_d(T{=}t)$ is the probability of drawing the term t, and θ_{d,qt_i} is the distribution parameter of the term qt_i in language model of document d. To show that the language model parameters $\boldsymbol{\theta}_d$ are estimations, they are sometimes included in the notation of Bayesian probabilities, $P_d(T{=}t) = P_d(T{=}t|\boldsymbol{\theta}_d)$. Here, we focus on the probabilistic model used for ranking and consider the language model parameters as fixed.

For a given query, the *query likelihood model* [9] considers for each document in the collection the drawing of ql random terms[3]. Documents are ranked by the probability that the query terms are drawn, $P_d(\boldsymbol{T}{=}\boldsymbol{qt})$.

Hiemstra's model [3] considers documents, which the user has in mind for a query, and terms which the user drew from the language model of this document:

$$\mathcal{H} = \mathcal{D} \times \mathcal{T}_{ql}$$
$$D'((d, \boldsymbol{t}) \in \mathcal{H}) = \text{the document } d \text{ which the user had in mind}$$

Here, $\mathcal{H}$ is the sample space of Hiemstra's model, D' is the random variable stating which document the user had in mind, and $\boldsymbol{t}$ are the drawn terms, see Equation 11. Hiemstra's model ranks a document by the probability that the user had document d in mind given the observed query terms, $P_{\mathcal{H}}(D'{=}d|\boldsymbol{T}{=}\boldsymbol{qt})$.

The *risk-minimization model* [20] considers the process of drawing a single term (sample space $\mathcal{T}_1$) from a query language model and from the language model of each document. Documents are ranked by the Kullback-Leibner divergence between the distribution of the query language model and the document's language model.

$$KL(P_q||P_d) = \sum_{t \in \mathcal{T}} P_q(T{=}t) \ \log\left(\frac{P_q(T{=}t)}{P_d(T{=}t)}\right)$$

Here, P_q is the probability measure of the query language model. Note that it is rarely mentioned in literature that the risk-minimization framework only considers a single term draw. However, this must be the case because if the Kullback-Leibner divergence were considered for, say, ql term draws, the above summation would run over $|\mathcal{T}|^{ql}$ possible outcomes of the draws.

The *relevance language model* [6] considers for each document the process of drawing a single term from this document. The distribution is compared with a relevance model of the current query which considers the sample space of first drawing a relevant document and subsequently a term from this document. The sample space and the random variable for the drawn document of the relevance model are defined as follows:

$$\mathcal{RM} = \{(d, t) \in \mathcal{D} \times \mathcal{T}_1 | R(q^*, d){=}1\}$$
$$D''((d, t) \in \mathcal{RM}) = d \text{ was drawn}$$

[3] Following common usage, we interpret the query likelihood model as multinomial trials; it leads to the same ranking as the multi-Bernoulli interpretation considered in [9].

Here, $\mathcal{RM}$ is the sample space of the relevance model (a document-term pair), q^* is the current query, D' states the drawn relevant document. The relevance language model ranks by the negative cross entropy between drawing a term from the relevance model and from the document language model $-CE(P_r||P_d)$ of drawing terms. Here, the probability of drawing a term from the relevance model is determined by marginalization over D''.

4.2 Differences between PR Models and Language Models

Based on the definitions of PR models in Sect. 3.2 and the presented language models in the previous Sect. 4.1, we investigate whether we can bridge Gap2 at the probabilistic model level. To compare PR models and the presented language models, they are usually presented as different derivations of the probability $P(R|Q, D)$ [5, 7, 15]. However, the definition of each of these symbols differs among PR models and language models. In PR models, Q are query features, denoted in this paper as $\boldsymbol{QF}$, which are functions of the considered query. Therefore, given a query, its feature values are not random. On the other hand in the presented language models, the random variable Q, which is in our notation $\boldsymbol{T}_{ql}$, represents the outcome of randomly drawing ql terms and this does not depend on a query.

Furthermore, in PR models, D stands for document features, denoted in this paper as $\boldsymbol{F}$, which are functions of the considered document. Therefore, given a particular document the feature value is not random. On the other hand, in Hiemstra's model, D stands for the document the user had in mind and which is modeled as the outcome of a random process.

Also, the notion of relevance differs between its use in PR models, where it is a function of query-document pairs, and its use in the four presented language models. First, the query likelihood model and the risk-minimization model do not use the notion of relevance. Second, Hiemstra's model assumes only a single relevant document [16]. Finally, the notion of relevance in the relevance language model can be seen to be the same as in PR models. However, in the relevance language model, single, particular documents are drawn from the relevance model while PR models consider drawing any relevant query-document pair with certain features $P_\Omega(\boldsymbol{F}|R)$.

Therefore, we propose that the reasoning for the similarity between PR models and the presented language models is mainly guided by similar notation, and that PR models and the presented language models are too different to bridge Gap2 at the probabilistic model level.

5 Bridging Gap2 at the Functional and Ranking Level

In this section, we propose a new PR model which ranks similarly as the score function of the query likelihood model. Instead of considering probabilities of drawing a term, $P_d(T = t)$, we consider language scores as document features (functions of a document), a particular feature F in Equation 6:

$$LS_i(d) = \log\left(\frac{\theta_{d,qt_i}}{\alpha_d\ \theta_{\mathcal{D},qt_i}}\right) \tag{13}$$

Here, $LS_i(d)$ is the language score of document d for query term qt_i, θ_{d,qt_i} is the language model parameter for query term qt_i in document d, see Sect. 4.1, α_d ensures that $LS_i(d)$ is zero if query term qt_i is not in the document [19], and $\theta_{\mathcal{D},qt_i}$ is the constant collection prior. We denote the vector of language score feature functions for the current query as $\boldsymbol{LS} = (LS_1, ..., LS_{ql})$ and, evaluated for a document d, as $\boldsymbol{LS}(d) = (LS_1(d), ..., LS_{ql}(d))$. Based on these features, we define a PR model in which the probability of any query-document pair being relevant is represented by a discriminative logistic regression model [4]:

$$P_\Omega(R\,|Q{=}q^*, \boldsymbol{LS}{=}\boldsymbol{LS}(d^*)) = \frac{1}{1 + exp(-w_0 - \sum_{i=1}^{ql} w_i\ LS_i(d^*))} \propto \sum_{i=1}^{ql} w_i\ LS_i(d^*) \tag{14}$$

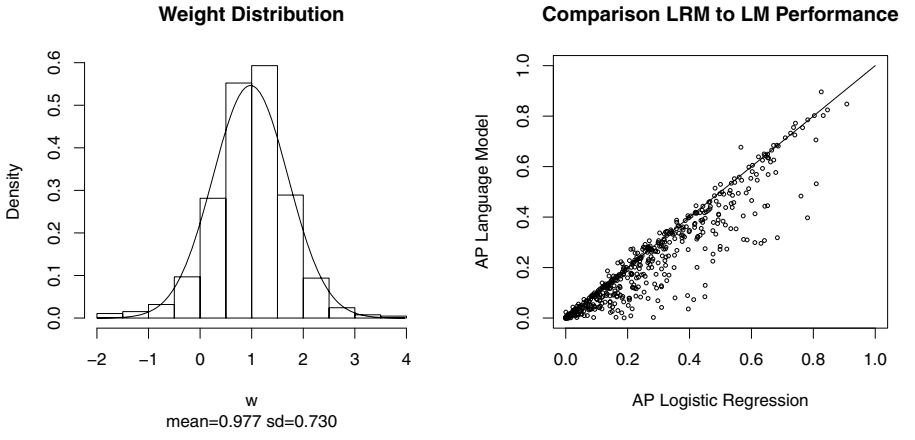
Here, q^* is the current query, $\boldsymbol{LS}(d^*)$ are the language scores of the current document d^*, w_0 is the intercept of the logistic regression model representing the relevance prior, and w_i is the language score weight of query term qt_i. From the calculated probability $P_\Omega(R\,|Q{=}q^*, \boldsymbol{LS}{=}\boldsymbol{LS}(d^*))$ we see that the PR model implements Model 2 of the unified framework of PR models, for which we have shown that it bridges Gap1. The middle term of Equation 14 is the definition of the logistic regression model, and the rightmost term is a rank equivalent score function see [17] for a derivation.

Now, we compare the logistic regression model in Equation 14 with the score function of the query likelihood model, which is defined as follows [19]:

$$P_d(\boldsymbol{T}{=}\boldsymbol{qt}) \propto \sum_{i=1}^{ql} \log\left(\frac{P_d(T{=}qt_i)}{\alpha_d\ \theta_{\mathcal{D},qt_i}}\right) + |\mathcal{T}|\ \alpha_d + const$$

Here, α_d has the same function as in Equation 13. From expanding the rightmost term of Equation 14 by the definition of language scores in Equation 13 and using the relationship $P_d(T{=}qt_i) = \theta_{d,qt_i}$, we see that the logistic regression model score function has a similar structure to the score function of the query likelihood model, except for the missing expression $|\mathcal{T}|\ \alpha_d$ and the non-uniform language score weights w_i.

In order to quantify their similarity in practice, we compare the performance of the score functions of the logistic regression models and the query likelihood model using 550 queries from the ROBUST 04+05, TREC 09, TERABYTE 04-06 data sets. If we assume uniform language score weights w_i in the logistic regression model, the model practically performs identically to the query likelihood model in terms of mean average precision (MAP). Therefore, the term $|\mathcal{T}|\ \alpha_d$ has no significant influence on the ranking. Furthermore, we consider the hypothetical case of using the language score weights w_i which we trained on all relevance data for each query separately. Figure 2 (a) shows that the trained language score weights w_i are Gaussian distributed with an expected weight around



(a) (b)

Fig. 2. (a) Weight distribution over the query terms of 550 queries of the proposed logistic regression model (LRM) trained on all relevance data. (b) Performance comparison to the query likelihood model (LM).

one. Therefore, for a random query term we can *expect* the logistic regression weight w_i of the corresponding language score to be one. This expected weight also coincides with the uniform weight of the query likelihood model. Figure 2 (b) compares the performance of the hypothetical logistic regression model against the performance of the query likelihood model in terms of average precision. The models have a high performance correlation (Pearson correlation coefficient 0.92). We suggest that the uniform weights of the query likelihood score function can also be seen as a first approximation of the ideal language score weights w_i from Equation 8. As a result, the newly proposed logistic regression PR model bridges the Gap2 to the query likelihood model at a functional and ranking level.

Additionally, the score functions of the risk-minimization framework and the relevance model can be seen as methods to improve upon the uniform weights of the query likelihood model for an expanded set of query terms [18]. Hence, these weights potentially could also be approximations to the weights w_i of newly selected features. We postpone these investigations to future work.

Note that the similarity of the *score functions* of the described logistic regression model and the query likelihood model does not imply that ranking by the query likelihood *model* could not be justified otherwise.

6 Conclusions

In this paper, we bridged two gaps in the chain of reasoning used for two popular probabilistic IR models, PR models and language models. (Gap1) The PRP considers the relevance of particular documents which cannot directly be related

to the relevance of query-document pairs considered by the PR models, and (Gap2) the relevance of query-document pairs cannot directly be related to the term draws considered by language models.

In order to bridge Gap1, we defined a probabilistic model underlying the PRP, which considers all possible combinations of relevance labels of the documents in the collection. Probabilistic models which implement the PRP need to define the degree of belief that document d is relevant $P_\Phi(R_d)$. Furthermore, the (meta) Model 2 of the unified framework of PR models [13] considers the probability of relevance of any query-document pair with the query being the current query q^* and the document having the same features $\boldsymbol{F}(d)$ as the current document d^*, $P_\Omega(R|Q{=}q, \boldsymbol{F}{=}\boldsymbol{F}(d^*)$. We argued that, under the assumption that we can only distinguish documents by the features $\boldsymbol{F}$, we can take $P_\Omega(R|Q{=}q, \boldsymbol{F}{=}\boldsymbol{F}(d^*))$ as the degree of belief of relevance $P_\Phi(R_d)$. With this assumption, Gap1 was bridged. Similar assumptions of the other models of the unified framework require further investigations, which we will discuss in future work.

Furthermore, from the definition of the probabilistic model of PR models and language models, we found that the two models are different and we observed that previous comparisons were mainly based on similar notation with different meaning. Therefore, Gap2 could not be bridged at the probabilistic model level. Additionally, we proposed a new PR model derived from Model 2 of the unified framework of PR models, based on logistic regression. For 550 queries in six collections, we showed that the score functions of the logistic regression model and the query likelihood model were similar, and the performance of the two score functions was strongly correlated. Comparing the weights of both score functions showed that the uniform weights of the query likelihood model score function can be seen as the expected logistic regression weights for a random query. Therefore, we bridged Gap2 at the functional and ranking level, leading to an alternative explanation for the strong performance of language models.

Understanding and further exploring the apparent connection between language models and logistic regression models (or possibly other discriminative models) opens ample new research directions which we propose for future work. The proposed logistic regression model could for instance be used for score normalization, and existing research on feature selection for logistic regression models could be used for query expansion.

References

[1] Crestani, F., Lalmas, M., Rijsbergen, C.J.V., Campbell, I.: Is this document relevant?.probably: a survey of probabilistic models in information retrieval. ACM Comput. Surv. 30(4), 528–552 (1998) ISSN: 0360-0300

[2] Fuhr, N.: Probabilistic models in information retrieval. Comput. J. 35(3), 243–255 (1992)

[3] Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, University of Twente, Enschede (January 2001)

[4] Hosmer, D.W., Lemeshow, S.: Applied logistic regression. Wiley-Interscience Publication, Hoboken (September 2000) ISBN 0471356328

[5] Lafferty, J., Zhai, C.: Probabilistic Relevance Models Based on Document and Query Generation, ch. 1, pp. 1–10. Kluwer Academic Pub., Dordrecht (2003)
[6] Lavrenko, V., Croft, W.B.: Relevance models in information retrieval. In: Language Modeling for Information Retrieval, pp. 11–56. Kluwer Academic Publishers, Dordrecht (2003)
[7] Luk, R.W.P.: On event space and rank equivalence between probabilistic retrieval models. Information Retrieval 11(6), 539–561 (2008), ISSN 1386-4564 (Print) 1573-7659 (Online), doi:10.1007/s10791-008-9062-z
[8] Manning, C.D., Schuetze, H.: Foundations of Statistical Natural Language Processing, 1st edn. The MIT Press, Cambridge (June 1999) ISBN 0-26213-360-1
[9] Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR 1998, pp. 275–281. ACM, New York (1998) ISBN 1-58113-015-5, doi:10.1145/290941.291008
[10] Robertson, S.E.: The probability ranking principle in IR. Journal of Documentation 33, 294–304 (1977)
[11] Robertson, S.E.: On event spaces and probabilistic models in information retrieval. Information Retrieval 8(2), 319–329 (2005) ISSN 1386-4564 (Print) 1573-7659 (Online), doi:10.1007/s10791-005-5665-9
[12] Robertson, S.E., Spärck-Jones, K.: Relevance weighting of search terms. Journal of the American Society for Information Science 27(3), 129–146 (1976), doi:10.1002/asi.4630270302
[13] Robertson, S.E., Maron, M.E., Cooper, W.S.: Probability of relevance: A unification of two competing models for document retrieval. Information Technology: Research and Development 1(1), 1–21 (1982)
[14] Roelleke, T., Wang, J.: A parallel derivation of probabilistic information retrieval models. In: SIGIR 2006, pp. 107–114. ACM, New York (2006) ISBN 1-59593-369-7, doi:10.1145/1148170.1148192
[15] Roelleke, T., Wang, J.: Tf-idf uncovered: a study of theories and probabilities. In: SIGIR 2008, pp. 435–442. ACM, New York (2008) ISBN 978-1-60558-164-4, doi:10.1145/1390334.1390409
[16] Spärck-Jones, K., Robertson, S.E., Zaragoza, H., Hiemstra, D.: Language modelling and relevance. In: Language Modelling for Information Retrieval, pp. 57–71. Kluwer, Dordrecht (2003)
[17] Yan, R.: Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval. PhD thesis, Canegie Mellon University (2006)
[18] Zhai, C.: Statistical language models for information retrieval a critical review. Found. Trends Inf. Retr. 2(3), 137–213 (2008) ISSN 1554-0669, doi:10.1561/1500000008
[19] Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004) ISSN 1046-8188, doi:10.1145/984321.984322
[20] Zhai, C., Lafferty, J.: A risk minimization framework for information retrieval. Inf. Process. Manage. 42(1), 31–55 (2006) ISSN 0306-4573, doi:10.1016/j.ipm.2004.11.003

Cognitive Processes in Query Generation

Claudia Hauff* and Geert-Jan Houben

WIS, Delft University of Technology, Delft, the Netherlands
{c.hauff,g.j.p.m.houben}@tudelft.nl

Abstract. *Known-item* search is the search for a specific document that is known to exist. This task is particularly important in Personal Information Management (PIM), where it is the most common search activity. A major obstacle to research in search technologies for PIM is the lack of publicly accessible test corpora. As a potential solution, pseudo-desktop corpora and automatic query generation have been proposed. These approaches though do not take the cognitive processes into account that take place when a user formulates a re-finding query. The human memory is not perfect, and many factors influence a user's ability to recall information. In this work, we propose a model that accounts for these cognitive processes in the automatic query generation setting.

1 Introduction

A vital component of research in information retrieval is the testing of research ideas on realistic test collections. Creating such test collections is both time-consuming and cost-intensive. For this reason, several initiatives, such as TREC[1] and CLEF[2], have been set up over the years. They provide researchers with standardized test corpora and retrieval tasks.

While we now have access to, among others, newspaper and Web corpora, test corpora for Personal Information Management (PIM) research are still lacking due to privacy concerns. PIM is concerned with the acquisition, storage, organization and the retrieval (re-finding) of information collected by a user. Due to the ever increasing reliance on digital communication channels and functions (email, chat, etc.) as well as digitally available information, the amount of data a user stores is growing continuously. A stored item can be, for instance, an email in the user's inbox, a scientific paper the user downloaded from the Web, or a calendar entry. Re-finding an item that the user has accessed before, a process known as *known-item* retrieval, is the most common search activity in PIM. Note, that known-item retrieval is also a usage scenario of Web search engines, which users may rely on to re-find previously visited web pages [1].

Research in PIM related search technologies is hampered significantly by the lack of public test corpora. To alleviate this problem, automatic [2,16] and human

* This research has received funding from the European Union Seventh Framework Programme (FP7/2007-2013), grant agreement no ICT 257831 (ImREAL project).

[1] Text REtrieval Conference http://trec.nist.gov/

[2] Cross Language Evaluation Forum http://www.clef-campaign.org/

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 176–187, 2011.

computation game [17] based topic set generation approaches have been proposed in the past. Given a test corpus, that resembles a generic user's personal or work Desktop, a document of the test corpus is selected as the "known item" for which a query is created. The automatic approaches construct topics by selecting terms of the document in question according to particular rules; for example, the most discriminative terms are selected with a higher probability or randomly (noise). In the human computation game scenario, the document in question is shown to human study participants who create queries with the goal to return the item as high in the retrieved ranking as possible. Note, that the human participants are actually shown the document, they do not need to remember it.

These two known-item topic creation approaches assume either (i) a perfect human memory where users remember the document's content fully and correctly and it is only a matter of selecting the "right" keywords to create a good query (in the human computation game approach), or, (ii) a human memory that fails randomly (in the automatic query generation approach). Human memory is neither perfect nor failing randomly, however. Indeed, research into so-called *false memories* is an important field of study in psychology where it is often motivated by the question of eyewitness reliability [7,21] and the correct recall of childhood experiences [12]. In this paper, we argue that for known-item retrieval to be more realistic, topic generation approaches need to take into consideration the imperfection of human memory and the tendency to create false memories. A similar argument was already made by Lansdale [19] who believed that the cognitive abilities of users need to be taken into account in the design of PIM tools. This argument is also supported by user studies in PIM, which have shown that users recall different aspects of their stored documents to different degrees [11]. Based on these findings, we propose a query generation model that includes false memories in order to generate more realistic queries.

If the imperfections of human memory are not reflected in a PIM test corpus, developing new search algorithms based on perfect memory queries or randomly failing memories may lead to false estimates of the algorithms' abilities. For instance, the TREC Enterprise track 2005 [8] contained a known-item task where the best systems retrieved the known item within the top ten ranks for more than 80% of all queries, which implies very well-performing known-item retrieval algorithms Some of the known-items in question, though, where ten year old emails (at the time of topic creation), which are unlikely to be remembered correctly in a realistic search setting.

The main contributions of our work are (i) an argument for the inclusion of false memories into test corpora for known-item tasks that is based on psychology research, (ii) a model for automatic query generation that includes a false memory component, and, (iii) an investigation into the TREC Enterprise track 2005 and the influences of false memories in it.

The rest of the paper is organized as follows: Sec. 2 describes research in false memories, both in psychology and PIM. Sec. 3 describes the inclusion of false memories into an existing query generation procedure. Experimental results are presented in Sec. 4, followed by the conclusions in Sec. 5.

2 Related Work

False Memories. A particular type of experiment, the Deese-Roediger-McDermott (DRM) paradigm [22], is widely used in psychology to study the effects of false memories (or memory illusions, memory distortions). A false memory is a person's recall of a past experience which differs considerably from the true course of events [23]. The DRM setup is as follows: given a critical word (e.g., *foot*) a list of no more than 15 semantically related words is created (e.g., *shoe*, *hand*, *toe*). Subjects first study the list of related terms (without the critical term), and are then asked to freely recall the terms in the list without resorting to guessing (this occurs immediately after having studied the list). Routinely, it is observed that the subjects recall the critical term, which is the elicited false memory, with a similar probability as the terms on the list. It is also notable, that study subjects are confident about having studied the critical term. One theoretical explanation for this observation has been provided by the *Source Monitoring Framework* [15,13] (SMF), which postulates that false memories are created because of confusions about a memory's source. A source can either be internal (thinking of *foot* while having heard the terms in the list) or external (the experimenter said *foot*).

According to the SMF, a memory's source is not directly encoded in memory, instead a number of memory characteristics are exploited in order to determine the source when retrieving a memory: sensory information (sound, color), contextual information (location, time), semantic detail, affective information (the emotional state), and evidence of cognitive operations (records of organizing the information). This means for instance, when a person recalls if he has read a statement in an email, heard it from a colleague, saw it during a presentation of a talk, or thought of it himself, attributing the source will depend on the person recalling the voice of the attributor, the color of the presentation, the time of reading the email or the thought process that lead to the statement. The amount of detail remembered for each memory characteristic determines which source the person finally attributes the statement to.

Source confusion or misattribution is deemed as the main cause of false memories. Source confusion occurs when the experience is poorly encoded into memory, for instance, if somebody reads an email while being distracted by a phone call or someone walking into his office. Later, a correct recall of the email content will be more difficult than if the person would have concentrated on just reading it. Stress, distractions and a strong emotional state [14] also degrade the encoding process. When retrieving from memory, these factors influence the ability to attribute the source correctly as well. Thus, false memory attributions can be based both on the encoding and the decoding phase. Moreover, if encoded memories have largely overlapping characteristics, source confusion is more likely; recalling the differences between the memories will be difficult, while remembering the general similarities, or the gist of the memories, is easier.

While SMF explains why subjects in the DRM experiments falsely recall the critical item (they confuse thinking and reading/hearing it), the activation and monitoring theory [22] explains why they think of the critical item in the first

place when hearing semantically related terms. When hearing the list terms, the memories of these terms are activated which in turn also leads to the activation of related memories (such as the critical item).

Another finding of memory research is that, the gist of a document, i.e., the meaning of the content, is longer retained in memory than specific details [24,18]. With respect to generating topics for known-item search this means, that we need to take the amount of time passed since the user last viewed the document, or more generally the access pattern of the document to be re-found, into account.

An additional factor to consider is age. It has been shown that older adults are more susceptible to false memories than younger adults [20,9].

If we translate those findings to PIM search tools, we can argue that a PIM search system should be adapted to each individual user and the context. For instance, a PIM search system can take the age of a user into account and treat queries posed by older users differently from queries posed by a younger adult. Similarly, if the PIM search system has an indication that the user is stressed or tired (an indication may be be derived from the user's activities on the system within the last hours), a posed query may be treated differently than a query posed by a calm and relaxed user.

Personal Information Management. Blanc et al. [5] describe the results of a user study, in which the ability to recall attributes of the users' own documents (both paper and digital ones) and their ability to re-find those documents in their work place was investigated. It was observed that the study participants when being asked to recall the title and keywords of the document in question were most often mixing true and false memories; for 32% of the documents the recalled keywords were correct, while for 68% they were only partially correct ("partial recall" in [5]). Recalling the title was more difficult: 33% correctly recalled document titles, vs. 47% partially correct and 20% completely false recollections.

Elsweiler et al. [11] performed a user study to investigate what users remember about their email messages and how they re-find them. The most frequently remembered attributes of emails were found to be the topic, the reason for sending the email, the sender of the email and other temporal information. No indication was given if the memories were (partially) false or correct. Another finding, in line with research in psychology, was that memory recall declines over time, that is, emails that had not been accessed for a long time were less likely to have attributes remembered than recently read emails. That users are indeed accessing old documents on their Desktop has been shown in [10], where up to eight year old documents were sought by users in a work environment.

In general it has been found across a range of studies, e.g., [3,6,5,4,25], that in PIM re-finding, users prefer to browse to the target folder and to visually inspect it in order to find the target document instead of relying on the provided Desktop search tools. It is argued that the current PIM search tools are not sophisticated enough to deal with what and how users remember aspects of the target documents. For this reason we propose the inclusion of false memories into generated known-item queries, to make the test corpora more realistic and more in line with true user queries.

3 Methodology

In this section, we will first introduce the two types of false memories that we distinguish, based on an information retrieval point of view. Then, the automatic topic generation process, proposed in [2,16], is briefly described before we introduce our adaptation which+ takes false memories into account.

Types of False Memories and System Responses. Recall, that in the DRM experiment (Sec. 2), the elicited false memories are semantically closely related to the true memories, as a result of the experimental setup. This type of false memories (we denote it with FM_R) can be addressed by retrieval mechanisms that add related terms to a query (e.g., synonym-based expansion, rule-based expansion, pseudo-relevance feedback). If a user searches in his emails with the query "John Saturday meeting" and the email in question contains the term "weekend" instead of "Saturday", the email can be found by such mechanisms.

While this type of false memories does not render retrieval systems ineffective, false memories that lead to a wrong recollection of the nature of the content (we denote this type with FM_F) pose a far more serious problem. For instance, the user might query the system with "John Monday meeting" or "Paul Saturday meeting"; here, the user either incorrectly remembers the time or the person he is going to meet, maybe because the user confused two meetings with each other or remembered the sender of the email, sent a long time ago, incorrectly. In these cases, current retrieval systems are likely to fail or retrieve the correct item at a low rank. Such queries do not (or very rarely) occur in the available known-item topic sets. At the same time, they are likely to occur to some extent in the real-world setting and thus they should be included in topic sets that are utilized to test and evaluate PIM retrieval systems.

Automatic Topic Generation. The known-item topic generation approach originally proposed by Azzopardi et al. [2] was later refined by Kim et al. [16] for the more specific case of PIM test corpora, where a document usually contains a number of fields (such as email sender, calendar entry time, Word document creator, etc.). A known-item/query pair is then generated in five steps:

1. Initialize an empty query $q = ()$
2. Select document d_i to be the known-item with probability $P_{doc}(d_i)$
3. Select the query length s with probability $P_{length}(s)$
4. Repeat s times:
   (a) Select the field $f_j \in d_i$ with probability $P_{field}(f_j)$
   (b) Select the term t_k from field language model of f_j: $P_{term}(t_k|f_j)$
   (c) Add t_k to q
5. Record d_i and q as known-item/query pair

Kim et al. [16] verified that this query generation procedure is more similar to queries generated in their human computation game than queries generated without considering the separate fields. In their work, P_{term} is based only on the

target document, that is, no noise is included in the query generation process. In contrast, Azzopardi et al. [2] proposed to interpolate P_{term} with random noise from the background model (collection language model) to simulate a user with an incomplete recollection of the content. If applied to fields, the term selection probability becomes:

$$P_{term} = \alpha P_{term}(t_k|f_j) + (1-\alpha)P(t_k), \tag{1}$$

where $P(t_k)$ is the probability of drawing t_k from the background model of the respective field. The probabilities P_{field}, P_{doc}, P_{term} and P_{length} can be chosen in a number of ways. Following the experiments in previous work, we draw fields uniformly at random [16], we draw the query length s from a Poisson distribution [2], and rely on TF.IDF based term selection. The TF.IDF based term selection has been shown in [16] to lead to generated queries that are more similar to manually created (TREC) queries than other approaches.

Modelling False Memory. Based on Eq. 1, a first step is to make the parameter α dependent on the time the known item was last seen, instead of fixing it to a particular value across all documents. This step can be motivated by the increase of false memories over time: if a document has not been seen in a year, a user is more likely to have a false memory of it compared to a document last viewed the day before.

Let x_{d_i} be the number of time units since document d_i was last seen and let x_{max} be a time unit where no document specifics are remembered anymore (and $x_{d_i} \leq x_{max}$), then we can model α as follows:

$$\alpha_{d_i} = \left(\frac{x_{max} - x_{d_i}}{x_{max}}\right)^n, \alpha \in [0,1] \text{ and } n > 0 \tag{2}$$

If document d_i has recently been viewed α_{d_i} will be ≈ 1 and little noise is introduced in the query generation process. On the other hand, if a document has not been viewed for a long time, α_{d_i} will be ≈ 0 and a large amount noise is introduced. The parameter n determines how gradual or swift the introduction of noise is over time: the closer n is to 0, the more gradual the memory loss; conversely, the greater n, the quicker the introduction of noise. Adapting the level of noise to the access pattern of the target document is not the only possibility. In Sec. 2 we described how numerous factors (stress, emotional state, context, etc.) affect the encoding and decoding of a memory and if those factors can be measured, they should influence the noise level as well.

We have stated earlier, that random noise (terms drawn from the collection) is not a realistic modelling decision, as users are likely to retain some sense of what the document they look for is about (e.g., a meeting with some person on some day). Recall how in Sec. 2 we discussed the source monitoring framework which has been proposed and empirically validated as an explanation of false memories. Based on it, we model the noise (false memories) as coming from different sources $S_1, .., S_m$. One source may be constructed from the documents semantically related to the known item, another source may be derived from

all emails sent by a particular sender, and so on. External sources may also be utilized as source, e.g., news stories that were published at the time the target document was received/read/sent.

As a consequence, we adapt step 4(b) in the query generation process to include levels of noise that are dependent on the amount of time passed since the document was last seen by the user and to draw noise from a number of sources that are related to the target document:

$$P_{term} = \alpha_{d_i} P_{term}(t_k|f_j) + (1 - \alpha_{d_i}) \left(\sum_{\ell=1}^{\ell=m} \beta_\ell P_{term}(t_k|S_\ell) \right), \; with \; \sum_{\ell=1}^{\ell=m} \beta_\ell = 1 \tag{3}$$

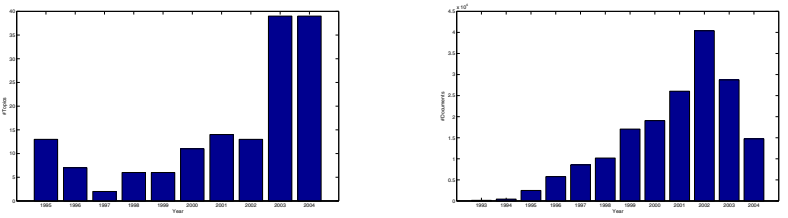
4 Experiments

As PIM test corpora are not publicly available, we consider instead the email corpus (W3C corpus) introduced at the TREC Enterprise track in 2005 [8]. The Enterprise track was developed with the question in mind of how people use enterprise documents (intranet pages, emails, etc.) in their workplace. One of the tasks was the re-finding of emails, which is the task we investigate here. We consider it a reasonable approximation of a PIM search corpus and note that it was also utilized in previous Desktop search experiments [16].

Data Set Analysis. The W3C corpus contains (among others) 198,394 email messages from the public mailing list *lists.w3.org*. A total of 150 topics were developed (25 for training and 125 for testing) by the task participants. Though no detailed information is given in [8] concerning the topic creation process, it can be assumed that the task participants viewed the email messages while developing the topics.

A total of 67 runs were submitted to TREC in 2005 for the email re-finding task. The retrieval effectiveness was measured in mean reciprocal rank (MRR) and success at 10 documents (S@10). The best system achieved a performance of 0.62 (MRR) and 82% (S@10). The task was not further developed in the following years; the performance of the best systems appeared to indicate that known-item search in such an email corpus is not a difficult problem. In the subsequent paragraphs we show that this conclusion can only be drawn if we assume the existence of perfect memory.

In Sec. 2 we described studies that have shown that memory degrades over time. An obvious question is then, how distributed are the documents in this corpus and the 150 target documents (qrels) over time. In Fig. 1 we present histograms (in years) across all corpus documents and the relevant documents only. The documents cover a ten year time span, from 1995 to 2004. While most relevant documents are from 2003 and 2004, more than ten known items are emails written in 1995. If we assume (due to a lack of user logs to investigate actual document access patterns) that the documents were read once when they were received, it becomes clear that perfect queries for those documents is an unreasonable assumption.



(a) Qrels documents (b) Corpus documents

Fig. 1. Histograms of the number of documents according to year of sending

The query generation process in Sec. 3 takes the fields of a document into account. From the corpus we extracted the following fields: *sender*, *subject*, *body* and *sending date*. We then manually assessed the 150 topics and assigned their terms and phrases to one or more of the fields. This assessment evaluated false memories of type FM_F : if the query terms match the subject line (or email body, sender, date) semantically, the terms are judged as being correct memories, even if not all terms occur as such in the emails. If a query's terms are applicable to several fields, e.g., subject and body, they are assigned to all applicable fields. Query terms are deemed a FM_F false memory if they are false in the context of the target email document. For instance, topic *KI6* (Fig. 2) is: *Conference on accessibility and assistive technology at schools*; the known-item specifically discusses a conference on assistive technologies for colleges and universities, not schools; this topic thus contains a FM_F false memory. Due to the topic construction process, we expect very few topics to contain false memories, which we argue is in contrast to real-world queries.

```
<annotatedTopic>
<num>KI6</num>
<qrel>lists-076-5352080</qrel>
<originalEntry>Conference on accessibility
and assistive technology at schools</originalEntry>
<sender></sender>
<date></date>
<subject>Conference assistive technology</subject>
<body>Conference on accessibility and assistive technology at</body>
<falseMemory>schools</falseMemory>
</annotatedTopic>
```

Fig. 2. Topic annotation example (FM_F)

Field	FM_F #Topics	FM_R #Topics
sender	23	22
date	12	17
subject	32	129
body	147	132
false memory	14	51

Fig. 3. Number of topics containing information present in a field

We also performed this topic set analysis automatically, focusing on false memories of type FM_R , that is, we considered the syntactic matching between query terms and document terms. The email corpus and the topics were stemmed (Krovetz) and stopwords were removed[3]. Here, a topic contains a false memory, if at least one of the query terms does not occur in the email document.

[3] All retrieval experiments were performed with the Lemur Toolkit: http://www.lemurproject.org/

In Tab. 3 the results of this analysis are shown. For both FM_F and FM_R , the vast majority of topics contain elements from the subject and/or the email body. Few topics contain additional aspects such as the sender or the date of sending. While the number of false memories is low in the FM_F setting, about a third of emails contain false memories of type FM_R .

In order to investigate how those false memories influence the performance of retrieval systems, we evaluated all 67 runs[4] submitted to TREC in 2005 on the four subsets of topics: (i) the topics without FM_F false memories, (ii) the topics with FM_F false memories, (iii) the topics without FM_R false memories, and, (iv) the topics with FM_R false memories. The question is: Do the same runs that perform well on topics without FM_R or FM_F false memory topics also perform well on the topics with these false memories? The results are shown in Fig. 4. Plotted are the system performances in MRR: the performance on topics without FM_F / FM_R false memories (x-axis) versus the performance on topics with false memories (y-axis). Fig. 4 (left) shows the scatter plot for the topic split according to FM_F and Fig. 4 (right) shows the topic split according to FM_R . We are interested in how similar the system rankings are. Ideally, the system rankings would be the same independent of the topic set. This is not the case, in fact, the rank correlation between the two sets of system performances for the FM_F based topic split is not statistically significantly different from zero (at $p < 0.01$). In contrast, for the FM_R based topic partition, the correlation is significant and a trend is recognizable. However, even here the best retrieval systems across *all* topics do not fare well. The best system across all topics is placed at rank 27 of the FM_F topics, while it is ranked ninth in the FM_R topics. In case of the correct topics, the best system is within the top five ranks, both for the topic partition without FM_F and without FM_R false memory topics. This result shows, that systems that perform well on one type of topics (topics without false memories) may perform rather poorly on topics with false memories; a factor that needs to be taken into account when researching retrieval approaches in PIM. This result also emphasizes the need for more realistic queries, i.e., those with realistic false memories.

Query Generation with False Memories. In this section, we report the results of our query generation approach and its influence on three standard retrieval approaches: TF.IDF, Okapi and Language Modeling with Dirichlet smoothing ($\mu = 1000$). As source S of false memory for a field f_j of the known-item document d_i, we utilize the 1000 most similar fields (cosine similarity) of f_j in the corpus. We evaluate two decay rates, $n = \{1, 2\}$. Finally, we derive topic sets, each of size 100, which contain known-items of different sending date (the "current date" is the day of the most recently correctly time-stamped document in the W3C corpus). The derived topic sets are:

– **Random:** the known-item documents are drawn at random from the corpus; their distribution of document age (document sending date) will resemble Fig. 1.

[4] The runs are available at http://trec.nist.gov/

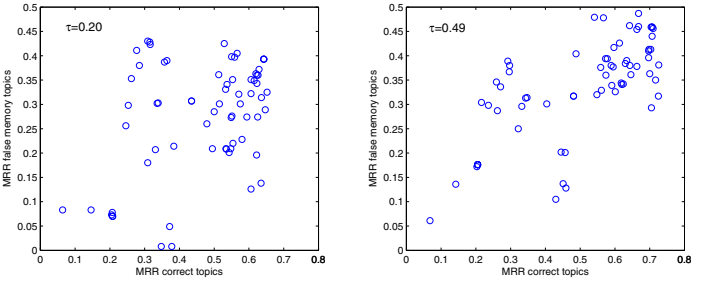


Fig. 4. Scatter plots of system performances (in MRR): on the left, the topics without FM_F false memories (x-axis) are plotted against the topics with FM_F false memories (y-axis). On the right, the topics without FM_R false memories are plotted against the topics with FM_R false memories.

- **Cold:** the known-item documents were not sent within the last year.
- **Warm:** the known-item documents were sent between a year and three months ago.
- **Hot:** the known-item documents were sent within the last three months.

Topics that belong to the "hot" (recently seen) category contain the smallest amount of noise, while topics in the "cold" (not seen for a long time) category are highly likely to contain a lot of noise (Eq. 2). The noise-controlling parameter α_{d_i} is determined for each known-item document d_i by calculating the fraction of years that have passed since the document's creation (x_{d_i}); x_{max} is set to 10 years (the time interval of the corpus). The results are presented in Tab. 1. The worst results are recorded for "cold" queries, which is not surprising as they were generated with the most noise. In general, the results confirm the expectations, no single retrieval approach performs best overall. The absolute performance changes drastically between the hot and cold query sets, indicating the suitability of the model to introduce false memories.

Ideally, we would like to compare the generated queries to an existing topic set (as done in [16]), to investigate the model's ability to generate queries and false

Table 1. Results of known-item retrieval (in MRR) for generated query sets with different sending date characteristics

Query Set	n	TF.IDF	Okapi	Dirichlet LM
Random	1.0	0.465	0.443	0.493
	2.0	0.311	0.368	0.390
Cold	1.0	0.249	0.260	0.251
	2.0	0.234	0.255	0.255
Warm	1.0	0.671	0.690	0.597
	2.0	0.583	0.587	0.596
Hot	1.0	0.701	0.713	0.777
	2.0	0.566	0.699	0.679

memories that are similar to manually created queries and naturally occurring false memories. This is, however, not yet possible, as no known-item topic set exists, which includes topics that were created in a realistic setting.

5 Conclusions

In this work, we have argued for taking cognitive processes into account when generating queries, in particular queries in the PIM setting and the known-item task. We have shown experimentally, that false memories can have a significant impact on the relative performance of retrieval systems and we proposed a false memory based adaptation of the existing query generation procedure.

A limitation of our work is the adhoc nature of the parametrization, e.g., we sampled known items uniformly from the corpus or according to a certain time-stamp range, though it would be very useful to know when the documents, that users typically search for, were last seen by them. In order to compare how well our model approximates the true amount and type of false memories in re-finding queries, we need to collect re-finding queries from real users. To that end, we plan to follow the following two approaches:

(1) In [16] it is argued that the introduced pseudo-desktop corpus is valuable, because the users who played the human computation game were already familiar with the documents. Instead of letting users "play a game" to find the best possible query, we plan to ask a set of users about such publicly accessible e-mails without letting them view the document. Choosing documents that were sent across a wide time span, will give an indication of how large the false memory problem is in this setting. A potential pitfall is here to direct the user to the right document, without biasing the keyword search through the description.

(2) False memories can also be observed in newsgroups and discussion fora. A typical post in a newsgroup or a forum may be: *"I saw a post about how to install this program, but I cannot find it anymore. Can someone post the information again please?"* and one or more of the replies then point to the original post the user was looking for (confirmed by an affirmative statement of the original requester). These are also false memories in a known-item setting: a user is certain that an item exists, but he cannot find it. The posting dates of the different entries also allows an investigation into false memories over time.

References

1. Adar, E., Teevan, J., Dumais, S.: Large scale analysis of web revisitation patterns. In: SIGCHI 2008, pp. 1197–1206 (2008)
2. Azzopardi, L., de Rijke, M., Balog, K.: Building simulated queries for KI topics: an analysis using six european languages. In: SIGIR 2007, pp. 455–462 (2007)
3. Barreau, D., Nardi, B.: Finding and reminding: file organization from the desktop. ACM SigChi Bulletin 27(3), 39–43 (1995)
4. Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., Whittaker, S.: Improved search engines and navigation preference in personal information management. ACM Trans. Inf. Syst. 26(4), 1–24 (2008)

5. Blanc-Brude, T., Scapin, D.: What do people recall about their documents?: implications for desktop search tools. In: IUI 2007, pp. 102–111 (2007)
6. Boardman, R., Sasse, M.: Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In: SIGCHI 2004, pp. 583–590 (2004)
7. Chrobak, Q., Zaragoza, M.: Inventing stories: Forcing witnesses to fabricate entire fictitious events leads to freely reported false memories. Psychonomic Bulletin & Review 15(6), 1190–1195 (2008)
8. Craswell, N., de Vries, A.P., Soboroff, I.: Overview of the TREC-2005 Enterprise Track. In: Proceedings of TREC 2005 (2005)
9. Dodson, C., Bawa, S., Slotnick, S.: Aging, source memory, and misrecollections. Learning, Memory 33(1), 169–181 (2007)
10. Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., Robbins, D.: Stuff I've seen: a system for personal information retrieval and re-use. In: SIGIR 2003, pp. 72–79 (2003)
11. Elsweiler, D., Baillie, M., Ruthven, I.: Exploring memory in email refinding. ACM Trans. Inf. Syst. 26(4), 1–36 (2008)
12. Hyman Jr., I., Husband, T., Billings, F.: False memories of childhood experiences. Applied Cognitive Psychology 9(3), 181–197 (1995)
13. Johnson, M., Hashtroudi, S., Lindsay, D.: Source monitoring. Psychological Bulletin 114(1), 3–28 (1993)
14. Johnson, M., Nolde, S., De Leonardis, D.: Emotional focus and source monitoring. Journal of Memory and Language 35, 135–156 (1996)
15. Johnson, M., Raye, C.: Reality monitoring. Psychological Review 88(1), 67–85 (1981)
16. Kim, J., Croft, W.B.: Retrieval experiments using pseudo-desktop collections. In: CIKM 2009, pp. 1297–1306 (2009)
17. Kim, J., Croft, W.B.: Ranking using multiple document types in desktop search. In: SIGIR 2010, pp. 50–57 (2010)
18. Kintsch, W., Welsch, D., Schmalhofer, F., Zimny, S.: Sentence memory: A theoretical analysis. Journal of Memory and Language 29(2), 133–159 (1990)
19. Lansdale, M.: The psychology of personal information management. Applied Ergonomics 19(1), 55–66 (1988)
20. Norman, K., Schacter, D.: False recognition in younger and older adults: Exploring the characteristics of illusory memories. Memory & Cognition 25(6), 838–848 (1997)
21. Roediger, H., Jacoby, J., McDermott, K.: Misinformation effects in recall: Creating false memories through repeated retrieval. Journal of Memory and Language 35, 300–318 (1996)
22. Roediger, H., McDermott, K.: Creating false memories: Remembering words not presented in lists. Journal of Experimental Psychology-learning memory and cognition 21(4), 803–814 (1995)
23. Roediger III, H., McDermott, K.: False perceptions of false memories. Journal of Experimental Psychology: Learning, Memory, and Cognition 22, 814–816 (1996)
24. Sachs, J.: Recopition memory for syntactic and semantic aspects of connected discourse. Attention, Perception, & Psychophysics 2(9), 437–442 (1967)
25. Teevan, J., Alvarado, C., Ackerman, M., Karger, D.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: SIGCHI 2004, pp. 415–422 (2004)

Protocol-Driven Searches for Medical and Health-Sciences Systematic Reviews

Matt-Mouley Bouamrane[1], Craig Macdonald[2], Iadh Ounis[2], and Frances Mair[1]

[1] Institute of Health and Wellbeing
College of Medical, Veterinary and Life Sciences
[2] School of Computing Science
University of Glasgow, Scotland, UK
{FirstName.LastName}@glasgow.ac.uk

Abstract. Systematic reviews are instances of a critically important search task in medicine and health services research. Along with large and well conducted randomised control trials, they provide the highest levels of clinical evidence. We provide a brief overview of the methodologies used to conduct systematic reviews, and report on our recent experience of conducting a *meta-review* – i.e. a systematic review of reviews – of preoperative assessment. We discuss issues associated with the large manual effort currently necessary to conduct systematic reviews when using available search engines. We then suggest ways in which more dedicated and sophisticated information retrieval tools may enhance the efficiency of systematic searches and increase the recall of results. Finally, we discuss the development of tests collections for systematic reviews, to permit the development of enhanced search engines for this task.

1 Introduction

Systematic reviews (SR) and meta-analyses (MA) of the medical literature are considered to provide – along with large and well-conducted Randomised Control Trials (RCT) – the highest existing level of clinical evidence (level I) [1]. SRs are now routinely used as the starting point for developing clinical guidelines [2]. Guidelines affect the promotion of health care interventions by policy-makers and clinical managers, as well as the provision of care to patients. When systematic reviews fail to produce sufficient evidence to issue guidelines, clinical recommendations are typically based on expert opinions, considered to be the lowest level of clinical evidence (level IV) [1]. SRs often do not provide definitive and authoritative answers to a research question, because of a lack of sufficient available or reliable evidence in the scientific literature. In these cases, the SR highlights gaps in the existing evidence, which in turn may subsequently shape the future agenda for medical interventions as well as research funding priorities [3].

From an information retrieval (IR) perspective, an SR is an instance of a search task with a clearly defined information need *(the research question)*, which entails an explicitly specified, systematically-developed and constrained notion of relevance, in the form of a *search protocol*. Indeed, the process underpinning

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 188–200, 2011.

an SR is guided by published peer-standards, including a *protocol* for deriving search queries and the relevance screening of search results. Hence, we describe SRs to represent a *protocol-driven* search task. Moreover, an SR can be seen as *recall-focused*, as *all relevant* literature must be found.

This paper contributes an overview on the background, motivations and methodologies for conducting SRs, which we believe are both unfamiliar and useful to the IR community. Moreover, using a case study to provide motivations, we make comparisons with other recall-focused IR tasks, and discuss how IR research can potentially contribute to aiding SRs. The remainder of this paper is structured as follows: Section 2 contains a brief introduction to the motivations of SRs; Section 3 introduces the methodologies used to conduct SRs in medicine and health services research; In Section 4, we discuss our recent experience of conducting a systematic review and the challenges encountered using currently available search tools; Section 5 provides a roadmap for research in information retrieval, and describe how test collections might be obtained to allow future evaluation of search tools for SRs; Concluding remarks follow in Section 6.

2 Systematic Reviews

2.1 Issues with the Reporting of Clinical Outcomes

Several studies have highlighted substantial issues within the reporting of clinical outcomes in the literature [4]. For instance: *Publication bias* is the tendency for scientific publications to be biased towards the reporting of significantly effective treatments or studies with a proven demonstration of practical efficiency [5]; *Outcomes reporting bias* occurs when only a selected subset of measures are reported, which produces an incomplete or inaccurate evaluation of study outcomes [6].

To minimise the potential for these biases to misrepresent the effectiveness of treatments, the assessment of clinical evidence in the medical literature is increasingly relying on systematic reviews and meta-analyses. A systematic review (SR) identifies and aggregates *all available evidence* pertaining to a specific research question, using a rigorous and transparent methodological *protocol* guided by peer standards. The protocol specifies clear eligibility and exclusion criteria – in order to provide reliable, accurate, and critically appraised evidence-based clinical reports with a minimum of bias [7,8]. SRs are now common in medicine and other fields. Indeed, in 2004, more than 2500 SRs were reportedly published [10]. A *meta-analysis* (MA) study also provides a similar high level of evidence, but uses statistical methods to aggregate the quantitative results of independent RCT studies [9].

2.2 IR Searches in Systematic Reviews

From an IR perspective, an SR represents an instance of a search task with well-defined information needs and highly constrained definitions of relevance. Moreover, as an SR must assert that *all potentially relevant documents* are retrieved – i.e. full recall *must* be achieved – typically all papers matching the

query are examined, leading to a very low overall precision of the results. The entire retrieved set is screened for relevance, with many potentially topically relevant papers being excluded if they do not meet strict pre-defined inclusion or exclusion criteria. For instance, exclusion criteria may be methodological or based on the type of study (e.g. RCT, case-control, cohort studies). This assures the ultimate integrity of the clinical evidence, by discarding lower quality studies or inadequate methodological approaches, which could undermine the validity of the clinical evidence. The latter can be particularly difficult for existing search engines to detect. In many cases, documents may have been indexed with some meta-data, such as study type or study categories (e.g. Medical Subject Headings (MESH) terms[1]) but this meta-data often remains insufficiently reliable for practical high precision searches, often due to the coarse granularity of the indexing categories given the high specificity of the SR search task [11].

Overall, to attain quality and reproducible SRs, the entire search process is driven by the search protocol. Moreover, these searches are "manually" labour-intensive, with low precision, and are mainly conducted – or at least designed and overseen – by domain-specific experts. Hence, they are *expensive* in time, labour and expertise. In this paper, we describe SRs as representing an archetypal example of a *recall-focused* and *protocol-driven* search task. The IR community can make a significant contribution to supporting search tasks underpinning SRs, if it were capable of developing tools to *optimise searches*, increase the *precision* of searches, while guaranteeing the full recall of *all relevant* documents. In the following, we review the existing standard protocols for systematic reviews, before reporting on the authors' recent experience of conducting an SR, and discussing how IR can contribute to the process of performing systematic reviews.

3 Protocols for Systematic Reviews

As highlighted above, SRs must abide by a search protocol, which detail the survey methodology. In particular, steps such as formulating the query from the information need, screening of results and reporting of conclusions are discussed. In this section, we provide details on the current PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) protocol, to facilitate the explanation of the SR case study that follows in Section 4.

The PRISMA statement [12] is a peer-recommended methodology for conducting SRs. The statement was devised to update earlier recommendations for dealing with issues of perceived inconsistencies and biases in the reporting of meta-analyses of RCTs – for instance, failure to explicitly report the status of intervention concealment[2].

[1] http://www.nlm.nih.gov/mesh/

[2] Intervention concealment ensures that in RCTs, the patients receiving - or not - the treatment, *and* the health professionals directly involved in the provision of the treatment are both blinded to whether the patient belongs to the intervention or control groups, to minimise the bias in the estimates of the effectiveness of treatments.

To address the identified common shortcomings in the methodology of reporting clinical evidence, PRISMA recommends a protocol-based methodological process of reporting critical items identified in the literature reviewed. The omission of these items could undermine the validity of the results reported. PRISMA recommends that SRs report a study selection trial flow in order to determine the criteria that led to studies being included or rejected from the review as well as a methodological check list. The check list provides a method to assess the searches strategies used to identify clinical evidence, the selection criteria, data and characteristics of studies, as well as the processes for validity assessment and quantitative analysis. A structured methodology for reporting meta-analyses is provided, in order to ensure consistency and reliability.

In addition, PRISMA specifies how the identified studies should be filtered at each step of the review, in the form of a flow diagram, shown in Figure 1. In doing so, the quality of the studies included in the report must be pro-actively assessed in order to exclude lower quality studies, the inclusion of which would risk undermining the validity of the synthesised clinical evidence. Hence, the reliability and validity of the results reported in the review could potentially be critically assessed and guaranteed through a process of third-party replication. Moreover, to enable the full reproducibility of the search, authors need to thoroughly describe their methodological protocol for conducting the SR, as well as reporting the quality assessment of the included studies, against the set of reporting criteria defined in PRISMA. The statement recognises that an SR is inherently an iterative search process. The refinement of the review protocol in the course of the study is therefore possible as long as it is both justified and explicitly reported. Moreover, the statement stresses that the review protocol ought to be publicly accessible for peer-review.

Finally, the different forms of biases should be addressed to assure the validity of the results reported in the review. In particular, specific attention needs to be paid to minimise the risk of biases by performing both *"study-level"* and *"outcomes-level"* assessment, while selection bias should be explicitly addressed by reporting the publication status of the included studies.

As will be seen in the following case study, an SR is an instance of a complex IR task, whereby a well-motivated and developed information need is formulated into a cumulative series of queries through an interactive development process. Matching papers are obtained using the queries on a database of publications – such as Medline[3] – which are then exhaustively screened for relevance. The inclusion and exclusion criteria used when screening for relevance are specified *a-priori* in the search protocol. By providing check lists and methodology steps for query formulation, relevance screening and summarising, search protocols such as PRISMA ensure the reproducibility of an SR, but do not reduce the time or expense in conducting it.

Figure 1 shows the flow process of a systematic review (SR). Currently, IR tools are only involved in the initial *Identification* stage. Instead, we argue that they could, and should, provide support for the later *Screening*, *Eligibility* and

[3] http://www.nlm.nih.gov/bsd/pmresources.html

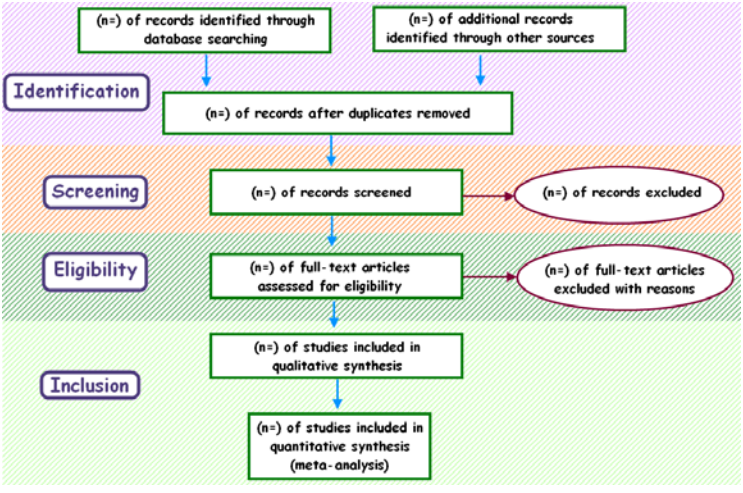


Fig. 1. The PRISMA phases flow of studies selection for a systematic review [8]

Inclusion stages by providing more advanced retrieval models and search interfaces. Yet, several aspects of the iterative process of an SR represent substantial challenges for existing IR techniques. In the following section, we summarise our recent experiences in conducting an SR. Later, we relate these experiences to other investigated tasks in IR, such as legal and patent retrieval. Moreover, we provide new challenges for IR and suggest how models and tools should be enhanced to further support future SR search tasks.

4 Systematic Review Case Study

In this section, we provide the motivations behind our systematic review case study, and the methodology used. Moreover, we provide an overview of the efforts spent and issues identified while screening a large sample of retrieved documents. We use these to formulate the motivation behind improving IR systems to reduce the efforts of conducting SRs.

4.1 Motivation and Methodology

Our systematic review is concerned with the evidence of effectiveness of the existing practices of assessing patients before a surgery (i.e. *preoperative assessment*). The World Health Organisation has estimated that more than 230 million surgical procedures are conducted annually [13]. However, patient-related factors (e.g. hypertension on the day of surgery) can lead to cancellation of surgery. It has been reported that up to two-thirds of day-case and 50% of in-patients cancellations can be attributed to patient-related factors. Indeed, more efficient preoperative processes may prevent a significant number of these cancellations [14].

(anesth* or anaesth* or surgery or surgical or ambulatory or orthopedic procedure* or neurosurg* or preoperative* or elective or minimally invasive of minor surg* or peri-operativ* or pre-procedur* or preoperativ* or preprocedure* or pre-anaesthe* or preanesthe* or preanaesthe* or pre-anaesth* or posteroperative complication* or intraoperative complication* or intra-operative complication*) and (risk* or assess* or test* or scor* or screen* or evaluat* or stratif*)

Fig. 2. Example of a complex Boolean sub-query used within our SR

Between March 2010 and March 2011, we performed an SR of the medical literature in search of the reported evidence underpinning the effectiveness of existing preoperative assessment practices. As such, our SR can be described as a meta-review, in that we sought to identify all previous SRs and MAs on preoperative assessment processes, against well defined eligibility criteria. We developed a search protocol according to the NHS Centre for Reviews and Dissemination guidance for undertaking reviews in health care [7]. This guidance provides step-by-step instructions in developing a search protocol, which we used to complement the PRISMA check list.

We performed a Medline database search in July 2010 using the Ovid search portal tool[4]. Medline is the U.S. National Library of Medicine's bibliographic database. It contains over 18 million references to articles in biomedicine and life sciences. The search was performed and refined over a series of five meetings between the SR protocol team and an information scientist of the University of Glasgow who specialises in bibliographic searches for life sciences. We initially used a broad search strategy for identifying reviews of preoperative assessment using generic query terms such as "preoperative risk assessment, evaluation, screening, testing", combining keywords, MESH terms, and study types such as "reviews" and "meta-analyses". MESH terms returned entirely unmanageable numbers of results (i.e. in excess of 30,000). The search strategy was progressively refined, until a manageable 8522 abstracts were retrieved from Medline (after restricting to English language). The final used query comprised a series and combination of over 20 complex Boolean sub-queries. An example of one subquery is given in Figure 2, demonstrating manual stemming and Boolean aspects.

The retrieved titles and abstracts were screened independently by two expert reviewers, using strict inclusion and exclusion criteria, such that only SRs and MAs of preoperative assessments were included in the study.

4.2 Results Overview

The results of the SR are reported in detail in [15]. However, Figure 3 shows the trial flow (as per the PRISMA protocol), and the number (n) of studies included or excluded at each stage of the study. In particular, a majority (n=8040) of titles and abstracts screened, did not meet our strict inclusion criteria, because of two reasons (i) *low relevance*: the studies focused on a clinical intervention

[4] http://www.ovid.com/site/products/ovidguide/medline.htm

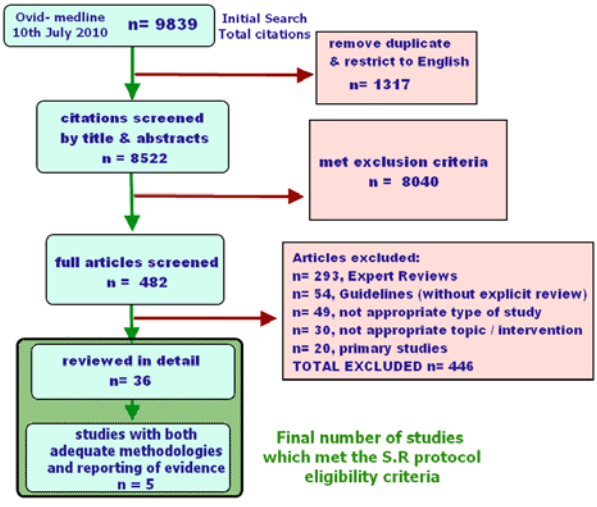


Fig. 3. Flow chart of our SR on preoperative assessments

that was only marginally relevant to preoperative assessment (e.g. preoperative therapy, intervention or medication, intra- or post-operative interventions and treatments) or was not the appropriate type of study (e.g. a primary study of a clinical intervention rather than a systematic review of the effectiveness of processes) or (ii) *too high specificity*: the studies described highly specific clinical processes, interventions or populations (e.g. surgery on a specific organ for a specific type of disease), which were not generic and thus not deemed useful for the purpose of our review. The full text of a further n=482 studies were screened, with only 36 studies meeting the final selection criteria, both for relevance and quality of methodology. These 36 studies were analysed in detail, from which n=5 were deemed relevant with respect to all inclusion and exclusion criteria, including the reporting of adequate clinical levels of evidence. We identified several issues while conducting this SR:

High Screening Workload: The screening of 8522 abstracts, retrieving and assessing 482 full articles and selecting and extracting data from the final set of 36 included studies - with respect to the search protocol - is an extremely laborious process. Each abstract or paper was independently reviewed by two researchers, in order to minimise the risk of bias, as recommended by PRISMA. Initially, a large amount of time was spent screening and discarding clearly irrelevant studies. Of the full paper screening, many of these studies were subsequently rejected based on methodological criteria. Here, although the topic of the study was clearly relevant to the search, the quality of the methodology of the studies was deemed insufficient, according to the search protocol criteria, to guarantee the identification of reliable clinical evidence.

Table 1. Ranks of papers examined within the ranking ranges of the initial Ovid Medline search

Criteria	Ranking ranges
n=482 full paper screened	rank 35 to 8457
n=36 relevant & met inclusion criteria	rank 419 to 8457
n=5 met inclusion criteria & specified level of clinical evidence + n= 3 further documents identified	rank 521 to 4096 (521, 1989, 2281, 3502, 4096)

Relevance Ranking: Studies that did fully meet our inclusion criteria for relevance (n=419) and clinical evidence (n=521) were very lowly ranked in the Ovid search results (see Table 1). In addition, one relevant document was identified at a very low rank (8457) which would suggest that the number of results examined was justified for achieving high recall.

Search Snapshots: A practical issue faced during the systematic review was related to the reproducibility of searches using the Ovid tool. Indeed, while it is possible to save queries in Ovid (e.g. to formulate complex Boolean queries), it is not possible to save the actual *search results* themselves. About 700,000 new records are added to Medline every year, which translates to in excess of tens of thousands of new studies every week. Instead, to facilitate easy management of the systematic review, as well as the reproducibility of the results, the ability to obtain a snapshot of the search results at a given point in time is paramount.

Manual indexing: Many articles indexed as "reviews" were not necessarily reviews - never mind systematic reviews - while the search queries typically retrieved a large volume of articles that were only marginally relevant to the core topic of preoperative assessment. Of those articles that were deemed relevant to our research topic (full-paper screening), a vast majority were expert opinion reviews or expert opinion-based guidelines (over 70% of full papers screened) and thus did not meet the study type selection criteria (SR or MA).

Overall, of the 8522 papers screened in the initial Ovid Medline search, only 5 studies both adequately reported their search protocol and selection criteria, and provided explicit and adequate grading of clinical recommendations. The mean rank at which these 5 studies were retrieved in the initial Medline search was 2478, demonstrating the lack of precision in the search results. A further 3 relevant studies were identified through a complementary search in other local repositories - a problem also noted by [16]. Effectively, this means that less than 1 in 236 (8522/36) studies met our search protocol inclusion criteria and less than 1 in a 1000 documents explicitly provided grading of clinical recommendations. Moreover, only the most basic IR functionalities were used. We argue that SRs could be easier to conduct if there were new retrieval models and search engines that can support complex constraints and the recall-focused nature of the task. For instance, a faceted search interface that contains facets pertaining to common inclusion or exclusion criteria (e.g. built using classification techniques) would

allow the exploration and iterative reduction of the set to be screened. In the next section, we discuss other recall-focused tasks that have recently been investigated in IR, then describe a roadmap for improving the IR technology used for SRs, and discuss possible evaluation methodologies for such techniques.

5 Towards Protocol-Driven IR

In this section, we compare and contrast the systematic review of medical literature with other domain-specific recall-focused IR tasks, before proposing a roadmap of how IR research can address the SR search task. Finally, we discuss evaluation methodologies to facilitate IR research in the SR task.

5.1 Recall-Focused Search Tasks

e-Discovery is the process of a negotiated discovery of electronically-stored documentary evidence during a legal case. In particular, lawyers negotiate a complex Boolean query, which aims to identify relevant (known as 'responsive') documents to the plaintiff party, and to exclude private documents belonging to the defendant that should not be revealed to the plaintiff. Similar to systematic reviews, recall is important, as a legal argument may hinge on the discovery of a supporting responsive document. However, the search protocols within systematic reviews place more constraints on relevance. Since 2006, the TREC Legal track has operationalised the e-Discovery task within an IR research setting. Results thus far indicate that the negotiated Boolean queries can miss up to 50% of the responsive documents (a similar problem has been reported for SRs [17]), but that single retrieval systems could only demonstrate small improvements over the retrieval using the negotiated Boolean queries [18].

Patent Prior-Art Search is the process whereby other patents relating to a given patent are identified. Such patents may be cited within the patent, or could be used to invalidate the patent. Once again, recall is an essential aspect of this task. However, in contrast to a systematic review or e-Discovery, the given patent can be used as the query, instead of a complex Boolean query developed within a search protocol or by legal negotiations. Patent prior-art search has been investigated within several evaluation forums (e.g. TREC & CLEF). In particular, the TREC Chemical track has ran since 2009 [19], focusing on chemical patents alone. Participating groups made use of citations, as well as advanced entity tagging of chemical entities [19].

5.2 Roadmap of IR Research for Systematic Reviews

The systematic review search task is characterised by several dimensions. In particular, while recall is very important and the notion of relevance is very constrained, for every SR there is some practical maximum number of abstracts that can be screened. In the following, we enumerate ways in which recall can be enhanced, screening effort can be minimised and how the IR system can be more fully utilised in the SR process:

Maximising Screening Effort: In SRs, Boolean queries have been classically used to limit the size of the retrieval set (but at the risk of reduced recall [20]). However, more intelligent methods of deciding on a cut-off for the retrieved set are possible. For instance, [21] suggests using already-identified studies to find a cut-off point which ensures recall but minimises the size of the retrieved set to be screened. Instead, as an alternative, in a similar manner to the Legal track, relaxing Boolean queries to use proper relevance rankings should permit more relevant papers to be identified [20]. Moreover, we believe that it is possible to make probabilistic guarantees on the number of identified relevant documents attained by a given rank cut-off, inspired by [22].

Enhancing Relevance Ranking: [21] found that simply adding the term 'versus' to the query improved the results quality for 61 SR searches in Medline. Although a heuristic, this suggests that enhanced query reformulation and ranking models could improve SR searches. For instance, classical recall enhancement techniques such as query expansion and collection enrichment [23] may introduce further relevant documents not found using the strict Boolean queries. Moreover, the work of the TREC Genomics track (2003-2007) [25] is of note, for its handling of genome-related retrieval from Medline. However, it did not tackle recall-focused tasks such as SRs. Lastly, with the prevalence of feature-based models in modern IR, we see the potential for further improving retrieval by the deployment of learning to rank techniques [24] adapted to high recall environments.

Constrained Relevance: With many dimensions of relevance prescribed by the search protocol and the various inclusions and exclusion criteria, the IR system should aim to facilitate common selection criteria by providing various document classification models [26]. For instance, in our SR case study in Section 4, studies that have been conducted or written according to agreed standards (e.g. graded clinical recommendations) are relevant, and could be identified using citation analysis. Moreover, we found that many expert opinions were retrieved. Techniques from NLP [27] and IR [28] for identifying subjective documents may be appropriate for identifying expert opinions (which should not be relevant to a systematic review). This is an example of a negative relevance problem, which has been found to be challenging in areas such as relevance feedback.

Exploratory Search Interface: With many options for constraining the retrieved documents, an improved search engine over Medline could provide a faceted search interface [29], allowing the researchers to iteratively explore the retrieved documents and develop inclusion and exclusion criteria in a manner directly supported by the engine. While faceted retrieval systems have been popular in supporting transactional search tasks such as shopping, recent developments have encompassed their application to identify key blogs on the blogosphere [28], and to automatically suggest appropriate facets for each query [29].

As can be seen from the list above, research addressing problems in systematic reviews encompass different problem areas in IR, including machine learning, models and interfaces. In the next section, we provide suggestions on appropriate methodologies for evaluating IR research on the SR search task.

5.3 Evaluation Methodologies

Given the volume [10] and expense [21] of the systematic reviews that are conducted every year, as well as the potential for IR technology to improve the SR process, we argue that there is a case for the development of standard IR test collections covering this search task. A test collection for SRs could leverage the experience garnered by the TREC Legal and Chemical tracks in evaluating recall-focused tasks. Of note, a characteristic common to both the Legal and Chemical tracks is the use of stratified sampling in the relevance assessment of the documents identified by the participating systems, to reduce the assessing workload to a manageable level. However, using such sampling methods means that, in practice, only estimates of recall of the participating systems are obtained. In contrast, an SR test collection could be created in co-operation with an on-going systematic review, thereby potentially enhancing the recall of the study, as well as obtaining the relevance assessments as a side-product of the review. Boudin et al. [30] describe another alternative methodology, where relevance assessments are bootstrapped from already published SRs, but with the disadvantage of missing relevant documents not identified by the original systematic reviews.

Judges have recognised that technology derived from the TREC Legal track may become acceptable for e-Discovery [31]. Similarly, once lessons learned from IR research into SRs result in improved search systems for medical researchers, revisions to the PRISMA search protocols may relax the effort burden in the searching for relevant literature in SRs when such enhanced IR tools are utilised.

6 Conclusions

We described the motivations behind conducting systematic reviews of the medical literature, namely the identification of *all relevant* clinical results pertaining to a specific research question. SRs are conducted using peer-accepted standard protocols that define the search process. Moreover, as a case study, we reported our experience of conducting a recent SR. We compared SRs to similar recall-focused IR tasks, and provided a roadmap for future IR research to enhance SR searches. Finally, we discussed the possibilities, difficulties and benefits of conducting a TREC-style evaluation efforts for systematic reviews search tasks.

Acknowledgements. This research is funded by the Scotland Chief Scientist Office through a postdoctoral training fellowship (2010/2013 - M-M. Bouamrane).

References

1. Evans, D.: Hierarchy of evidence: a framework for ranking evidence evaluating healthcare interventions. Journal of Clinical Nursing 12(1), 77–84 (2003)
2. NICE: The guideline development process - an overview for stakeholders, the public and the NHS (3rd ed.). National Institute for Health and Clinical Excellence (2007)
3. Doyle, J., Waters, E., Yach, D., McQueen, D., De Francisco, A., Stewart, T., Reddy, P., Gulmezoglu, A.M., Galea, G., Portela, A.: Global priority setting for Cochrane systematic reviews of health promotion and public health research. Journal of Epidemiology and Community Health 59(3), 193–197 (2005)
4. Chan, A.W., Hróbjartsson, A., Haahr, M.T., Gøtzsche, P.C., Altman, D.G.: Empirical evidence for selective reporting of outcomes in randomized trials. Journal of the American Medical Association 291(20), 2457–2465 (2004)
5. Sterne, J.A.C., Egger, M., Smith, G.D.: Investigating and dealing with publication and other biases in meta-analysis. BMJ 323(7304), 101–105 (2001)
6. Kirkham, J.J., Dwan, K.M., Altman, D.G., Gamble, C., Dodd, S., Smyth, R., Williamson, P.R.: The impact of outcome reporting bias in randomised controlled trials on a cohort of systematic reviews. BMJ 340 (2010)
7. NHS-CRD: Centre for Reviews and Dissemination's guidance for undertaking systematic reviews in health care. University of York (2009), http://www.york.ac.uk/inst/crd/
8. Liberati, A., Altman, D.G., Tetzlaff, J., Mulrow, C., Gotzsche, P.C., Ioannidis, J.P., Clarke, M., Devereaux, P., Kleijnen, J., Moher, D.: The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. Journal of Clinical Epidemiology 62(10), e1–e34 (2009)
9. Moher, D., Cook, D.J., Eastwood, S., Olkin, I., Rennie, D., Stroup, D.F.: the QUOROM Group: Improving the quality of report of meta-analyses or randomised controlled trials: the QUOROM statement. The Lancet 354, 1896–1900 (1999)
10. Moher, D., Tetzlaff, J., Tricco, A.C., Sampson, M., Altman, D.G.: Epidemiology & reporting characteristics of systematic reviews. PLoS Medicine 4(3), e78 (2007)
11. Liu, Y.H.: On the potential search effectiveness of MeSH (medical subject headings) terms. In: Proceedings of IIiX 2010, pp. 225–234 (2010)
12. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G.: The PRISMA Group: Preferred reporting items for systematic reviews & meta-analyses: The PRISMA statement. PLoS Medicine 6(7) (2009)
13. World Health Organization: Safe surgery saves lives. WHO world alliance for patient safety. WHO report (2008)
14. NHS Modernisation Agency: National good practice guidance on pre-operative assessment for in patient surgery (2003)
15. Bouamrane, M.-M., Gallacher, K., Marlborough, H., Jani, B., Kinsella, J., Richards, R., van Klei, W., Mair, F.S.: Processes of preoperative assessment in elective surgery: a systematic review of reviews (2011), under review
16. Beahler, C.C., Sundheim, J.J., Trapp, N.I.: Information retrieval in systematic reviews: Challenges in public health arena. American Journal on Preventative Medicine 18, 6–10 (2000)
17. Golder, S., McIntosh, H., Loke, Y.: Identifying systematic reviews of the adverse effects of health care interventions. BMC Medical Research Methodology 6(1), 22 (2006)

18. Oard, D.W., Baron, J.R., Lewis, D.D.: Some lessons learned to date from the TREC Legal track (2006-2009). Technical Report, University of Maryland (2010)
19. Lupu, M., Huang, J., Zhu, J., Tait, J.: TREC-CHEM: large scale chemical information retrieval evaluation at TREC. SIGIR Forum 43, 63–70 (2009)
20. Pohl, S., Zobel, J., Moffat, A.: Extended Boolean retrieval for systematic biomedical reviews. In: Proceedings of ACCS 2010, pp. 117–126 (2010)
21. Zhang, L., Ajiferuke, I., Sampson, M.: Optimizing search strategies to identify randomized controlled trials in MEDLINE. BMC Medical Research Methodology 6(23) (2006)
22. Arampatzis, A., Kamps, J., Robertson, S.: Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In: Proceedings of SIGIR 2009, pp. 524–531 (2009)
23. Kwok, K.L., Grunfeld, K., Chan, M., Dinstl, N., Cool, C.: TREC-7 ad-hoc, high precision & filtering experiments using PIRCS. In: Proceedings of TREC-7 (1998)
24. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
25. Roberts, P.M., Cohen, A.M., Hersh, W.R.: Tasks, topics and relevance judging for the TREC Genomics track: five years of experience evaluating biomedical text information retrieval systems. Inf. Retr. 12(1), 81–97 (2009)
26. Cohen, A., Hersh, W., Peterson, K., Yen, P.Y.: Reducing workload in systematic review preparation using automated citation classification. Journal of the American Medical Informatics Association 13(2), 206–219 (2006)
27. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
28. Macdonald, C., Santos, R.L.T., Ounis, I., Soboroff, I.: Blog track research at TREC. SIGIR Forum 44 (2010)
29. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Yee, K.P.: Finding the flow in web site search. Commun. ACM 45, 42–49 (2002)
30. Boudin, F., Nie, J.Y., Dawes, M.: Deriving a test collection for clinical information retrieval from systematic reviews. In: Proceedings of DTMBIO 2010, pp. 57–60 (2010)
31. Oard, D.W., Hedin, B., Tomlinson, S., Baron, J.R.: Overview of the TREC 2008 Legal track. In: Proceedings of TREC 2008 (2008)

Enhanced Information Retrieval Using Domain-Specific Recommender Models

Wei Li, Debasis Ganguly, and Gareth J.F. Jones

Centre for Next Generation Localisation
School of Computing, Dublin City University, Dublin 9, Ireland
{wli,dganguly,gjones}@computing.dcu.ie

Abstract. The objective of an information retrieval (IR) system is to retrieve relevant items which meet a user information need. There is currently significant interest in personalized IR which seeks to improve IR effectiveness by incorporating a model of the user's interests. However, in some situations there may be no opportunity to learn about the interests of a specific user on a certain topic. In our work, we propose an IR approach which combines a recommender algorithm with IR methods to improve retrieval for domains where the system has no opportunity to learn prior information about the user's knowledge of a domain for which they have not previously entered a query. We use search data from other previous users interested in the same topic to build a recommender model for this topic. When a user enters a query on a new topic, an appropriate recommender model is selected and used to predict a ranking which the user may find interesting based on the behaviour of previous users with similar queries. The recommender output is integrated with a standard IR method in a weighted linear combination to provide a final result for the user. Experiments using the INEX 2009 data collection with a simulated recommender training set show that our approach can improve on a baseline IR system.

Keywords: Domain-Specific Information Retrieval, Recommender Algorithm.

1 Introduction

The ever increasing volume of information available in our daily lives is creating increasing challenges for document retrieval technologies. One area of growing interest in information retrieval (IR) research is the exploration of methods to enable users to find documents which meet their personal information needs by taking advantage of their previous search history. This is the focus of the area of Personalized Information Retrieval (PIR), seeks to form a model of each user's search interests using their previous search history, and then uses this to assist in more reliably retrieving documents of interest to this user in subsequent search activities. Where the user is searching in a topical area of on-going interest such an approach can prove effective. However, in practice, users may enter queries on new topics which they have not searched on previously. The related field of Recommender Systems (RSs) exploits ratings of items from multiple users to make predictions of

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 201–212, 2011.

items which future users interested in the same topic may find useful. In recent years, RSs have started to appear in many applications where user feedback is available, for example in online applications such as *YouTube*, *Amazon*, and *Ebay*. These systems record the behaviour of users to build models of user interests, and use these to predict items which may be interest to a current user based on feedback from previous ones.

Existing PIR methods require personal information from the specific user in order to build user profiles. This data can be collected by asking users to input their personal information preferences, including for example topics of interest or keywords, recording their search queries and clicking and viewing behaviour when browsing retrieved results or by asking them to rate some items or give other explicit feedback. In other web search personalization technologies, data is collected without user involvement by exploiting the clustering of retrieved documents in order to create a complete personal user profile based on characterization of their search history. These approaches have been found to perform well in the modelled domains [8][9]. However, this approach will not work for new domains where the individual user has not provided personalized information, and it is not realistic to gather such information from them before retrieval operations begin. In this situation it is desirable to make use of any information which is available from previous searchers with similar interests to improve retrieval effectiveness for a user without previous search experiences in the topical domain of his/her query. To do this, we propose to gather feedback from previous user queries, either recording explicit feedback of relevance of retrieved items to a query or implicit feedback in terms of the time a user dwells on each item. This feedback information then can be used to train recommender models for potentially interesting items for any new searchers who is interested in this topical domain. In this work we introduce an approach to do this by combining recommender technologies with a standard IR method to produce domain-specific IR where user driven domain models are used to enhance the effectiveness of standard IR.

We explore our proposed method using a simulated search scenario based on the INEX 2009 Wikipedia document collection. We simulate previous user search behaviour by automatically constructing variations of selected search topics to train a recommender model for the topical domain of each search topic. These recommender models are then used in combination with a language modelling based standard IR system for search with the original INEX 2009 search topics. The combined search method shows improvement in IR search effectiveness for both precision and recall metrics over a baseline standard IR approach.

The remainder of this paper is organized as follows: Section 2 provides a brief review of relevant existing research in PIR and RSs, Section 3 presents the framework of our proposed combined domain-specific IR model, Section 4 describes our experiments using the INEX 2009 test collection and gives experimental results, and finally Section 5 provides conclusions of our work so far and details of our planned further investigations.

2 Related Work

A number of existing studies have explored the topic of PIR aiming to provide users with more personalized information provision, while other studies have explored the development of RSs. Personalization involves capturing the search interests of individuals and using these to train individual user interest models [25]. There are two broad methods of capturing information for personalization: i) implicit feedback, where user interests are inferred from their behaviour such as which documents they click on in the output of a search, their reading time for retrieved documents or their scrolling actions on a document; ii) explicit feedback where users manually confirm document relevance or their topical interests [25]. Both IR and RSs use these two methods to perform personalization. In this section we look first at existing work in PIR and then review relevant studies looking at RSs.

2.1 Personalized Information Retrieval

PIR is currently being explored mainly in the area of Web search [25]. For example, some standard search engines are examining implicit feedback (mainly by extracting some useful information from the items which a user has so far viewed) to refine the user's query to provide a more personalized response, e.g. Google, Yahoo! Meanwhile some web search applications are exploring explicit feedback and hybrid approaches combined implicit and explicit feedback, e.g. Flickr, Youtube. These systems ask users to provide tags for source collections or to express their personal descriptions or opinions about some items. For instance, in Flickr, users store and annotate their own photos. These tags can be considered to be expressions of user interests, and can be used to build user profiles which can be exploited in personalized search. Our research is currently looking at only the use of implicit feedback for domain-specific IR, since gathering explicit feedback in the environments that we are considering would be less practical in terms of user participation.

In addition to web search, PIR is also becoming an important factor in other areas, e.g. education, healthcare [25]. Explicit feedback cannot always be gathered in these areas since users may be reluctant to express their opinions or give ratings to items. Because of this, many researchers focus only on collecting implicit feedback. Some studies take account of information about users' behavioural information, such as click-through data, dwell time while browsing, etc. which can be obtained implicitly from user observation. Kelly and Belkin [11] report that using only display time information averaged over a group of users to predict document usefulness is not likely to be accurate, nor is it accurate using display time for a single user without taking into account contextual factors[15]. For this reason, user information beyond the content of the issued queries are taken into account [15][1], which is information about users and their context information. This additional information is often gathered implicitly from user behaviour and contextual data, topic knowledge and task information. In our research looking at search in a specific domain, dwell time is the most important factor since it is the only personal data that we can gather from users. In the environment we are working with the topic knowledge of the user and the associated contextual data is unavailable. Thus despite its apparent limitations we

are exploring whether dwell time can be exploited as useful for information for the construction of domain-specific recommender models.

2.2 Recommender System

RSs attempt to recommend items that are likely to be of interest to users [25]. Typically, a RS compares user profiles with some reference characteristics, and uses these to predict the rating that a user may give to a new item which he has not considered yet. These characteristics may be associated with the information contained in the item (the content-based approach) or the user's social environment (the collaborative filtering approach) [25]. In this paper we consider only the collaborative filtering approach to RS, other recommender algorithms will be the subject of future work. As exemplified in [22][16], a RS collects user profile information in the same ways as IR systems. Since, as described earlier, we are unable to collect explicit feedback in our environment, we consider collection of implicit feedback. Implicit data collection includes:

- Observing the items that a user views.
- Analyzing item/user viewing time.
- Keeping a record of the items that a user has purchased.
- Obtaining a list of items that a user has listened to or watched on their computer.
- Analyzing the user's social network and discovering similar likes and dislikes.

RSs compare the collected data to similar or non-similar data collected from previous users. A list of recommended items can then be calculated for the current user. For our recommender model, we simulate collection of user data from the first two sources as: i) observing the items user views; ii) the viewing time for that item. However, in our preliminary experiment, we assume that when the user inputs a query to our model, we compare the similarity between this query and previous users search information to select a suitable recommender model.

The collaborative filtering approach makes automatic predictions about the interests of a user by collecting preference information from many other users [25]. There are different types of collaborative filtering methods available: memory-based (measures the similarity between pairs of users to give the prediction, the Pearson scheme is a well-known memory-based scheme), model-based (finds patterns from the training data, such as SVD, Bayes methods [25]) and the rating-based approach (predicts how a user would rate an item from other users rating, such asthe *inDiscover* website) [13]. In our investigation, we explore the rating-based collaborative filtering approach. In our work we chose the Weighted SlopeOne algorithm to compute predictions since it is efficient to query, reasonably accurate, and supports both online querying and dynamic updates, which makes it a good candidate for real-world systems[13]. The Weighted SlopeOne algorithm comprises of the following two steps:

- Looking for other users who rate the same item as the current user.
- Using the ratings from these like-minded users to give predictions for our current user.

Rating-based collaborative filtering requires data pairs from a user: the item ID and its relevance weighting entered either explicitly or inferred from implicit data such as viewing time. The response of a RS to a query pair is an array of recommended pairs (item ID, rating) of items based on the training data captured from previous users, which the current user has not rated yet.

For simplicity of explanation in our investigation individual users are assumed to enter one query on a topic, but this need not be the case in an operational system.

3 Combining Information Retrieval with Domain-Specific Recommender Models

Our method for integrating domain-specific recommender models with information retrieval proceeds as follows:

- The system records each query entered by previous users to search the available document archive, and implicit feedback from the users of the relevance rating of each retrieved document to viewed by each user, indicated by the time that the searcher spends on viewing the document.
- The ratings of each viewed document for each topical query domain are then used to train a recommender model domain using the Weighted Slope One algorithm.
- When a query is entered into the combined search method, a standard IR technique is used to retrieve search results from the available document collection. This query is also used to select an appropriate recommender model from the available domain-models generated from previous search data. The RS is then used to give predictions of potentially relevant documents based on the selected recommender model.
- The results of the IR search and RS predications are then integrated using a linear combination of the scores for each retrieved document. Documents are then re-ranked using the combined scores and returned to the user.

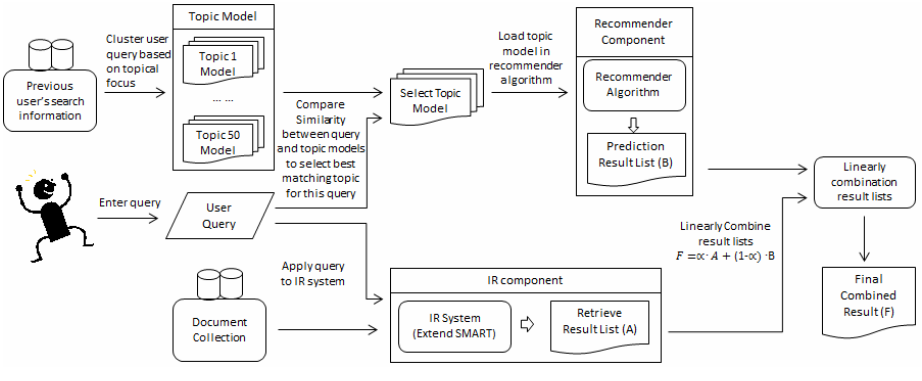


Fig. 1. Overview of the enhanced domain-specific IR incorporating the recommender component

Fig.1. shows in the combined domain-specific IR and RS model. This has two components: IR search and recommender prediction. In our experimental implementation of this approach, we use the extend SMART IR system to use a language modelling IR method [18]to retrieve the results for IR component.

The training and prediction of each recommender domain-model operates as follows: the domain-model training set for each recommender is based on a set S of all previous queries closely related to the new query. This can be viewed as the following matrix (Equation (1)): where $P_{n,m}$ is a pair of data (D_m, $R_{n,m}$), where D_m is document m and $R_{n,m}$ is the rating given to document m for previous query n. This information is then used to run the Weighted Slope One algorithm (2).

$$S = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \dots \\ P_n \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \dots & P_{1,m} \\ P_{2,1} & P_{2,2} & P_{2,3} & \dots & P_{2,m} \\ P_{3,1} & P_{3,2} & P_{3,3} & \dots & P_{3,m} \\ \dots & \dots & \dots & \dots & \dots \\ P_{n,1} & P_{n,2} & P_{n,3} & \dots & P_{n,m} \end{bmatrix} \tag{1}$$

$$P_{(u)_j} = \frac{\sum_{i \in (S-\{j\})} (u_i + dev_{j,i}) \cdot card(S_{j,i})}{\sum_{i \in (S-\{j\})} card(S_{j,i})} \tag{2}$$

This algorithm can take into account both the information from other users who rated the same item and from other items rated by the current user. For our current experiment we only consider the former. The number of ratings is also observed which means the number of users who rated the pair of items i and j are recorded and considered. If we know that user u gives rating u_i to item i, then we can predict the rating u_j that this user will give to item j based on all previous user information in the recommender set S. This is computed by Equation(2):where $P_{(u)_j}$ is the prediction of the rating that user u will give to item j; $card(S_{j,i})$is the number of previous queries receiving a rating for i and j in set S; $dev_{j,i}$ is the average deviation of document I with respect to item j, computed using Equation (3).

$$dev_{j,i} = \frac{\sum_{P \in S_{j,i}} P_{k,j} - P_{k,i}}{card(S_{j,i})} (i \neq j) \tag{3}$$

The similarity score of an item j with respect to query q(denoted as $RS(q|j)$), is computed using the standard language modelling approach implemented into the SMART retrieval system (see Equation (4)). Equation (4) ranks a document j by the probability of generating the given query q from it, denoted as $RS(q|j)$. $P(t|j)$ denotes the probability of generating a term t from document j and $P(t)$ denotes the probability of generating it from the collection, λ being the smoothing parameter to account for the inverse document frequency (*idf*) factor of a term t. The final weight of document

$j(FW_j)$ is computed using a linear combination of the recommender and IR scores as shown in Equation (5).

$$RS(q \mid j) = \prod_{t \in q} \lambda \cdot P(t \mid j) + (1 - \lambda) \cdot P(t) \tag{4}$$

$$FW_j = \alpha \cdot P_{(u)_j} + (1 - \alpha) \cdot RS(q \mid j) \tag{5}$$

For IR systems, the main challenge is to improve the search results for a user's query, in this integrated method, a recommender algorithm is exploited to address this challenge. On the other hand, the key problem for RSs is cold start, the Weighted Slope One algorithm needs other users search information to give predictions for the current user. However, if the recommender set *S* suffers from data sparse, the prediction results will not be worth considering. In practice the data sparse condition will apply for some user's queries, in this case there will not be an effective topical domain model available for them, the output of recommender algorithm will be empty or unreliable. Thus integration is only advisable if the recommender domain model has sufficient training data, otherwise conventional IR approach should be preferred. The question of when to apply our integrated approach in the case of limited training data will be considered as part of our future work.

4 Experiment

To investigate our proposed approach to domain-specific search, we require a suitable set of experimental data which enables us to build recommender models for a range of topical search interests based on previous users' interaction behaviour within a suitably challenging IR task. These requirements mean that experimentation poses many challenges. Since this is a new research area, there are no suitable test collections readily available. Ideally we would like to have access to real collections and importantly large logs of queries and interaction data from real users querying this data. However, since we do not have access to this type of datasets, and it is unrealistic for us to be able to collect such a dataset working in an academic environment or to gain access to such data from other sources, we must seek ways to simulate them in order to explore our proposed approach. In order to conduct this initial study we extended an existing test collection to simulate our search environment. We assume a scenario of avisitor to a museum with an interest in a topic entering a query to identify items which may be of interest to them within the available collection. Subsequently other users enter similar but different queries on the same topical area. The relevant items returned for each of these queries can then be gathered to form the training set for a recommender model for this topic. The relevance rating for each occurrence of each relevant document is taken from the viewing time for each relevant document. Thus anitem retrieved for many queries with high ratings will be given a high relevance prediction value by the RS for this topic. For our experiment we assume search interests for items on a number of separate topical areas and use these to build recommender model for each of them.

The following subsections describe the development of the test collection used for our initial investigation.

4.1 Data Collection

The INEX 2009 Wikipedia document collection comprising of 2,666,190 documents was selected as our starting data collection. For this investigation we simulated previous user search interaction information as follows: 20 topics were chosen from the INEX 2009 topic dataset, the criteria for choosing topic is that they should be 4 words or longer in length. 10 variations of each topic were created as a simulation of similar queries in the same topical area entered by previous users. Topic variants were made by randomly deleting one or two words from the original topic. Hence we needed to select topics statements of 4 or more words. For example, for the original topic: *"Physicists scientists alchemists periodic table elements",* two of the variations were as follows: 1) *Physicists scientists alchemists periodic results; 2) Physicists scientists alchemists table results.*

This is obviously a very simple strategy for creating topic variations, but serves to enable us to carry out our current experiment. A particular issue which needs to be considered when modifying queries in this way is the potential impact on the set of documents which are relevant to each topic. For this initial investigation, the relevance set is assumed to not vary a lot for each topic variation. We are currently working on more sophisticated methods to simulate query variants to obtain more realistic training datasets for our experiments.

In this experiment, each recommender model is built in the following way:

- For each original topic, make 10 variants by random as deleting one or two words.
- The 10 topic variants for each search topic were entered as queries into the extended SMART system to obtain 10 ranked lists of potentially relevant documents.
- The retrieved results for the 10 variations of each topic were clustered in one group. As described above, the aim of this step is to simulate results obtained for 10 different users who interested in the same topic. The topic variations mean that slightly different result lists are obtained for each pseudo user search query.
- The 10 ranked lists for each topical area were compared against the *qre l* files of the original topics to identify retrieved true relevant documents retrieved for each topic variant. Rating values were then assigned to each document randomly. The ratings simulated browsing time as an indication of implicit feedback. These were assigned in the range 0.5-1.0 for each relevant document for the original topic, and 0.0-0.49 for documents which were non-relevant.
- Each of the top 150 documents in the retrieved ranked lists with rating information is seen as one previous users searching behaviours. The processed retrieved ranking list for the 10 variants were integrated into one group and used as a recommender model for their corresponding original query. We thus obtained our simulation data for previous users searching the document collection.

4.2 Experiment Setup

Recommender models for the 20 selected topics were built as described in section 4.1. For our experiment we assume that a searcher has an interest in one of our 20 topical domains and enters the original search query to look for relevant documents that they might be interested in. The query is applied to the extended SMART retrieval system to obtain a ranked document list. This represents our baseline retrieval output. The ranked IR retrieval list is then compared to find the appropriate recommender model from those available. The recommender model selection proceeds as follows. We assume the retrieved ranked list is a vector $Q=(d_{1,q}, d_{2,q}, d_{3,q},\ldots\ldots,d_{t,q})$,we have 20 recommender models in our experiment$(R_1, R_2, R_3,\ldots\ldots, R_{20})$, recommender model k is a set S_k (Equation (1)) and can also be viewed as a vector, i.e. recommender model k can be seen as a vector $R_k = (P_{1,k}, P_{2,k}, P_{3,k}, \ldots\ldots,P_{n,k})$ $(j\in[1,20])$, where $P_{i,j}$, is the result for one previous query in recommender model k. The similarity between the query vector and each recommender vector is:

$$Sim(Q, R_k) = \sum_{t=1}^{20} f(d_{j,q}, R_k) \tag{6}$$

Where $f(d,R)$ is the frequency of item $d$in the recommender model R_k based on set S_j. The recommender model with the highest similarity is selected as the best matching topical domain for the input query. Here t from 1 to 20, which means we only go through the top 20 documents in the retrieved ranked list. This is then used to calculate the prediction of the rating that our current user would give to each of the available documents. Finally, the recommended ranking results are linearly combined with the baseline retrieval list to output our final integrated results. In this experiment, the parameter α as shown in Equation (5) is set to 0.25 using informal empirical experimentation.

4.3 Result

Our scenario task is a user entering a query on entering a museum. We thus focus on providing results to users which show them where they should go next. Our aim here then is to find the most relevant documents on the top of the list in order to give the user most precisely results which direct them to really interesting documents in their next step, i.e. we are interested particularly in precision at high rank cutoff of the retrieved list.

Results for this experiment are calculated using the standard trec_eval software, and are shown in Table 1. Our baseline (BL) IR results are output by the SMART system. The domain-specific IR (DSIR) results combine this result with the recommender system as shown in Equation (5). Results are shown for total number of documents retrieved, no of relevant items retrieved, precision at rank cut-offs of 5, 10, 20 and 100, and standard Mean Average Precision (MAP).

Table 1. Retrieval results for 20 topics with simulated recommender training

Topic	20_Topic_BL	20_Topic_DSIR
Total Number Retrieved	29950	29950
Total Number Relevant	1166	1166
Total Relevant Retrieved	355	355
MAP	0.0744	0.1303(+75.03%)
P@5	0.2600	0.5000(+92.30%)
P@10	0.2200	0.3560(+61.81%)
P@20	0.1750	0.2000(+14.23%)
P@100	0.0835	0.1320(+58.08%)

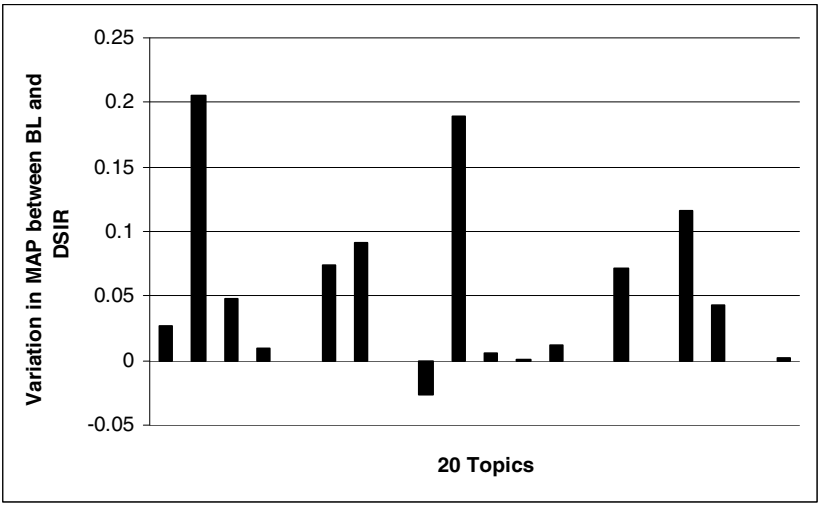


Fig. 2. Variation of MAP between BL and DSIR approaches for 20 original test topics. Calculated as MAP of DSIR - MAP of BL

From Table 1 we can see that the DSIR approach achieves a MAP of 0.1303 which represents an impressive increase of+75.03% on the baseline IR system. The precision at top 5 cut-off is increased from baseline 0.2600 to 0.5000 (+92.30%). This is partially matches our aim of seeking to promote relevant documents to the top of the ranked list. This demonstrates that the recommender algorithm can help to aid standard IR methods. Figure 2 shows the deviation of MAP between BL and DSIR approach for 20 original topics, calculate by the MAP (DSIR)- MAP (BL). From Figure 2, we can clearly see that for the selected 20 topics, the average performance of DSIR is better than our baseline. The reason that it cannot perform well on all topics is that its results depend on the previous users visiting information. If the recommender model we choose for the current user is correct and contains items that are relevant to its topic, the recommender algorithm will locate it and give it as a prediction to the user. In this experiment, of the 20 evaluation topics, 4 of them were

assigned to the wrong recommender model. Improving the reliability of recommender assignment will be an area of our further work.

5 Conclusion and Future Work

In this paper we have proposed a domain-specific IR method combining ranked IR and RSs methods. Experiments with a simulated search environment show that this integration has the potential to improve retrieved results over standard IR methods. While this initial experiment shows promising results, further work needs to be done to develop a more realistic experimental environment. For example, to use a more sophisticated model for query variations in training the RSs. Additionally, while the results so far are encouraging, there are various ways to improve the baseline IR, including methods such as relevance feedback. In our further work, we will explore integration of methods such as these to compare their contribution to improving IR with that of the recommender based approach.

The scenario we are exploring here considers a searcher exploring a new domain of interest. Thus we expect our searcher to be exploring a number of items. When doing this we can make use of feedback as they explore items to adapt the RS in a personalized manner. Also, if they are learning about a new topic, there will often be a preferable order in which information should be viewed. So ultimately we are interested not just in identifying relevant items, but also determining the order in which they are presented in order to maximize the efficiency with which information is provided to the searcher. New evaluation strategies will be required in order to measure the effectiveness with which relevant items can recommended to the searcher in an optimally efficient sequence.

Acknowledgements. This research is supported by the Science Foundation Ireland(Grant 07/CE/I1142) as part of the Centre for NextGeneration Localisation (CNGL) project at DublinCityUniversity.

References

1. Belkin, N.J.: Some (what) Grand Challenges for Information Retrieval. ACM SIGIR Forum 42(1), 47–54 (2008)
2. Billsus, D., Pazzani, M.: Learning Collaborative Information Filtering. In: Proceedings of the AAAI Workshop on Recommender Systems (1998)
3. Campi, A., Mazuran, M., Ronchi, S.: Domain Level Personalization Technique. In: Proceedings of VLDB 2009 (2009)
4. Chee, S.H.S., Han, J., Wang, K.: An Efficient Collaborative Filtering Method. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2001. LNCS, vol. 2114, pp. 141–151. Springer, Heidelberg (2001)
5. Geva, S., Kamps, J., Lethonen, M., Schenkel, R., Thom, J.A., Trotman, A.: Overview of the INEX 2009 Ad Hoc Track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2009. LNCS, vol. 6203, pp. 4–25. Springer, Heidelberg (2010)
6. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of ACM SIGIR 1999 (1999)

7. Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center of Telematics and Information Technology, University of Twente (2001)
8. Jansen, B.J., Spink, A., Bateman, J., Saracevic, T.: Real Life Information Retrieval: A Study of User Queries on the Web. ACM SIGIR Forum 32(1) (1998)
9. Jeon, H., Kim, T., Choi, J.: Adaptive User Profiling for Personalized Information Retrieval. In: Proceedings of ICCIT 2008, pp. 836–841 (2008)
10. Jin, R., Si, L.: A Study of Methods for Normalizing User Ratings in Collaborative Filtering. In: Proceedings of ACM SIGIR 2004, pp. 568–569 (2004)
11. Kelly, D., Belkin, N.J.: Display Time as Implicit Feedback: Understanding Task Effects. In: Proceedings of SIGIR 2004, pp. 377–384 (2004)
12. Kuhlthau, C.C.: Inside the Search Process: Information Seeking from the User's Perspective. Journal of the American Society for Information Science 42(5), 361–371 (1991)
13. Lemire, D., Maclachlan, A.: Slope One Predictors for Online Rating-Based Collaborative Filtering. In: Proceedings of SIAM Data, SDM 2005 (2005)
14. Linden, G., Jacobi, D., Jennifer, A., Benson, E.A.: Collaborative Recommendations Using Item-to-item Similarity Mappings. In: Proceedings of SPSS (2001)
15. Liu, J., Belkin, N.J.: Personalizing Information Retrieval for Multi-Session Tasks. The Roles of Task Stage and Task Type. In: Proceedings of ACM SIGIR 2010 (2010)
16. Oard, D., Kim, J.: Implicit Feedback for Recommender System. In: Proceedings of the AAAI Workshop on Recommender Systems (1998)
17. Pennock, D.M., Hirvutz, E.: Collaborative filtering by personality diagnosis. In: Proceedings of IJCAI 1999 (1999)
18. Ponte, J.M., Croft, W.B.: A Language Modeling Approach to Information Retrieval. In: Proceedings of ACM SIGIR 1998 (1998)
19. Saric, A., Hadzikadic, M., Wilson, D.: Alternative Formulas for Rating Prediction Using Collaborative Filtering. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 301–310. Springer, Heidelberg (2009)
20. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-Based Collaborative Filtering Recommender Algorithms. In: Proceedings of WWW 2010, pp. 285–295 (2001)
21. Shaw, J.A., Fox, E.A.: Combination of Multiple Searches. In: Proceedings of ACM SIGIR 1993 (1993)
22. Song, Y., Nauyen, N., He, L., Imig, S., Rounthwaite, R.: Searchable Web Sites Recommendations. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011 (2011)
23. Vallet, D., Cantador, I., Jose, J.M.: Personalizing Web Search with Folksonomy-based User and Document Profiles. 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. 420–431. Springer, Heidelberg (2010)
24. White, R., Kelly, D.: A Study of the Effects of Personalization and Task Information on Implicit Feedback Performance. In: Proceedings of CIKM 2006, pp. 297–306 (2006)
25. Wikipedia, `http://en.wikipedia.org`

Exploring Ant Colony Optimisation for Adaptive Interactive Search

M-Dyaa Albakour[1], Udo Kruschwitz[1], Nikolaos Nanas[2], Dawei Song[3], Maria Fasli[1], and Anne De Roeck[4]

[1] University of Essex, Colchester, UK
malbak@essex.ac.uk
[2] Centre for Research and Technology - Thessaly, Greece
[3] Robert Gordon University, Aberdeen, UK
[4] Open University, Milton Keynes, UK

Abstract. Search engines have become much more interactive in recent years which has triggered a lot of work in automatically acquiring knowledge structures that can assist a user in navigating through a document collection. Query log analysis has emerged as one of the most promising research areas to automatically derive such structures. We explore a biologically inspired model based on ant colony optimisation applied to query logs as an adaptive learning process that addresses the problem of deriving query suggestions. A user interaction with the search engine is treated as an individual ant's journey and over time the collective journeys of all ants result in strengthening more popular paths which leads to a corresponding term association graph that is used to provide query modification suggestions. This association graph is being updated in a continuous learning cycle. In this paper we use a novel automatic evaluation framework based on actual query logs to explore the effect of different parameters in the ant colony optimisation algorithm on the performance of the resulting adaptive query suggestion model. We also use the framework to compare the ant colony approach against a state-of-the-art baseline. The experiments were conducted with query logs collected on a university search engine over a period of several years.

1 Introduction

Search engine interfaces have evolved rapidly in the last decade. Modern Web search engines do not only return a list of documents as a response to a user's query but they also provide various interactive features that help users to quickly find what they are looking for and assist them in browsing the result space. Google wonder wheel[1] is a popular example of an interactive interface that provides visualised query refinement suggestions. Beyond Web search we also observe more interaction emerging as illustrated by the success of AquaBrowser[2] as a navigation tool in digital libraries.

Studies have shown that users want to be assisted in this manner by proposing keywords [29], and despite the risk of offering wrong suggestions they would prefer having

[1] http://www.googlewonderwheel.com
[2] http://serialssolutions.com/aquabrowser/

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 213–224, 2011.

them rather than not [28]. Query recommendation systems are at the core of these interfaces and they can for example rely on a domain model that reflects the domain characteristics such as a concept hierarchy or simply some term association graph. Several methods have been proposed in the literature to build such models. This includes mining query logs which capture the "collective intelligence" of the user population's search behaviour to build either static models or structures that evolve over time.

Inspired from the social behaviours of animals in nature swarm intelligence has attracted a lot of attention from Artificial Intelligence (AI) researchers [3]. Ant Colony Optimisation (ACO) has been studied extensively as a form of swarm intelligence technique to solve problems in several domains such as scheduling [25], classification [19] and routing problems in telecommunication [5]. Recently ACO has been applied to learn adaptive knowledge structures from query logs [7]. However, *adaptive* knowledge structures are inherently difficult to assess and here we present the first study that investigates how ACO can be explored systematically (and fully automated) to build adaptive knowledge structures for interactive search.

The main contribution of this study is to firstly demonstrate how ACO can be used in a continuous learning process for interactive search, and secondly to illustrate how this biologically inspired mathematical model can be applied in areas such as Information Retrieval (IR) and interactive search where user assessments are critical and can therefore present a serious bottleneck. User evaluations that explore novel algorithms (for interactive search for example) typically rely on user studies, tend to be expensive, are not easy to replicate and do not allow an exploration of the multitude of parameters that are often needed to be tuned in machine learning systems which would require a large number of iterations in the evaluation process. Furthermore, they can be affected by a great deal of subjectivity in the users' perceptions. Therefore it is desirable to be able to perform extensive offline experiments where these models can be tested and their performance over time is observed before applying them in a realistic environment.

A new evaluation framework has recently been proposed to automatically assess the performance of query suggestion systems over time based on actual query logs [20]. This framework is capable of measuring the performance of an adaptive system over time and comparing a number of different adaptive systems under the same experimental conditions. This paper presents an experimental study of an ACO-based algorithm to build an adaptive model applying this automatic evaluation framework. We explore different variations of the ACO model for providing query recommendation and discuss the outcomes in comparison with a sensible baseline approach. The experiments were conducted on search logs comprising more than 1 million queries that have been collected on a university search engine over a period of 3 years.

The evaluation framework helped us in exploring the effects of different parameters of the ACO algorithm which results in a deeper understanding of what factors are effective in generating good query recommendations and which ones are not.

2 Related Work

Interactive information retrieval has received much attention in recent years, e.g. [21,18, 27]. Furthermore, increased activity in developing interactive features in search systems

used across existing popular Web search engines suggests that interactive systems are being recognised as a promising next step in assisting information seeking.

The idea of supporting a user in the search process by interactive query modifications has been discussed extensively [10]. There is also evidence that integrated interactive IR systems can offer significant advantages over baseline systems [31]. Query suggestions can help in the search process even if they are not clicked on [16].

AI researchers have always been interested in representing knowledge in such a way that it can be utilised by automatic reasoning systems. Conceptual graphs are examples of such structures [26]. The knowledge structures more commonly used to make query modification suggestions in an interactive search engine are similar but often simpler than conceptual graphs and can be constructed automatically from documents content, e.g. [22, 17, 30].

With the increasing availability of search logs obtained from user interactions with search engines, new methods have been developed for mining search logs to capture "collective intelligence" for providing query suggestions as it has been recognised that there is great potential in mining information from query log files in order to improve a search engine [13, 23]. Given the reluctance of users to provide explicit feedback on the usefulness of results returned for a search query, the automatic extraction of implicit feedback has become the centre of attention of much research. Queries and clicks are interpreted as "soft relevance judgements" [6] to find out what the user's actual intention is and what the user is really interested in. Query recommendations can then be derived, for example, by looking at the actual queries submitted and building query flow graphs [4], query-click graphs [6] or association rules [11]. Jones *et al.* combined mining query logs with query similarity measures to derive query modifications [15].

The automatic evaluation of search systems that does not rely on expensive user judgements has long been attracting the IR researchers e.g. [24, 12]. This is however not an easy exercise and unlike commonly understood TREC [3] measures (such as Mean Average Precision), there is no commonly agreed automatic evaluation measure for adaptive search. One approach for automatic evaluation is using search logs. Joachims shows how clickthrough data can replace relevance judgements by experts or explicit user feedback to evaluate the quality of retrieval functions [14]. Search logs were used by Dou *et al.* to simulate different personalisation re-ranking strategies and then evaluate those from the actual user clicks [9]. The evaluation framework we employ is to the best of our knowledge the only framework that allows the automatic evaluation of evolving query recommendation suggestions based on actual query logs.

3 ACO for Learning from Query Logs

ACO has been applied recently to adapt domain models from intranet query logs [7].

The domain model built with ACO takes the form of a graph structure where nodes are query phrases and edges point to possible query refinements. To illustrate what the domain model looks like, Figure 1 represents a partial directed graph learned from actual query logs of an academic organisation. The weights on the edges encodes the importance of the association between the nodes (the queries). The graph can be used

[3] `http://trec.nist.gov`

to recommend queries by starting from the initial query node and then traversing the graph edges to identify and rank associated query nodes. For the query 'timetable' and using the graph one can recommend all the directly associated nodes as query refinements and rank those using the edge weights which would result in the list ('exam timetable','courses','timetable office', 'departmental timetable').

The ACO analogy is used to first populate and then adapt the directional graph. In this analogy the edges in the graph are weighted with the pheromone levels ants, in this case users, leave when they traverse the graph.

Constructing the Model. The user traverses a portion of the graph by using query refinements (analogous to the ant's journey), the weights of the edges on this route are reinforced (increasing the level of pheromone). Over time all weights (pheromone levels) are reduced by introducing some evaporation factor to reflect unpopularity of the edge if it has not been used by ants. In other words, we reduce the weight of non-traversed edges over time, to penalise incorrect or less relevant phrase refinements. In addition we expect outdated terms to be effectively removed from the model, i.e., the refinement weight will become so low that the phrase will never be recommended to the user.

Let us assume that we update the pheromone levels on a daily basis. For the edge $q_i \rightarrow q_j$ the pheromone level w_{ij} is updated using equation 1

$$w_{ij} = N * ((1-\rho)w_{ij} + \Delta w_{ij}) \tag{1}$$

where:

- N is a normalisation factor, as all pheromone trail levels are normalised to sum to 1.
- ρ is an evaporation co-efficient factor
- Δw_{ij} the amount of pheromone deposited at the end of the day for the edge $q_i \rightarrow q_j$. The amount of pheromone deposited should correspond to ant moves on the

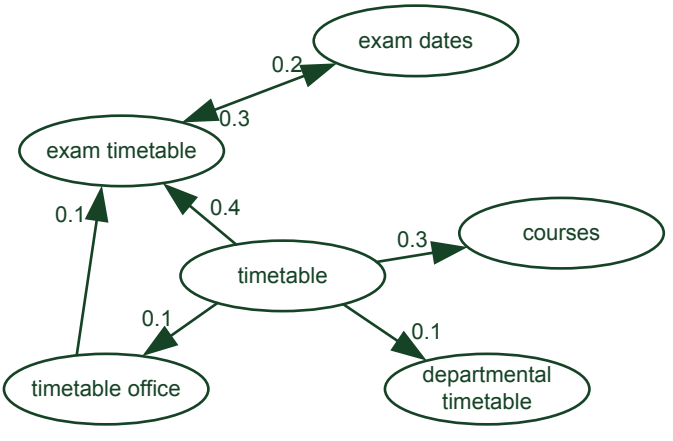


Fig. 1. A partial domain model learned from query logs

graph. In our case, this can be the frequency of query refinements corresponding to the edge. Also the cost of ant moves can be taken into account when calculating the amount of pheromone deposited. Generally it can be calculated using equation 2 [8].

$$\Delta w_{ij} = \Sigma_k Q/C_k; \text{ For all ant moves on edge } q_i \rightarrow q_j \quad (2)$$

where:

- Q is a constant, for this constant we chose the average weight of all edges in the graph in the previous day.
- C_k is the cost of ant k journey when using the edge $q_i \rightarrow q_j$.

With the automatic evaluation we will experiment with different values for the evaporation co-efficient factor in equation 1.

We will experiment with different pheromone calculation schemes for equation 2. In any of these cases all user sessions in the log are extracted for the day for which we are updating the edge weights for and the queries for each session are time ordered. The following schemes are considered:

1. **Immediate Refinements:** In this case, we will consider only consecutive refinements found in user sessions, e.g., for a session containing a query modification chain q_k, q_l, q_m we consider the edges $q_k \rightarrow q_l$ and $q_l \rightarrow q_m$.
   When calculating Δw_{ij} as in equation 2, it will be set to zero if no edges are identified as above. The cost C will be set to 1, and therefore Δw_{ij} will be equal to the frequency of the refinements identified multiplied by the average weight.
2. **Linking All:** In this case, not only consecutive refinements are taken into account but for each query in a user session, apart from the last one, all following queries in the session are linked to that query. e.g., for a session containing a query modification chain q_k, q_l, q_m, we consider the edges $q_k \rightarrow q_l$, $q_k \rightarrow q_m$, and $q_l \rightarrow q_m$ as ant movements.
   When calculating Δw_{ij} as in equation 2, it will be set to zero if no edges are identified as above. For the cost C_k we chose the value of *dist* where *dist* is the length between the queries in the session, e.g. for the edge $q_k \rightarrow q_m$ the cost will be 2.

The ACO procedure is illustrated in Algorithm 1. Note that a nominal update value of 1 for the constant Q in Equation 2 is used for our first day, however, any positive real number could have been chosen without affecting the outcome of normalisation.

Although in our description of the algorithm, the weights are updated on a daily basis, update sessions could be run hourly or weekly, or even when a certain number of user sessions have completed. In addition, it is possible to run the algorithm from any point in the user log to any other, this allows us to compare how the model performs for particular time periods.

Recommending Query Modifications with ACO. To suggest possible modification for a query phrase, we first find the original query phrase in the graph, then list the connected nodes ranked by the edge weights connecting them to the original query phrase. Indirect associations could also be taken into account. In this case sub-trees

Algorithm 1. The ACO-based algorithm to build and evolve the domain model

Input: domain model as a graph G, daily association A_d, number of days DAY_NUMS
Output: G

```
1  for d ← 1 to DAY_NUMS do
2      A_d ← FindAllAssociations(d)
       /* update weights of traversed edges                    */
3      foreach (q, q') ∈ A_d do
           /* Query q' is associated to q in a session on day d.
              */
4          n ← FindNode(G, q)
5          if n = NULL then n ← AddNode(G, q)
6          n' ← FindNode(G, q')
7          if n' = NULL then n' ← AddNode(G, q')
8          e ← FindEdge(G, n, n')
9          τ ← CalculateDepositedPheromone(q, q')
10         if e = NULL then
11             e ← AddEdge(G, n, n')
12             SetWeight(G, e, τ)
13         else
14             SetWeight(G, e, τ + GetWeight(G, e))
15     NormaliseAllWeights(G)
```

can be used to link the original node with nodes further down in the sub-trees. More sophisticated approaches can be utilised for query recommendation such random walks on the graph proposed by Boldi *et al.* [2].

To understand the value of adding these indirect associations, in our automatic evaluation we experimented with two approaches:

1. **Depth 1:** Only direct associations are considered.
2. **Depth 2:** Indirect associations are also considered. However only nodes which are linked with a maximum number of 2 edges are considered. In this case the weights of these indirect links are calculated as the product of the weights of both edges.

4 The Evaluation Framework

The automatic evaluation framework assesses the performance of query suggestion systems over time based on actual query logs by comparing suggestions derived from a domain model to query modifications actually observed in the log files. What we call domain model here and in the following discussion can be any type of model that produces for any given query a ranked list of query recommendation suggestions. The validity of the framework has been confirmed with a user study [1].

The model's evaluation is performed on arbitrary intervals, e.g. on a daily basis. For example, let us assume that during the current day, three query modifications have been submitted. For each query modification pair, the domain model is provided with the initial query and returns a ranked list of recommended query modifications. We take

the rank of the actual modified query (i.e., the one in the log data) in this list, as an indication of the domain model's accuracy. So for the total of three query modifications in the current day, we can calculate the model's Mean Reciprocal Rank (*MRR*) score as $(1/r_1 + 1/r_2 + 1/r_3)/3$, where r_1 to r_3 are the ranks of the actual query modifications in the list of modifications recommended by the model in each of the three cases. More generally, given a day d with Q query modification pairs, the model's Mean Reciprocal Rank score for that day MRR_d is given by Equation 3 below.

$$MRR_d = (\sum_{i=1}^{Q} \frac{1}{r_i})/Q \tag{3}$$

Note that in the special case where the actual query modification is not included in the list of recommended modifications then $1/r$ is set to zero. The above evaluation process results in a score for each logged day. So overall, the process produces a series of scores for each domain model being evaluated. These scores allow the comparison between different domain models. A model M_1 can therefore be considered superior over a model M_2 if a statistically significant improvement can be measured over the given period.

The described process fits perfectly a static model, but in the case of dynamic experiments as we are conducting here, the experimental process is similar. We start with an initially empty domain model, or an existing domain model. Like before, the model is evaluated at the end of each daily batch of query modifications, but unlike the static experiments it uses the daily data for updating its structure.

It is important to mention here that we do not try to identify query modifications within a user session that are actually related. Therefore even subsequent queries that are not related are treated as a query modification pair. However these noisy query modification pairs do not affect the evaluation methodology as this noise is common for all evaluated models.

5 The Experimental Setup

The aim of the experiments is to illustrate how the performance of the ACO-based algorithm can be assessed systematically using an automated evaluation framework and report the results with reference to different parameters that affect the performance of the algorithm.

The experiments conducted try to answer the following questions:

- Is an ACO-based algorithm capable of learning to produce better query modification suggestions in a continuous learning cycle?
- What are the effects of the evaporation co-efficient and different pheromone updating schemes in ACO?
- What are the effects of using direct or indirect associations in ACO?
- How does ACO compare to a state-of-the-art baseline system?

In this section we first provide a description of the search logs used in these experiments. Then we introduce the experimental design and illustrate the different models being tested.

5.1 Search Log Data

The search log data in our experiments are obtained from the search engine of the website of an academic institution. These logs have been collected since November 2007 and so far more than 1.5 million queries have been collected. Each record in our query logs contains a time stamp of the transaction, the query that has been entered and the session identifier.

5.2 Experiments

Following the approach in [11] and to reduce noise we only consider sessions where the number of queries is less than 10 and those which span over the period of less than 10 minutes. We used weekly batches to update the domain models. The number of sessions used for testing and training per weekly batch varies between around 800 sessions and over 3000 sessions. Sessions which do not satisfy the above criteria and are only one query long are not counted here as they are not used for testing and training. The variation in number of sessions between batches is due to some busy periods throughout the year. For instance the first weeks of both academic years 2008 and 2009 consist of more than 3000 sessions each.

ACO Evaluation. The following ACO configurations have been tested:

1. **Depth:** As mentioned before, the depth refers to the number of jumps between nodes used to recommend queries. Two value are used (depth=1, depth=2). The default is 1.
2. **The evaporation co-efficient factor:** We use two different values to test the effect of the evaporation co-efficient ($\rho = 0$, $\rho = 0.5$). The default value is 0.
3. **The pheromone updating schemes:** Two different update schemes are considered (immediate refinements, linking all) The default one is 'immediate refinements'.

The automatic evaluation framework has been run on a combination of settings for these three parameters. The following combinations have been used: ACO, ACO(depth=2), ACO($\rho = 0.5$) and ACO(Linking all). Note that when the parameter is not mentioned the default value will be used.

For each of these combinations, the automatic evaluation has been run on the log data over the period of 3 years (156 weeks) from 20 Nov. 2007 to 20 Nov. 2010. This gives us *MRR* scores for each model on weekly intervals.

Comparing ACO a against a baseline. We compare the ACO runs against a simple alternative based on association rules [11]. Fonseca's approach represents a sensible baseline for a different way of adapting the search because it accesses exactly the same resources as our proposed methods and it has been shown to work well on Web log data. The idea is to use session boundaries and to treat each session as a transaction. Related queries are derived from queries submitted within the same transaction.

6 Results

After running the evaluation framework for the models ACO, ACO(depth=2), ACO($\rho = 0.5$) and ACO(Linking all) we obtain the weekly *MRR* scores for each model. Using

the MRR scores, we can assess the model performance over time, and compare the performance of different models.

Table 1 summarises for the three different variations ACO(depth=2), ACO($\rho = 0.5$) and ACO(Linking All) the average percent increase of the *MRR* scores over the default ACO model and the p value of the two-tailed t-test against the default ACO model.

Table 1. Summary of ACO results

vs. ACO	per. increase(%)	paired t-test
ACO(depth = 2)	4.35	< 0.0001
ACO($\rho = 0.5$)	-2.43	< 0.0001
ACO(Linking All)	-0.64	< 0.0001

The difference between ACO and ACO(depth=2) is extremely significant with an average percent increase of 4.35% which suggests that using indirect associations can be useful. However the introduction of the evaporation co-efficient has a negative effect and the model performance degraded with a positive ρ value. With evaporation the model forgets less commonly used and seasonally incorrect query refinements over time and thus awards those which are becoming more popular. The results here suggest that forgetting does not have a positive effect on the performance. Moreover, the 'Linking All' pheromone update scheme resulted in a negative impact on the performance which suggests that linking only subsequent queries is a better strategy for building the model.

Figure 2 illustrates the results of running the automatic evaluation framework to perform a comparison between the best performing ACO model and the Fonseca baseline. We see that despite the spikes the general trend is upwards indicating that different adaptive learning methods are able to learn from past log data over time. The

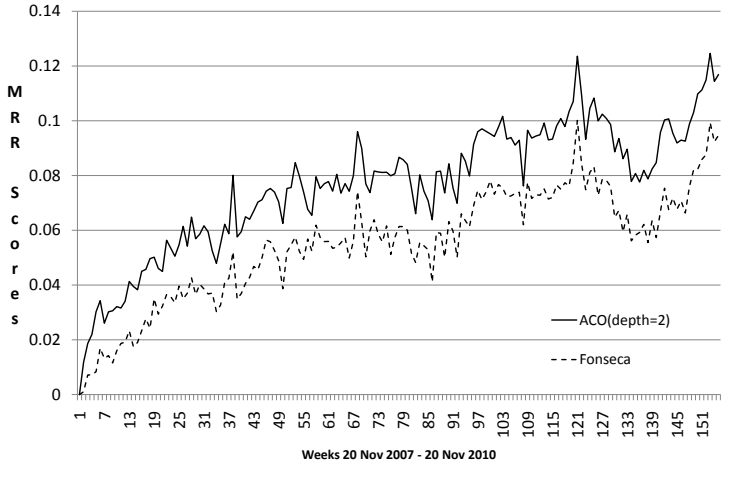


Fig. 2. Comparing ACO against a baseline

ACO(depth=2) model is outperforming the Fonseca baseline. The average percent increase for ACO(depth=2) over Fonseca is 30.62% and the p value for the two-tailed t-test is ($p < 0.0001$) indicating that the difference is statistically extremely significant. This demonstrates the power of using an automatic evaluation framework to experiment in vitro with various models based on real world data and improve their performance through fine tuning and appropriate modifications.

7 Discussion and Future Work

Query recommendations have become a popular aspect of interactive search, and the ant colony optimisation algorithms have shown to be a promising approach to learn useful structures from query logs that can be utilised in query recommendation. In this paper we explored variations of the ant colony optimisation algorithm by conducting controlled, deterministic and fully reproducible experiments. The experiments are based on an automatic evaluation framework that uses real world data to assess the performance of adaptive models.

Our in vitro experiments allowed us to quantitatively answer a series of research questions and to draw very useful conclusions. When evaluating the ACO model, we observed that taking into account indirect associations between queries has a positive effect on the accuracy of the recommendations, while in contrast, a positive pheromone coefficient has a negative effect on performance, because it causes the "forgetting" of important query associations over time. We also found out that associating only subsequent queries in a user's session is a better strategy for building the model as associating all the following queries to a single query had a negative impact on the performance.

ACO performs significantly better than an association-rule-based approach. This exemplifies the importance of continuous learning for the task at hand. We are confident, that the experimental framework will allow us to further improve the learning algorithms and to try further alternatives. In fact we have already conducted a set of experiments to evaluate yet another biologically inspired algorithm to build adaptive models but the results were not reported due to the lack of space.

Other future plans include extending the experiments with further data collected from our search engine and trying out different experimental settings to investigate various aspects of the interactive search and query recommendation. We intend for example, to investigate a way to incorporate click through data and text analysis to further improve the domain models and their evaluation. The in vitro experimental methodology provides the means for extensive further research work.

Acknowledgements. This research is part of the AutoAdapt research project. AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1.

References

1. Albakour, M.-D., Kruschwitz, U., Nanas, N., Kim, Y., Song, D., Fasli, M., De Roeck, A.: Autoeval: An evaluation methodology for evaluating query suggestions using query logs. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 605–610. Springer, Heidelberg (2011)

2. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Proceeding of CIKM 2008, pp. 609–618. ACM, New York (2008)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc., New York (1999)
4. Bordino, I., Castillo, C., Donato, D., Gionis, A.: Query similarity by projecting the query-flow graph. In: Proceedings of SIGIR 2010, Geneva, pp. 515–522 (2010)
5. Caro, G.D., Dorigo, M.: Antnet: Distributed stigmergetic control for communications networks. Journal of Artificial Intelligence Research 9, 317–365 (1998)
6. Craswell, N., Szummer, M.: Random Walks on the Click Graph. In: Proceedings of SIGIR 2007, Amsterdam, pp. 239–246 (2007)
7. Dignum, S., Kruschwitz, U., Fasli, M., Kim, Y., Song, D., Cervino, U., De Roeck, A.: Incorporating Seasonality into Search Suggestions Derived from Intranet Query Logs. In: Proceedings of WI 2010, Toronto, pp. 425–430 (2010)
8. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Intelligent Systems 1, 28–39 (2006)
9. Dou, Z., Song, R., Wen, J.-R.: A large-scale evaluation and analysis of personalized search strategies. In: Proceedings of WWW 2007, pp. 581–590. ACM, New York (2007)
10. Efthimiadis, E.N.: Query Expansion. In: Williams, M.E. (ed.) Annual Review of Information Systems and Technology (ARIST), vol. 31, pp. 121–187. Information Today (1996)
11. Fonseca, B.M., Golgher, P.B., de Moura, E.S., Ziviani, N.: Using association rules to discover search engines related queries. In: Proceedings of the First Latin American Web Congress, Santiago, Chile, pp. 66–71 (2003)
12. Hauff, C., Hiemstra, D., Azzopardi, L., de Jong, F.: A case for automatic system evaluation. 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. 153–165. Springer, Heidelberg (2010)
13. Jansen, J., Spink, A., Taksa, I. (eds.): Handbook of Research on Web Log Analysis. IGI (2008)
14. Joachims, T.: Evaluating retrieval performance using clickthrough data. In: Franke, J., Nakhaeizadeh, G., Renz, I. (eds.) Text Mining, pp. 79–96. Physica/Springer Verlag, Heidelberg (2003)
15. Jones, R., Rey, B., Madani, O.: Generating query substitutions. In: Proceedings of the 15th International World Wide Web Conference (WWW 2006), pp. 387–396 (2006)
16. Kelly, D., Gyllstrom, K., Bailey, E.W.: A comparison of query and term suggestion features for interactive searching. In: Proceedings of SIGIR 2009, Boston, pp. 371–378 (2009)
17. Lawrie, D., Croft, W.B.: Discovering and Comparing Topic Hierarchies. In: Proceedings of RIAO 2000, Paris, pp. 314–330 (2000)
18. Marchionini, G.: Human-information interaction research and development. Library and Information Science Research 30(3), 165–174 (2008)
19. Martens, D., De Backer, M., Vanthienen, J., Snoeck, M., Baesens, B.: Classification with Ant Colony Optimization. IEEE Transactions on Evolutionary Computation 11, 651–665 (2007)
20. Nanas, N., Kruschwitz, U., Albakour, M.-D., Fasli, M., Song, D., Kim, Y., Cervino, U., De Roeck, A.: A Methodology for Simulated Experiments in Interactive Search. In: Proceedings of the SIGIR 2010 SimInt Workshop, Geneva (2010)
21. Ruthven, I.: Interactive information retrieval. Annual Review of Information Science and Technology (ARIST) 42, 43–92 (2008)
22. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proceedings of SIGIR 1999, Berkeley, CA, pp. 206–213 (1999)
23. Silvestri, F.: Mining query logs: Turning search usage data into knowledge. Foundations and Trends in Information Retrieval 4, 1–174 (2010)
24. Soboroff, I., Nicholas, C., Cahan, P.: Ranking retrieval systems without relevance judgments. In: Proceedings of SIGIR 2001, New Orleans, pp. 66–73 (2001)

25. Socha, K., Sampels, M., Manfrin, M.: Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoWorkshops 2003. LNCS, vol. 2611, pp. 334–345. Springer, Heidelberg (2003)
26. Sowa, J.F.: Conceptual graphs. In: Handbook of Knowledge Representation. Foundations of Artificial Intelligence, ch. 5, pp. 213–237. Elsevier, Amsterdam (2008)
27. Tunkelang, D.J.: Faceted search. Morgan & Claypool Publishers (2009)
28. White, R.W., Bilenko, M., Cucerzan, S.: Studying the Use of Popular Destinations to Enhance Web Search Interaction. In: Proceedings of SIGIR 2007, Amsterdam, pp. 159–166 (2007)
29. White, R.W., Ruthven, I.: A Study of Interface Support Mechanisms for Interactive Information Retrieval. JASIST 57(7), 933–948 (2006)
30. Widdows, D., Dorow, B.: A Graph Model for Unsupervised Lexical Acquisition and Automatic Word-Sense Disambiguation. In: Proceedings of COLING 2002, Taipei, Taiwan, pp. 1093–1099 (2002)
31. Yuan, X., Belkin, N.J.: Supporting multiple information-seeking strategies in a single system framework. In: Proceedings of SIGIR 2007, Amsterdam, pp. 247–254 (2007)

Applying the User-over-Ranking Hypothesis to Query Formulation*

Matthias Hagen and Benno Stein

Faculty of Media
Bauhaus-Universität Weimar, Germany
{firstname.lastname}@uni-weimar.de

Abstract. The User-over-Ranking hypothesis states that the best retrieval performance can be achieved with queries returning about as many results as can be considered at user site [21]. We apply this hypothesis to Lee et al.'s problem of automatically formulating a single promising query from a given set of keywords [16]. Lee et al.'s original approach requires unrestricted access to the retrieval system's index and manual parameter tuning for each keyword set. Their approach is not applicable on larger scale, not to mention web search scenarios. By applying the User-over-Ranking hypothesis we overcome this restriction and present a fully automatic user-site heuristic for web query formulation from given keywords. Substantial performance gains of up to 60% runtime improvement over previous approaches for similar problems underpin the value of our approach.

1 Introduction

Experienced web search users come up with a whole set of keywords that describe their information need. But if the entire set is submitted as a single query, it is likely that only very few or even no results are returned. On the other hand, single-word queries can usually not satisfy intricate information needs, as such queries are not sufficiently specific. In practice, users then often strive for a longer and more specific query from their keywords, which finally leads to the desired result.

Lee et al. examined the corresponding problem of automatically formulating a single good query from given keywords [16]. Their approach selects the k best keywords according to a learnt ranking function and achieves good performance on TREC topics when given (1) the known relevant documents, (2) full access to an index of the document collection, and (3) a hand-tuned k for each topic—making the algorithm semi-automatic only. Their approach is not applicable in a standard web search scenario since search engines protect their proprietary indexes from direct access. Searches are only possible through interfaces and entail costs; at the very least some non-negligible amount of time is consumed, and larger contingents of queries may entail monetary charges.[1] We overcome the issues of Lee et al.'s approach with a fully-automatic external algorithm (i.e., working at user site against public web search interfaces) that

* Extended version of a paper presented at the TIR 2010 workshop [7].

[1] E.g., \$0.40–\$0.80 per 1000 Yahoo! BOSS queries, http://www.ysearchblog.com/2011/02/08/latest-on-boss/ (accessed April 16, 2011).

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 225–237, 2011.

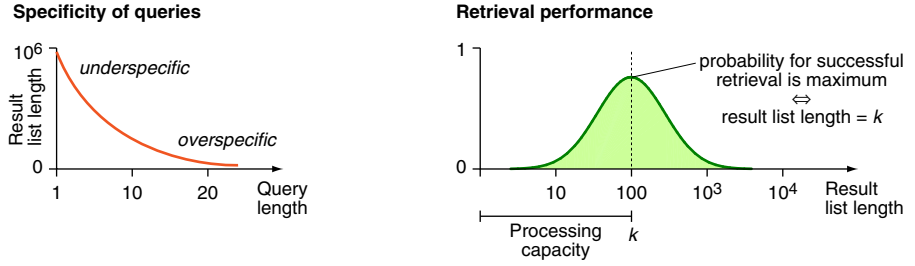


Fig. 1. Left: a query with few terms and many results is likely to be underspecific; queries with many terms and few results tend to be overspecific. Right: under the User-over-Ranking hypothesis a result list length of the user's capacity k maximizes the retrieval performance [21].

for given keywords finds a web query returning a reasonable number of results. The term *reasonable* deserves closer consideration. Typically, the number of search results a user will consider is constrained by a processing capacity $l_{\max}$, determined by the user's reading time, available processing time, etc. From queries returning more than $l_{\max}$ results only a fraction—typically the top-ranked results—are considered at user site. Often the search engine's ranking works reasonably well to bring up relevant documents to the top even for shorter queries. But what if the first $l_{\max}$ results of a short query don't satisfy the user's information need? The user can never be sure that there are no lower-ranked relevant documents and will try another query containing more or other keywords and thus being more descriptive of her information need. The User-over-Ranking hypothesis states that the user is best served by queries returning in the order of $l_{\max}$ results [21]. In this case, the user can avoid any ranking issues that she cannot influence by simply processing all the documents. The hypothesis is underpinned with a TREC-style experiment showing best performance for queries returning about 100 results [21]. See Figure 1 for an idealized illustration of the hypothesis.

If one follows the User-over-Ranking hypothesis, a central question is: Which subset of the keywords must be chosen to obtain about $l_{\max}$ results? This question is a natural web generalization of Lee et al.'s problem setting. But instead of analyzing the quality of keywords in isolation, the task now is to analyze keyword combinations in order to find queries that are sufficiently specific of a user's information need while not exceeding the user's capacity. The corresponding problem is formalized as follows.

PROMISING QUERY

Given: (1) A set W of keywords.
(2) A query interface for a search engine $\mathcal{S}$.
(3) Upper and lower bounds $l_{\max}$ and $l_{\min}$ on the result list length.

Task: Find a query $Q \subseteq W$ containing the most keywords possible that meets the result list constraints against $\mathcal{S}$.

The convenience lower bound $l_{\min}$ is introduced to rule out queries that return too few result (i.e., setting $l_{\min} = 1$ means that we do not tolerate queries that do not return any result). Note that the PROMISING QUERY formulation is a variant of Lee et al.'s initial setting: given a set W, find one good query from W. An important difference is

that PROMISING QUERY targets the real web setting, where users can only consider a small amount of results and do not have full access to the engine's index. Obviously, a series of web queries must be submitted to a search engine when solving PROMISING QUERY. As querying consumes time and may entail monetary costs too, we measure the costs at user site with the number of submitted queries and address the optimization problem of minimizing the average number of submitted queries.

1.1 Use Cases

Since today's search engines do not suggest promising subsets of a user's keywords, several use-cases can benefit from an external user-site approach. Note that our experimental evaluation in Section 4 is also based on these use cases.

Known Item Finding. Assume a user who once had access to a document on the web, forgot to save the document, and now comes up with some keywords that describe the document's content or that occur in the document. Re-finding the desired document has to be done via a web search engine and can be tackled by automatically constructing a good query from the user's set of keywords. Such a query should not return too many results since then the expectation is that the query is not descriptive enough to bring up the known item on the top of the result list. Furthermore, some of the remembered keywords might be "wrong" and should be omitted. Solving an instance of PROMISING QUERY provides a potential way-out.

Search Session Support. Assume a web search engine that recognizes sessions of consecutive web queries. Starting with some query, the user submits reformulated queries with varying keywords until she is satisfied or gives up. To support such a user, the engine itself can suggest a promising query returning a reasonable number of results from all the keywords submitted in the session [20]—an instance of PROMISING QUERY. According to the User-over-Ranking hypothesis [21], such promising query suggestions could improve the user's search experience in sessions that did not lead to the satisfaction of the user's information need.

Empty Result Lists. A search engine should avoid showing an empty result list on a user's web query. If a query does not produce any hit, an interesting option is to present a largest subset of the keywords that still give a reasonable number of results. According to the User-over-Ranking hypothesis [21], this will raise the probability of satisfying the user's initial information need. Solving an instance of PROMISING QUERY, the engine can provide an appropriate result list instead of an empty page.

1.2 Related Work

Besides Lee et al.'s keyword ranking, a lot of research has been done on approaches for better results on better queries. A promising idea is to estimate or to predict a given query's performance [5, 6, 9, 10, 14]. Especially the pre-retrieval predictors, which can be evaluated prior to the actual result retrieval phase, could be interesting for avoiding submission of too many queries. However, the evaluation of most predictors needs access to knowledge the user site does not have in a standard web search scenario. For example, the simplified query clarity predictor [10] needs the total keyword frequencies

for the whole corpus—web search engines just return an estimation of the number of documents in the corpus that contain the keyword. The query scope predictor [10] needs the number of documents in the index—most web search providers stopped publishing it. The mutual information based predictors [14] need the frequency of two keywords in a sliding window with given size over the whole corpus—no engine reports such values. Nevertheless, two approaches on reducing long queries successfully use query quality predictors [14, 17]—but with unrestricted access to the index. The task of long query reduction can be primarily described as the formulation of queries from verbose text, similar to the description parts of TREC topics or queries to medical search engines.

Reducing a large set of keywords to reasonable queries, as is the case in our setting, is also often termed as long query reduction. The interest in the issue of how to handle long queries is on the rise [1, 4, 12, 13, 15], as a significant part of todays typical web queries is becoming longer or "more verbose." In contrast to our setting, most of the existing research approaches assume full access to the system's index; only Huston and Croft use the notion of a black box search engine [11]. However, they focus on a scenario different from ours, namely: finding answers to verbose "wh-"questions in collaborative question answering systems, and they do not analyze the number of submitted web queries. Shapiro and Taksa explicitly deal with the problem setting of formulating queries while considering a bound on the number of results [19]. They suggest a rather simple "open end query formulation." Since their approach does not apply an exhaustive search, it is straightforward to construct situations in which promising queries exist, but the open end approach cannot produce even one of them.

1.3 Notation and Basic Definitions

Like in Lee et al.'s setting [16], our starting point for query formulation is a set $W = \{w_1, \ldots, w_n\}$ of keywords (phrases are also allowed). These keywords may be entered by a user or be generated automatically, by an automatic query expansion for example. Subsets $Q \subseteq W$ can be submitted as web queries, with the notion that phrases are included in quotation marks. A web search engine's reply to a query Q consists of a ranked list L_Q of snippets and URLs of documents relevant to the keywords from Q, along with an estimation l_Q for the real result list length $|L_Q|$.

Lee et al. try to identify the k "best" keywords in W. Their approach relies on the assumption that relevant documents for the information need described by W are known and that k can be manually determined for different W. Since this is unrealistic in web search scenarios, we combine the problem setting of automatic query formulation with the User-over-Ranking hypothesis [21]. Hence, our approach will select keywords to form a query that does not return too many results. Not the length of the query is the criterion, which is suggested by Lee et al., but the length of the result list: the PROMISING QUERY problem asks to find a largest subset $Q \subseteq W$ that satisfies $l_{\min} \leq l_Q \leq l_{\max}$ for given constant lower and upper bounds $l_{\min}$ and $l_{\max}$. Requiring Q to be as large as possible ensures Q to be as specific as possible, while the result list constraints reflect the user's capacity, this way accepting the User-over-Ranking hypothesis. Adopting the notation of Bar-Yossef and Gurevich [2], we say that for $l_Q < l_{\min}$ (too few results) the query Q is *underflowing*, whereas for $l_Q > l_{\max}$ (too many results) it is *overflowing*. Queries that are neither under- nor overflowing are *valid*. A valid query Q is *maxi-*

mal iff adding any keyword from $W \setminus Q$ results in an underflowing query. As for the PROMISING QUERY setting we are interested in the largest maximal queries.

In the process of finding a promising query Q, we count the overall number $cost$ of queries that are submitted to the search engine. Since a typical web query takes several hundred milliseconds, the time for internal client site computations will be clearly smaller than the time for submitting the web queries. An approach saving a significant number of web queries will dominate other approaches with respect to runtime, too.

In all query formulation algorithms of this paper, the result list length estimations l_Q of commercial search engines will be used, although they often overestimate the correct numbers. However, the estimations usually respect monotony (queries containing additional keywords have smaller l-value, indicating an AND-semantics at search engine site), and the shorter the result list, the better the estimations. Hence, in the range of our user constraints, where we require at most 100 results, they are pretty accurate.

2 Baseline Strategies for Promising Queries

The baseline approach can be described as a simple depth-first search on a search tree containing all possible queries; pseudo code listing given as Algorithm 1. A first pre-check removes underflowing keywords (first two lines of the listing) because they cannot be contained in a promising query. Such validity checks for queries always cause a submission to the search engine. A second pre-check (fourth line) ensures that the remaining set W of non-underflowing keywords itself is underflowing, since otherwise W itself is the promising query or no valid query can be found at all. Afterwards, Algorithm 1 invokes an exhaustive search such that it is guaranteed to find a promising query if one exists. Revisiting nodes in the search tree is prohibited by processing the keywords in the order of their indices. The algorithm starts trying to find a maximal valid query containing the first keyword w_1. It then adds the keywords w_2, w_3, etc., as long as the query remains non-underflowing. Whenever the query underflows, the last keyword is removed and the next one tried. If all keywords have been tried and the resulting query is valid, it is the current candidate to be a promising query. The algorithm then backtracks to other possible paths in the search tree. Pruning is done whenever the

Algorithm 1. Baseline algorithm for PROMISING QUERY

Input: keywords $W = \{w_1, \ldots, w_n\}$, result list bounds $l_{\min}$ and $l_{\max}$
Output: a largest valid query $Q_{\text{prom}} \subseteq W$

```
for all w ∈ W do
    if {w} is underflowing then W ← W \ {w}
Q_prom ← ∅
if W is underflowing then
    while (W ≠ ∅) ∧ (|W| > |Q_prom|) do
        w ← keyword with lowest index from W
        W ← W \ {w}
        ENLARGE({w}, W)
    output Q_prom
else output {W}
```

```
procedure ENLARGE(query Q, keywords W_left)
    while (W_left ≠ ∅) ∧ (|Q ∪ W_left| > |Q_prom|) do
        w ← keyword with lowest index from W_left
        W_left ← W_left \ {w}
        if Q ∪ {w} is overflowing or valid then
            Q' ← ENLARGE(Q ∪ {w}, W_left)
            if Q' is valid and |Q'| > |Q_prom| then
                Q_prom ← Q'
    return Q
```

current candidate cannot become larger than the currently stored promising query. A valid query that contains more keywords than the current promising query is stored as the new promising query.

Note that Algorithm 1 outputs the lexicographically first promising query with respect to the initial keyword ordering. Here *lexicographically* means the following. Let Q and Q' be two different queries and let $w_{\min}$ be the keyword with lowest index in the symmetric difference $Q \triangle Q' = (Q \cup Q') \setminus (Q \cap Q')$. Then Q comes lexicographically before Q' with respect to the keyword ordering $w_1, \ldots, w_n$ iff $w_{\min} \in Q$. Computing the lexicographically first promising query can be seen as a reasonable approach reflecting the idea that users in their queries first type the keywords that are most descriptive of their information need.

2.1 Computing All Promising Queries

Whenever the current candidate can only become as large as the current promising query, Algorithm 1 prunes the search. Instead, a slightly adapted version can check the current candidate enlarged by the full set W_{left} for validity. If this query is valid, then it forms an additional promising query of the same size as the current promising query and could be stored. If eventually on some other path a valid query is found that is larger than the current promising queries, the stored promising queries are deleted and the set of promising queries is initialized with the then found larger one. Note that this slight change in the baseline yields all promising queries for a given W. All of them could be presented to the user, or the lexicographically first one could be selected.

The described technique requires the submission of more queries to the engine. But only in pathological cases, which hardly occur in practice, this will significantly influence the overall performance. Experiments show that computing all promising queries in practice usually requires in the order of $|W|$ additional queries compared to computing just one promising query. These few additional queries can be a worthwhile investment in our heuristics (cf. the corresponding discussion in Section 3).

2.2 Co-occurrence-Informed Baseline

The main drawback of the above uninformed baseline is that it submits all intermediate query candidates to the search engine. To overcome this issue, we improve the approach by informing it with keyword co-occurrence probabilities. A pre-processing step determines $l_{\{w\}}$ for each $w \in W$ and $l_{\{w,w'\}}$ for each pair $w, w' \in W$. Using these values, we store the yield factors $\gamma(w, w') = l_{\{w,w'\}}/l_{\{w\}}$ in a non-symmetric matrix. The yield factor $\gamma(w, w')$ multiplied by $l_{\{w\}}$ gives the yield of web results when the keyword w' is added to the query $\{w\}$. We do not consider the queries needed for obtaining the yield factors. Our rationale is that in case of substantial savings achievable by using the yield factors, a promising future research task is to identify local "sandbox corpora" from which good approximations of web yield factors can be computed at zero cost (e.g., a local index of Wikipedia documents or the ClueWeb collection). Here we show the potential of our yield-factor-informed methods. In order to fairly treat the uninformed baseline, we don't count the baseline's submitted web queries with just one or two keywords in our experimentation.

The informed baseline uses the yield factors to internally estimate the number of returned results for a query candidate. The idea is to check validity without invoking the search engine and to directly enlarge query candidates with overflowing internal estimations.

Let the current query candidate be Q_{cand} and assume that all queries Q from previous computation steps already have a stored value est_Q indicating an estimation of the length of their result lists. Let w' be the last added keyword. Hence, the informed baseline already knows the value est_Q for $Q = Q_{\text{cand}} \setminus \{w'\}$. It then sets $est_{Q_{\text{cand}}} = est_Q \cdot \text{avg}\{\gamma(w, w') : w \in Q\}$, where avg denotes the mean value. During specific analyses we observed that $l_Q > est_Q$ for most queries Q (i.e., the internal estimations usually significantly underestimate the real number of search results). If this would always be the case, queries with overflowing internal estimations could directly be enlarged. However, our analyses also contained some very rare cases where Q is valid or underflowing but $est_Q > l_{\max}$ (i.e., even the tendency of the internal estimation is wrong). For this reason, the informed heuristic does not blindly follow the internal estimations but only trusts them when $est_{Q_{\text{cand}}} \geq adj \cdot l_{\max}$ for an adjustment factor adj. The rationale is that as long as the internal estimations are sufficiently above the validity bound $l_{\max}$, the probability for a wrong validity check based on the internal estimation is negligible. Only when the internal estimation $est_{Q_{\text{cand}}}$ is close to or below the validity bound $l_{\max}$, the current query is submitted to the search engine in order to "adjust" the internal estimation with the search engine's $l_{Q_{\text{cand}}}$. Larger values of adj enlarge the adjustment range and thus guarantee to catch more of the rare cases where Q is valid but $est_Q > l_{\max}$. However, this comes with a larger amount of submitted web queries. Moreover, only huge values of adj can guarantee to return the same promising query as the uninformed baseline. We performed an experimental analysis to compare different reasonable settings of $adj = 1, 3, 5$, and 10. The somehow surprising outcome of these experiments is that a value of $adj = 1$ shows a good overall conformity of the informed baseline's derived promising query with the uninformed baseline's promising query. As setting $adj = 1$ significantly reduces the query cost compared to larger values, the very few differences to the uninformed baseline's results are compensated by a significantly reduced $cost$ resulting in a challenging informed baseline for our heuristics.

3 Heuristic Search Strategies

Both the uninformed and the informed algorithms follow the scheme of Algorithm 1 and process the keywords in their initial ordering. We improve upon this ordering and suggest to use co-occurrence information not only to save some of the intermediate queries by internal estimations but also to adopt a heuristic search strategy with a potentially better keyword ordering. Compared to the baselines, the more involved order of processing will save a lot of queries (cf. the experiments in Section 4).

There are two points where a heuristic re-ordering strategy seems to be reasonable: the choices of using the keywords with lowest index as the next to-be-processed keyword (sixth line of Algorithm 1 and second line of procedure ENLARGE). We propose the following two yield-factor-informed re-ordering heuristics.

1. The first heuristic picks as the next keyword $w \in W$ in the sixth line of Algorithm 1 the one with the largest value $l_{\{w\}}$. The rationale is that this will be the keyword with the least commitment: We assume that w together with the next added keywords W' will result in a query $\{w\} \cup W'$ having larger l-value than the queries for any other remaining $w' \neq w$.

   In the ENLARGE procedure, the heuristic chooses as the best keyword $w \in W_{\text{left}}$ the one having the largest value $\text{avg}\{\gamma(w', w) : w' \in Q_{\text{cand}}\}$. Again, the heuristic's assumption is that this will be the keyword with least commitment (i.e., adding w to the current query candidate Q_{cand} will decrease the web count the least). Since the heuristic processes the keywords by descending l-value and descending yield factor, it is called the *descending heuristic*.
2. The second heuristic is the *ascending heuristic*, which reverses the descending heuristic's ordering. It picks as the next keyword $w \in W$ in the sixth line of Algorithm 1 the one with the smallest value $l_{\{w\}}$. In the ENLARGE procedure, the keyword $w \in W_{\text{left}}$ with the smallest value $\text{avg}\{\gamma(w', w) : w' \in Q_{\text{cand}}\}$ is chosen. The rationale for the ascending heuristic's approach can be best seen in a scenario where some keywords do not "fit" the others: keywords with very small l-value or very small $\text{avg}\{\gamma(w', w) : w' \in Q_{\text{cand}}\}$ are often not contained in a promising query. The ascending heuristic's ordering checks these keywords first and thus can weed out them early, while the descending heuristic would unsuccessfully (and costly) try to add them at the end of every search path.

Observe that due to the re-ordering of the keywords the first promising query found by either heuristic may not be the lexicographically first one that the uninformed or the informed baseline compute as their first promising query. This issue can be easily addressed as described in Section 2.1, namely by computing all promising queries and then selecting the lexicographically first among them. An argument for outputting the lexicographically first promising query is that this query probably contains the keywords that are most important for the user as she typed them earlier. Another option is to present all promising queries and let the user select.

4 Experimental Analysis

We experimentally compare our two heuristic search strategies to the two baselines. The experimental setting is chosen to reflect the different use cases described in Section 1.1.

4.1 Known Item Finding

For the known item finding use case we utilized the corpus from our previous experiments [7]. We crawled a 775 document collection consisting of papers on computer science from major conferences and journals. From each such document—the known items to be found—a number of keywords is extracted by a head noun extractor [3]. We set the bounds $l_{\max} = 100$ (following the findings of the User-over-Ranking hypothesis) and $l_{\min} = 10$. For each document of the test collection we had runs of the baselines and our heuristics with $3, 4, \ldots, 15$ extracted keywords against the Bing API from October 11–23, 2010. A typical web query against the API took about 200–500ms.

Table 1. Experimental results on the known item use case

	Number of keywords								
	5	7	9	10	11	12	13	14	15
Number of documents where									
Promising query not possible	614	481	338	328	219	155	117	100	86
Promising query found	161	294	437	447	556	620	658	675	689
Average cost (number of submitted queries)									
Ascending heuristic	10.39	15.71	21.94	24.93	29.32	34.10	42.70	44.70	53.78
Descending heuristic	9.71	15.03	22.65	25.26	35.54	45.04	73.03	91.63	130.92
Informed baseline	10.36	16.13	24.19	27.01	36.82	47.33	71.90	70.41	108.78
Uninformed baseline	11.81	18.64	28.80	30.94	43.46	54.61	84.48	88.78	116.22
Average cost ratio (basis: uninformed baseline)									
Ascending heuristic	0.88	0.84	**0.76**	**0.81**	**0.67**	**0.62**	**0.51**	**0.50**	**0.46**
Descending heuristic	**0.82**	**0.81**	0.79	0.82	0.82	0.82	0.86	1.03	1.13
Informed baseline	0.88	0.87	0.84	0.87	0.85	0.87	0.85	0.79	0.94
Average size promising query									
Ascending heuristic	3.45	5.03	6.72	7.73	8.36	8.97	9.54	9.99	10.40
Descending heuristic	3.49	5.03	6.77	7.76	8.39	8.97	9.50	10.07	10.41
Informed baseline	3.50	5.02	6.79	7.83	8.38	8.98	9.53	10.07	10.55
Uninformed baseline	3.50	5.06	6.83	7.90	8.42	9.01	9.54	10.16	10.57
Average ratio of common result URLs (basis: uninformed baseline)									
Ascending heuristic	0.96	0.93	0.93	0.93	0.95	0.96	0.98	0.93	0.93
Descending heuristic	0.98	0.96	0.96	0.96	0.95	0.96	0.96	0.97	0.93
Informed baseline	0.99	0.98	0.98	0.98	0.99	0.98	0.98	0.97	0.98

Table 1 shows selected results. Especially for sets with few keywords, a promising query is often not possible, since the complete query containing all keywords is still overflowing. The table's statistics are computed for the documents for which a promising query is possible. On these remaining documents all four approaches always find a promising query. Furthermore, the known item—the source document from which the keywords were extracted—always is among the results returned by the search engine for the promising queries. This suggests that promising queries are a reasonable tool to support known item finding.

The average number *cost* of web queries submitted to solve PROMISING QUERY and the respective ratio of submitted queries compared to the uninformed baseline show that overall the ascending heuristic performs best (smaller *cost* and smaller ratio indicate better approaches). The ascending heuristic on average submits less than 54 queries saving more than 50% of the queries compared to the uninformed baseline. Note that the descending heuristic fails to save queries for sets of 14 or more keywords. A possible explanation is that among the extracted keywords a non-negligible part does not fit the rest in the sense that not all extracted keywords describe the same concept.

The quality of the heuristics' promising queries is comparable to the baselines as can be seen by comparing the average size of the generated promising queries and the overlap in the retrieved document URLs. The small differences compared to the uninformed baseline are due to some rare overestimations using the internal estimations, which

"hide" some of the queries the uninformed baseline finds (cf. the respective discussion on the adjustment factor setting at the very end of Section 2.2). However, intensive spot checks showed that the heuristics usually produce the same output as the uninformed baseline. This is also supported by the average ratio of common URLs retrieved compared to the uninformed baseline (always above 90%) indicating that the heuristics' results are comparable to the baseline's results.

For the time consumption of the algorithms (not reported in Table 1) we observed the expected behavior: the internal computation time of all approaches is always orders of magnitude lower than the time for web queries (a few milliseconds vs. several seconds or even minutes). Hence, the fastest approach always is the one that submits the fewest queries and the ratio of runtime savings is equivalent to the query savings.

4.2 Search Session Support

For the search session use case we utilized a corpus similar to our previous experiments [20]: sessions with at least two queries extracted from the AOL query log [18] using two session detection techniques. One is a temporal method with a 10 minute cutoff (two consecutive queries belong to a session iff they are submitted within 10 minutes) and the other is the cascading method [8] (two consecutive queries belong to a session iff the contained keywords overlap and a time constraint is fulfilled, or if the queries are semantically very similar). Stopwords were removed from the derived sessions and for each method a random sample of 1000 sessions containing i keywords for every $i \in \{4, 5, \ldots, 15\}$ was selected. As before, we set the bounds $l_{\max} = 100$ and $l_{\min} = 10$. For each session we had runs of the algorithms against the Bing API from September 27–October 13, 2010. A typical web query took about 300–600ms.

Table 2 contains the experimental results; the table's organization follows that of Table 1. Again, sessions with few keywords often do not allow for a promising query because the complete query containing all words is still overflowing. Such sessions are filtered out and the statistics are derived just for the remaining sessions. Note that on these remaining sessions all approaches always find a promising query.

The average number $cost$ of submitted web queries and the respective ratio over the uninformed baseline again show that overall the ascending heuristic performs best. The possible savings seem to converge to about 60% of the queries compared to the uninformed baseline. The descending heuristic does not really improve upon the baselines.

Similar to the known item experiments the quality of the heuristics' promising queries is comparable to the baselines as can be seen by comparing the average query size and the overlap in the retrieved document URLs. As for the time consumption of the algorithms we again observe the expected behavior: the internal computation time of all approaches is always orders of magnitude lower than the web query time. Hence, the fastest approach always is the one that submitted the fewest queries.

4.3 Empty Result Lists

For the use case of queries with empty result lists we sampled longer queries from the AOL query log [18]. The log contains 1 015 865 distinct queries with at least 5 and at most 30 keywords. From these, we sampled 497 queries without typos that returned

Table 2. Experimental results on the search session use case

	Number of keywords								
	5	7	9	10	11	12	13	14	15
Number of sessions where									
Promising query not possible	1939	1868	1796	1719	1671	1543	1387	1167	903
Promising query found	61	132	204	281	329	457	613	833	1097
Average cost (number of submitted queries)									
Ascending heuristic	11.18	19.25	33.28	44.38	63.62	73.75	81.32	97.98	102.76
Descending heuristic	11.41	22.11	57.03	105.32	149.83	172.08	192.39	234.70	273.18
Informed baseline	11.90	22.35	45.07	63.40	89.74	107.11	117.02	145.83	167.92
Uninformed baseline	14.59	29.02	73.70	110.40	156.48	175.59	198.34	227.86	250.63
Average cost ratio (basis: uninformed baseline)									
Ascending heuristic	**0.77**	**0.66**	**0.45**	**0.40**	**0.41**	**0.42**	**0.41**	**0.43**	**0.41**
Descending heuristic	0.78	0.76	0.77	0.95	0.96	0.98	0.97	1.03	1.09
Informed baseline	0.82	0.77	0.61	0.57	0.57	0.61	0.59	0.64	0.67
Average size promising query									
Ascending heuristic	3.31	5.10	6.57	7.48	8.48	9.56	10.79	11.83	12.78
Descending heuristic	3.20	5.07	6.67	7.60	8.58	9.69	10.85	11.89	12.77
Informed baseline	3.02	4.95	6.42	7.34	8.34	9.51	10.70	11.75	12.69
Uninformed baseline	3.34	5.22	6.77	7.70	8.67	9.73	10.87	11.92	12.83
Average ratio of common result URLs (basis: uninformed baseline)									
Ascending heuristic	0.97	0.93	0.92	0.91	0.92	0.94	0.95	0.97	0.97
Descending heuristic	0.95	0.93	0.95	0.93	0.96	0.98	0.98	0.99	0.97
Informed baseline	0.87	0.91	0.89	0.86	0.91	0.93	0.93	0.95	0.96

less than 10 results as of October 27–30, 2010 using the Bing API. Empty result lists are modeled by a threshold of 10 returned results (instead of 0) as there exist several mirror pages of the complete AOL query log on the web. Thus, each AOL query with an empty result list back in 2006 today will return several such "mirror" results.

The average query length of the sample is 20.93 keywords (including stopwords). We removed stopwords obtaining an average query length of 12.47 keywords. As before, we set the bounds $l_{\max} = 100$ and $l_{\min} = 10$. Hence, a promising query is possible for all 497 queries (all return less than 10 results). For each query we had runs of the four algorithms against the Bing API from November 04–November 06, 2010. A typical web query in this experiment took about 250–550ms.

All four approaches always find a promising query. On average, the ascending heuristic submitted 92.37 queries, the descending heuristic submitted 207.37, the informed baseline 129.66, and the uninformed baseline 215.41. Hence, the ascending heuristic again obtains the best ratio over the uninformed baseline (about 0.43) and, as before, the ascending heuristic is the fastest among the four approaches. Also the ratio of common URLs (above 0.90 for all approaches) and the size of the promising queries again show that the heuristics' results are comparable to the baselines.

5 Conclusion and Outlook

We applied the User-over-Ranking hypothesis to Lee et al.'s query formulation problem and showed the effects of a user-oriented query cost analysis when formulating queries against a web search engine. In such situations a user plays against the engine in order to satisfy her information need by submitting keyword queries. Our formalization forms the ground for fully automatic and external algorithms that can be applied against the standard web search engine interfaces.

Altogether, the ascending heuristic should be preferred over the other methods, which is underpinned by experiments for three use cases. The ascending heuristic always comes with substantial savings in the number of submitted queries, whereas these savings do not impair the quality of the found promising queries. Compared to the uninformed baseline, only 40% of the queries have to be submitted for larger problem instances which is equivalent to a 60% reduced runtime.

A very interesting task for future work is to analyze the use of dedicated sandbox corpora from which yield factors resembling the ones obtained through web queries can be derived at zero cost.

References

[1] Balasubramanian, N., Kumaran, G., Carvalho, V.R.: Exploring reductions for long web queries. In: Proceedings of SIGIR 2010, pp. 571–578 (2010)
[2] Bar-Yossef, Z., Gurevich, M.: Random sampling from a search engine's index. Journal of the ACM 55(5) (2008)
[3] Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Proceedings of AI 2000, pp. 40–52 (2000)
[4] Bendersky, M., Croft, W.B.: Discovering key concepts in verbose queries. In: Proceedings of SIGIR 2008, pp. 491–498 (2008)
[5] Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: Proceedings of SIGIR 2006, pp. 390–397 (2006)
[6] Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proceedings of SIGIR 2002, pp. 299–306 (2002)
[7] Hagen, M., Stein, B.: Search strategies for keyword-based queries. In: Proceedings of DEXA 2010 Workshop TIR 2010, pp. 37–41 (2010)
[8] Hagen, M., Rüb, T., Stein, B.: Query session detection as a cascade. In: Proceedings of ECIR 2011 Workshop SIR 2011 (2011)
[9] Hauff, C., Hiemstra, D., de Jong, F.: A survey of pre-retrieval query performance predictors. In: Proceedings of CIKM 2008, pp. 1419–1420 (2008)
[10] He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 43–54. Springer, Heidelberg (2004)
[11] Huston, S., Croft, W.B.: Evaluating verbose query processing techniques. In: Proceedings of SIGIR 2010, pp. 291–298 (2010)
[12] Kumaran, G., Allan, J.: Adapting information retrieval systems to user queries. Information Processing and Management 44(6), 1838–1862 (2008)
[13] Kumaran, G., Allan, J.: Effective and efficient user interaction for long queries. In: Proceedings of SIGIR 2008, pp. 11–18 (2008)

[14] Kumaran, G., Carvalho, V.R.: Reducing long queries using query quality predictors. In: Proceedings of SIGIR 2009, pp. 564–571 (2009)
[15] Lease, M., Allan, J., Croft, W.B.: Regression rank: Learning to meet the opportunity of descriptive queries. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 90–101. Springer, Heidelberg (2009)
[16] Lee, C.-J., Chen, R.-C., Kao, S.-H., Cheng, P.-J.: A term dependency-based approach for query terms ranking. In: Proceedings of CIKM 2009, pp. 1267–1276 (2009)
[17] Luo, G., Tang, C., Yang, H., Wei, X.: MedSearch: a specialized search engine for medical information retrieval. In: Proceedings of CIKM 2008, pp. 143–152 (2008)
[18] Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Proceedings of Infoscale, paper. 1 (2006)
[19] Shapiro, J., Taksa, I.: Constructing web search queries from the user's information need expressed in a natural language. In: Proceedings of SAC 2003, pp. 1157–1162 (2003)
[20] Stein, B., Hagen, M.: Making the most of a web search session. In: Proceedings of WI-IAT 2010, pp. 90–97 (2010)
[21] Stein, B., Hagen, M.: Introducing the user-over-ranking hypothesis. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 503–509. Springer, Heidelberg (2011)

How to Count Thumb-Ups and Thumb-Downs: User-Rating Based Ranking of Items from an Axiomatic Perspective

Dell Zhang[1], Robert Mao[2], Haitao Li[3], and Joanne Mao[4]

[1] Birkbeck, University of London
Malet Street, London WC1E 7HX, UK
dell.z@ieee.org
[2] Microsoft Research
1 Microsoft Way, Redmond, WA 98052, USA
robmao@microsoft.com
[3] Microsoft Corporation
1 Microsoft Way, Redmond, WA 98052, USA
lht1999@gmail.com
[4] MNX Consulting, 9833 Wilden Lane, Potomac, MD 20854, USA
joanne.mao@mnxconsulting.com

Abstract. It is a common practice among Web 2.0 services to allow users to rate items on their sites. In this paper, we first point out the flaws of the popular methods for user-rating based ranking of items, and then argue that two well-known Information Retrieval (IR) techniques, namely the Probability Ranking Principle and Statistical Language Modelling, provide simple but effective solutions to this problem. Furthermore, we examine the existing and proposed methods in an axiomatic framework, and prove that only the score functions given by the Dirichlet Prior smoothing method as well as its special cases can satisfy both of the two axioms borrowed from economics.

1 Introduction

Suppose that you are building a Web 2.0 service which allows users to rate items (such as commercial-products, photos, videos, songs, news-reports, and answers-to-questions) on your site, you probably want to sort items according to their user-ratings so that stuff "liked" by users would be ranked higher than those "disliked". How should you do that? What is the best way to count such thumb-ups and thumb-downs? Although this problem — user-rating based ranking of items — looks easy and occurs in numerous applications, the right solution to it is actually not very obvious.

In this paper, we first point out the flaws of the popular methods for user-rating based ranking of items (see Section 3), and then argue that two well-known Information Retrieval (IR) techniques, namely the Probability Ranking Principle [1] and Statistical Language Modelling [2,3], provide simple but effective solutions to this problem (see Section 4). Furthermore, we examine the existing and proposed methods in an axiomatic framework, and prove that only the score functions given

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 238–249, 2011.

by the Dirichlet Prior smoothing [3] method as well as its special cases can satisfy both of the two axioms borrowed from economics, namely the Law of Increasing Total Utility and the Law of Diminishing Marginal Utility [4] (see Section 5).

2 Problem

Let's focus on binary rating systems first and then generalise to graded rating systems later. Given an item i, let $n_\uparrow(i)$ denote the number of thumb-ups and $n_\downarrow(i)$ denote the number of thumb-downs. In the rest of this paper, we shall omit the index i to simplify the notation when it is clear from the context that we are talking about an item i in general. To sort the relevant items based on user-ratings, a score function $s(n_\uparrow, n_\downarrow) \in \mathbb{R}$ would need to be calculated for each of them.

3 Popular Methods

There are currently three popular methods widely used in practice for this problem, each of which has some flaws.

3.1 Difference

The first method is to use the *difference* between the number of thumb-ups and the number of thumb-downs as the score function, i.e.,

$$s(n_\uparrow, n_\downarrow) = n_\uparrow - n_\downarrow \ . \tag{1}$$

For example, Urban Dictionary, a web-based dictionary of slang words and phrases, is said to be using this method, as shown in Figure 1.

Assume that item i has 200 thumb-ups and 100 thumb-downs, while item j has 1,200 thumb-ups and 1,000 thumb-downs, this method would rank item i (whose score is 100) lower than item j (whose score is 200). However, this does not sound plausible, because item i has twice thumb-ups than thumb-downs, while item j has only slightly more thumb-ups than thumb-downs.

3.2 Proportion

The second method is to use the *proportion* of thumb-ups in all user-ratings as the score function, i.e.,

$$s(n_\uparrow, n_\downarrow) = \frac{n_\uparrow}{n_\uparrow + n_\downarrow} \ . \tag{2}$$

For example, Amazon, the largest online retailer company in the United States, is said to be using this method, as shown in Figure 2.

Assume that item i has 200 thumb-ups and 1 thumb-down, while item j has 2 thumb-ups and 0 thumb-down, this method would rank item i (whose score is 0.995) lower than item j (whose score is 1.000). However, this does not sound plausible, because although both item i and item j have almost none thumb-down, item i has hundreds of thumb-ups, while item j has only a couple of thumb-ups.

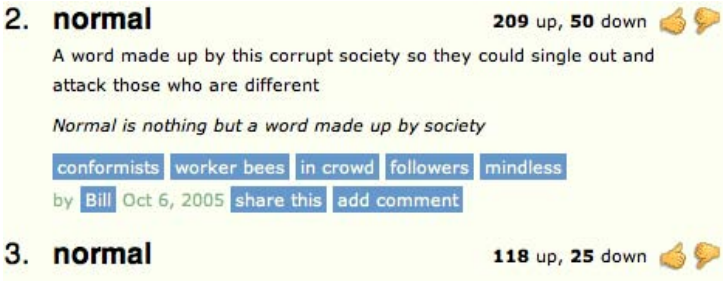


Fig. 1. An example of Urban Dictionary's ranking methods for user rated items, adapted from Evan Miller's online article[1]

3.3 Wilson Interval

The third method was advocated by Evan Miller's online article[1] on this topic to avoid the flaws of the above two simple methods. The idea is to treat the existing set of user-ratings as a statistical sampling of a hypothetical set of user-ratings from all users, and then use the *lower bound* of *Wilson score confidence interval* [5] for the proportion of thumb-ups as the score function, i.e.,

$$s(n_{\uparrow}, n_{\downarrow}) = \frac{\hat{p} + \frac{z^2_{1-\alpha/2}}{2n} - \sqrt{\frac{z^2_{1-\alpha/2}}{n}\left[\hat{p}(1-\hat{p}) + \frac{z^2_{1-\alpha/2}}{4n}\right]}}{1 + \frac{z^2_{1-\alpha/2}}{n}}, \tag{3}$$

Fig. 2. An example of Amazon's ranking methods for user rated items, adapted from Evan Miller's online article[1]

[1] http://www.evanmiller.org/how-not-to-sort-by-average-rating.html

where $n = n_\uparrow + n_\downarrow$ is the total number of user-ratings, $\hat{p} = n_\uparrow / n$ is the observed proportion of thumb-ups, and $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution. With the default parameter value $\alpha = 0.10$, the above score function estimates what the "real" proportion of thumb-ups at least is at 95% chance, therefore it balances the proportion of thumb-ups with the uncertainty due to a small number of observations. This method is considered as the current state of the art and thus adopted by many sites. For example, Reddit, a famous social news site, has mentioned in its official blog post[2] that this method is used for their ranking of comments, as shown in Figure 3.

Nevertheless, this method is not well justified either.

- First, the above formula cannot be applied to calculate scores for the items that have not received any user-rating yet: the prevailing implementation assigns score 0 to such items, which is wrong since this implies that "no user-rating yet" is roughly same as "zero thumb-up vs. one billion thumb-downs".
- Second, as the lower bound is biased towards one side only, it always underestimates the "real" proportion of thumb-ups.
- Third, it is not clear how tight the lower bound is, i.e., how far it deviates away from the "real" proportion of thumb-ups.
- Fourth, the difference between the lower bound and the "real" proportion of thumb-ups are inconsistent for items with different number of user-ratings.

Assume that item i has 1 thumb-up and 2 thumb-downs, while item j has 100 thumb-ups and 200 thumb-downs, this method would rank item i (whose score is 0.386) lower than item j (whose score is 0.575). However, this does not sound plausible, because while we are not really sure whether item i is good or bad, we have a lot of evidence that item j is bad, so we should rank item i higher than item j. For another example, using this method, we have $s(500, 501) > s(5, 1)$, i.e., an item with 500 thumb-ups and 501 thumb-downs would be ranked higher than an item with 5 thumb-ups and one thumb-down, which does not make much sense.

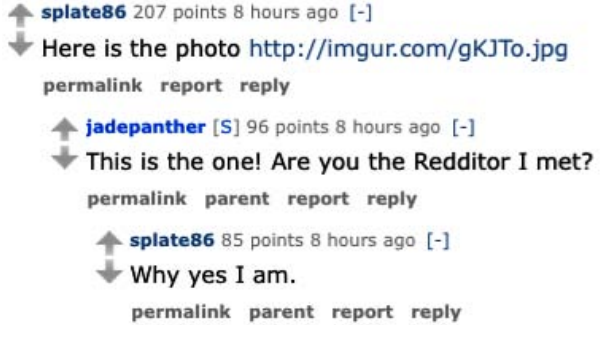


Fig. 3. An example of Reddit's ranking methods for user rated items, extracted from Reddit's blog post[2]

[2] http://blog.reddit.com/2009/10/reddits-new-comment-sorting-system.html

4 Proposed Approach

In this paper, we propose to address the problem of user-rating based ranking of items by formulating it as an extremely simple Information Retrieval (IR) system: each user-rating — thumb-up or thumb-down — is considered as a *term*; each item is considered as a *document* that consists of a number of those two terms. Since users would like to find good items from the collection, the ranking of the items could be regarded as searching the collection with a virtual *query* of one term — thumb-up ($q = \uparrow$). The better ratings an item has received from users, the more *relevant* it is to the query thumb-up.

According to the Probability Ranking Principle [1], we should rank documents by their probabilities of being relevant to the query, in our case, $\Pr[R = 1|i, \uparrow]$. This has been strictly proved to be the optimal retrieval strategy, in the sense that it minimises the expected loss (a.k.a. the Bayes risk) under 1/0 loss (i.e., you lose a point for either returning a non-relevant document or failing to return a relevant document) [6].

Making use of the Statistical Language Modelling [2,3] technique for retrieval, we treat each item i as a bag of user-ratings and construct a *unigram* model $M(i)$ for it, then the probability of an item being good (i.e., relevant to the query thumb-up) $\Pr[R = 1|i, \uparrow]$ can be calculated as the probability of the query being generated from its corresponding unigram model: $\Pr[\uparrow |M(i)]$.

So the problem becomes how we can accurately estimate the probability $\Pr[\uparrow |M(i)]$ for each item i. Given only a small number of observed user-ratings, the maximum likelihood estimator using the proportion of thumb-ups (i.e., the second method mentioned in Section 3) does not work due to the limitation of its frequentist view of probabilities, which is a well-known fact in the Information Retrieval community. For example, if item i has got 1 thumb-up and 0 thumb-down, the maximum likelihood estimator gives $\Pr[\uparrow |M(i)] = 1/(1 + 0) = 1$ and $\Pr[\downarrow |M(i)] = 0/(1 + 0) = 0$, which is apparently unreasonable — no thumb-downs so far does not mean that it is not possible to receive thumb-downs in the future, especially when we have seen one user-rating only. The solution is to *smooth* the maximum likelihood estimator so that we do not assign zero probability to unseen terms (user-ratings) and improve the accuracy of the estimated language model in general [7,8,3].

4.1 Additive Smoothing

Laplace Smoothing One of the simplest way to assign nonzero probabilities to unseen terms is Laplace smoothing (a.k.a. Laplace's rule of succession), which assumes that every item "by default" has 1 thumb-up and 1 thumb-down (known as pseudo-counts):

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow |M] = \frac{n_\uparrow + 1}{(n_\uparrow + 1) + (n_\downarrow + 1)} \ . \tag{4}$$

If item i has received 2 thumb-ups and 0 thumb-down from users, it would have 1+2=3 thumb-ups and 1+0=1 thumb-downs in total, so $\Pr[\uparrow |M(i)] =$

$3/(3+1) = 0.75$. If item j has got 100 thumb-ups and 1 thumb-down, it would have 100+1=101 thumb-ups and 1+1=2 thumb-downs in total, so $\Pr[\uparrow | M(j)] = 101/(101+2) = 0.98$. Thus we see that item j would be ranked higher than item i, which is indeed desirable.

Lidstone Smoothing. Although Laplace smoothing avoids most flaws of those popular methods (such as getting zero probability for unseen user-ratings), it probably puts too much weight on the pseudo-counts. A better choice is its more generalised form, Lidstone smoothing, which assumes that every item "by default" has ϵ thumb-ups and ϵ thumb-downs:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow | M] = \frac{n_\uparrow + \epsilon}{(n_\uparrow + \epsilon) + (n_\downarrow + \epsilon)} , \tag{5}$$

where $\epsilon > 0$ is a parameter. Previous research studies have shown that the performance of Lidstone Smoothing with $0 < \epsilon < 1$ is usually superior to $\epsilon = 1$ (i.e., Laplace Smoothing) [9].

4.2 Interpolation Smoothing

The above additive smoothing methods give all unseen user-ratings the same probability, which is not desirable if the user-ratings are generally imbalanced. A more reasonable smoothing strategy is to give different unseen user-ratings potentially different probabilities. This can be achieved by interpolating the maximum likelihood estimator of the item language model with a *background* language model M_b. Such a background language model can be specified a priori based on the domain knowledge. For example, in on-line shopping, users tend to be risk-averse so thumb-up should probably be given a lower probability than thumb-down in the background language model. More often, we may want to estimate the background language model based on the entire item catalogue. Suppose that there are totally N items in the catalogue. Let $p_\uparrow$ and $p_\downarrow$ denote the thumb-up probability and the thumb-down probability respectively in the background language model. Obviously $p_\downarrow = 1 - p_\uparrow$, so the background language model is determined as long as $p_\uparrow$ is known. There are two possible ways to estimate $p_\uparrow$ based on all the items $1, 2 \ldots, N$:

$$p_\uparrow = \Pr[\uparrow | M_b] = \frac{\sum_{i=1}^{N} n_\uparrow(i)}{\sum_{i=1}^{N} (n_\uparrow(i) + n_\downarrow(i))} , \tag{6}$$

$$p_\uparrow = \Pr[\uparrow | M_b] = \frac{1}{N} \sum_{i=1}^{N} \frac{n_\uparrow(i)}{n_\uparrow(i) + n_\downarrow(i)} . \tag{7}$$

Their difference is that in the former equation each user-rating contributes equally while in the latter equation each item contributes equally to the background language model. Which way is a better choice depends on which of these two assumptions is more sensible for the application domain.

Absolute Discounting Smoothing. The idea of this smoothing method is to lower the probability of seen user-ratings by subtracting a constant from their counts, and then interpolate it with the background language model:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow |M] = \frac{\max(n_\uparrow - \delta, 0)}{n_\uparrow + n_\downarrow} + \sigma p_\uparrow \ , \tag{8}$$

where $\delta \in [0, 1]$ is the discount constant parameter, and $\sigma = 1 - (\max(n_\uparrow - \delta, 0) + \max(n_\downarrow - \delta, 0))/n$ so that all probabilities sum up to one.

Jelinek-Mercer Smoothing. The idea of this smoothing method is to interpolate the maximum likelihood estimator of each document language model with the background language model using a fixed coefficient to control the amount of smoothing:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow |M] = (1 - \lambda)\frac{n_\uparrow}{n_\uparrow + n_\downarrow} + \lambda p_\uparrow \ , \tag{9}$$

where $\lambda \in [0, 1]$ is the fixed coefficient parameter.

Dirichlet Prior Smoothing. The idea of this smoothing method is to move from frequentist inference to Bayesian inference where probabilities are measures of uncertainty about an event. Before we see any user-rating for item i, we should have a prior belief about the probability for it to get thumb-ups which is given by $p_\uparrow$ from the background languagde model. After we see a user-rating for item i, we should revise or update our belief accordingly, i.e., increase $\Pr[\uparrow |M]$ when we see a thumb-up and decrease it otherwise. How much adjustment is appropriate depends on the probability distributions. Since there are only two random events (thumb-up or thumb-down), the natural choice is to model their occurrences as a binomial distribution for which the conjugate prior is a beta distribution. The beta distribution is the special case of the Dirichlet distribution with only two parameters. In order to keep the terminology consistent with the Information Retrieval literature, we call this Bayesian smoothing method Dirichlet Prior smoothing [3]. Such a prior essentially assumes that every item "by default" has $\mu p_\uparrow$ thumb-ups and $\mu p_\downarrow = \mu(1 - p_\uparrow)$ thumb-downs:

$$s(n_\uparrow, n_\downarrow) = \Pr[\uparrow |M] = \frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + \mu} \ , \tag{10}$$

where $\mu > 0$ is a parameter that determines the influence of our prior. Consequently, when we pool these pseudo-counts with the actual counts of user-ratings observed in the data, we would effectively interpolate the maximum-likelihood estimator of each item language model $M(i)$ with the background language model M_b using a dynamic coefficient that changes according to the number of user-ratings received so far: with more and more user-ratings available, the probabilities estimated using Drichlet Prior smoothing would be closer and closer to the maximum-likelihood estimator based on the observed data only.

4.3 Other Smoothing Techniques

There are many other smoothing techniques in Statistical Language Modelling, such as Good-Turing smoothing [7], but they do not seem to be suitable for our task because we only have two distinct "terms": thumb-ups and thumb-downs.

4.4 Generalisations

The proposed approach to ranking of items based on binary ratings (thumb-ups and thumb-downs) can be generalised to graded rating systems straightforwardly by taking each graded rating as multiple thumb-ups and thump-downs. Thus the "query" is still just one thumb-up, and each "document" (item) is still just a bag of thumb-ups and thumb-downs. For example, a 3-star rating in the 5-star scale system can simply be regarded as 3 thumb-ups and 5-3=2 thumb-downs. However, the semantic distance between 2-stars and 3-stars may be different from that between 3-stars and 4-stars. It is possible to take this into account by learning a real number of semantic thumb-ups for each graded rating from the user clickthrough data etc.

Furthermore, our approach can also be easily extended to take the ageing of user-ratings into account without affecting the computational efficiency through Time-Sensitive Language Modelling [10] techniques.

5 Axiomatic Examination

Which of the above mentioned ranking method, existing or proposed, is the best? To answer this question, we propose to examine their score functions in an axiomatic framework. The axioms that we use here are two fundamental principles in economics developed by Carl Menger [4] which nowadays are accepted as "irrefutably true" and widely used to interpret numerous economic phenomena.

Definition 1. *Given a score function s for user-rating based ranking of items, the* ***marginal utility*** *u of an additional thumb-up or thumb-down is the amount of difference that it can make to the score:*

$$\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) = s(n_{\uparrow}+1, n_{\downarrow}) - s(n_{\uparrow}, n_{\downarrow}) \ ,$$

$$\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) = s(n_{\uparrow}, n_{\downarrow}) - s(n_{\uparrow}, n_{\downarrow}+1) \ ,$$

where $n_{\uparrow}$ and $n_{\downarrow}$ are the current numbers of thumb-ups and thumb-downs respectively.

Axiom 1. ***The Law of Increasing Total Utility***
For any pair of non-negative integer numbers of thumb-ups and thumb-downs $n_{\uparrow}, n_{\downarrow} \in \mathbb{Z}^*$, a reasonable score function s must satisfy the following rules:

$$\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > 0 \ ,$$

$$\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) > 0 \ ,$$

which imply that each additional thumb-up or thumb-down should always make the score higher or lower correspondingly.

Axiom 2. ***The Law of Diminishing Marginal Utility***
For any pair of non-negative integer numbers of thumb-ups and thumb-downs $n_\uparrow, n_\downarrow \in \mathbb{Z}^*$*, a reasonable score function* s *must satisfy the following rules:*

$$\Delta_\uparrow^{(s)}(n_\uparrow, n_\downarrow) > \Delta_\uparrow^{(s)}(n_\uparrow + 1, n_\downarrow) \ ,$$

$$\Delta_\downarrow^{(s)}(n_\uparrow, n_\downarrow) > \Delta_\downarrow^{(s)}(n_\uparrow, n_\downarrow + 1) \ ,$$

which imply that the difference made by each additional thumb-up or thumb-down to the score should decrease as the number of thumb-ups or thumb-downs increases.

The above two axioms reflect our intuition about what a reasonable score function should be like.

Proposition 1. *The Difference method satisfies Axiom 1 but violates Axiom 2.*

Proposition 2. *The Proportion method violates both Axiom 1 and Axiom 2.*

Proposition 3. *The Absolute Discounting smoothing method violates both Axiom 1 and Axiom 2.*

Proposition 4. *The Jelinek-Mercer smoothing method violates both Axiom 1 and Axiom 2.*

It is relatively easy to show that the above propositions are true, by checking the score functions at the boundary condition $n_\downarrow = 0$, so their proofs are omitted.

Proposition 5. *The Wilson Interval method violates both Axiom 1 and Axiom 2.*

Proof. This can be shown by checking the score function (3) with $n_\uparrow = 1$.

It violates the Law of Increasing Total Utility, because along with the increase of $n_\downarrow$ the total score is not monotonically decreasing, as shown in Figure 4(a).

It violates the Law of Diminishing Marginal Utility, because along with the increase of $n_\downarrow$ the marginal utility is not decreasing but increasing, as shown in Figure 4(b). □

Theorem 1. *The Dirichlet Prior smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. The score function (10) obeys the Law of Increasing Total Utility because

$$\begin{aligned}
&\Delta_\uparrow^{(s)}(n_\uparrow, n_\downarrow) \\
&= s(n_\uparrow + 1, n_\downarrow) - s(n_\uparrow, n_\downarrow) \\
&= \frac{n_\uparrow + 1 + \mu p_\uparrow}{n_\uparrow + 1 + n_\downarrow + \mu} - \frac{n_\uparrow + \mu p_\uparrow}{n_\uparrow + n_\downarrow + \mu} \\
&= \frac{n_\downarrow + \mu(1 - p_\uparrow)}{(n_\uparrow + n_\downarrow + \mu)(n_\uparrow + n_\downarrow + \mu + 1)} \\
&> 0 \ ;
\end{aligned}$$

$$
\begin{aligned}
&\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) \\
&= s(n_{\uparrow}, n_{\downarrow}) - s(n_{\uparrow}, n_{\downarrow}+1) \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{n_{\uparrow} + n_{\downarrow} + \mu} - \frac{n_{\uparrow} + \mu p_{\uparrow}}{n_{\uparrow} + n_{\downarrow} + 1 + \mu} \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{(n_{\uparrow} + n_{\downarrow} + \mu)(n_{\uparrow} + n_{\downarrow} + \mu + 1)} \\
&> 0 \ .
\end{aligned}
$$

The score function (10) obeys the Law of Diminishing Marginal Utility, because

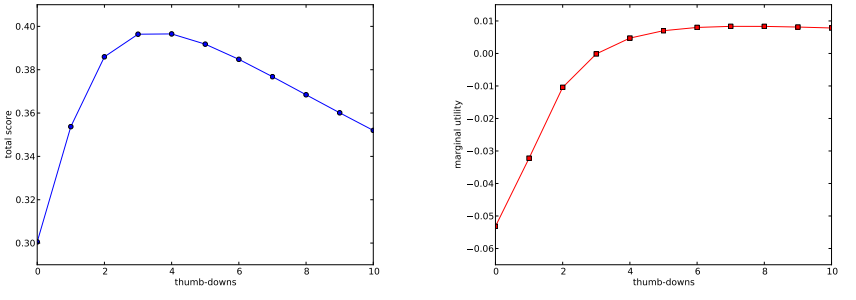
$$
\begin{aligned}
&\Delta_{\uparrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) - \Delta_{\uparrow}^{(s)}(n_{\uparrow}+1, n_{\downarrow}) \\
&= \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{(n_{\uparrow} + n_{\downarrow} + \mu)(n_{\uparrow} + n_{\downarrow} + \mu + 1)} - \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{(n_{\uparrow} + 1 + n_{\downarrow} + \mu)(n_{\uparrow} + 1 + n_{\downarrow} + \mu + 1)} \\
&= \frac{n_{\downarrow} + \mu(1-p_{\uparrow})}{n_{\uparrow} + n_{\downarrow} + \mu + 1} \left(\frac{1}{n_{\uparrow} + n_{\downarrow} + \mu} - \frac{1}{n_{\uparrow} + n_{\downarrow} + \mu + 2} \right) \\
&> 0 \ ;
\end{aligned}
$$

$$
\begin{aligned}
&\Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}) - \Delta_{\downarrow}^{(s)}(n_{\uparrow}, n_{\downarrow}+1) \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{(n_{\uparrow} + n_{\downarrow} + \mu)(n_{\uparrow} + n_{\downarrow} + \mu + 1)} - \frac{n_{\uparrow} + \mu p_{\uparrow}}{(n_{\uparrow} + n_{\downarrow} + 1 + \mu)(n_{\uparrow} + n_{\downarrow} + 1 + \mu + 1)} \\
&= \frac{n_{\uparrow} + \mu p_{\uparrow}}{n_{\uparrow} + n_{\downarrow} + \mu + 1} \left(\frac{1}{n_{\uparrow} + n_{\downarrow} + \mu} - \frac{1}{n_{\uparrow} + n_{\downarrow} + \mu + 2} \right) \\
&> 0 \ .
\end{aligned}
$$

□

Corollary 1. *The Laplace smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. It is because the Laplace smoothing method (4) is a special case of the Dirichlet Prior Smoothing method (10) with $\mu = 2$ and $p_{\uparrow} = 1/2$. □



(a) total score (b) marginal utility

Fig. 4. The Wilson interval $s(n_{\uparrow}, n_{\downarrow})$ with $n_{\uparrow} = 1$

Corollary 2. *The Lidstone smoothing method satisfies both Axiom 1 and Axiom 2.*

Proof. It is because the Lidstone smoothing method (5) is a special case of the Dirichlet Prior Smoothing method (10) with $\mu = 2\epsilon$ and $p_{\uparrow} = 1/2$. □

The axiomatic examination results about the existing and proposed ranking methods are summarised in Table 1. It is clear that only the score functions given by the Dirichlet Prior smoothing method as well as its special cases (Laplace smoothing and Lidstone smoothing) can satisfy both axioms borrowed from economics. Therefore the Dirichlet Prior smoothing method is our recommended solution for user-rating based ranking of items.

Table 1. The axiomatic examination results

	Increasing Total Utility	**Diminishing Marginal Utility**
Difference	Y	N
Proportion	N	N
Wilson Interval	N	N
Laplace smoothing	Y	Y
Lidstone smoothing	Y	Y
Absolute Discounting smoothing	N	N
Jelinek-Mercer smoothing	N	N
Dirichlet Prior smoothing	Y	Y

6 Conclusions

The main contribution of this paper is to show how the Information Retrieval techniques — Probability Ranking Principle and Statistical Language Modelling (with Dirichlet Prior smoothing) — can provide a *well justified* solution to the problem of user-rating based ranking of items in Web 2.0 applications.

The axiomatic approach to Information Retrieval has been studied by Bruza and Huibers [11], Fang and Zhai [12], and a few other researchers. To our knowledge, this paper is the first work that formulates user-rating based ranking of items as an Information Retrieval problem and examines the ranking methods for this problem from an axiomatic perspective.

References

1. Robertson, S.E.: The Probability Ranking Principle in IR. In: Readings in Information Retrieval, pp. 281–286. Morgan Kaufmann, San Francisco (1997)
2. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Melbourne, Australia, pp. 275–281 (1998)

3. Zhai, C.: Statistical Language Models for Information Retrieval. Morgan and Claypool (2008)
4. Menger, C.: Principles of Economics. New York University Press (1981)
5. Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association 22, 209–212 (1927)
6. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
7. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University (1998)
8. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), New Orleans, LA, USA, pp. 334–342 (2001)
9. Agrawal, R., Bayardo, R., Srikant, R.: Athena: Mining-based interactive management of text databases. In: Zaniolo, C., Grust, T., Scholl, M.H., Lockemann, P.C. (eds.) EDBT 2000. LNCS, vol. 1777, pp. 365–379. Springer, Heidelberg (2000)
10. Zhang, D., Lu, J., Mao, R., Nie, J.Y.: Time-sensitive language modelling for online term recurrence prediction. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 128–138. Springer, Heidelberg (2009)
11. Bruza, P., Huibers, T.W.C.: Investigating aboutness axioms using information fields. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland, pp. 112–121 (1994)
12. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Salvador, Brazil, pp. 480–487 (2005)

Aggregated Search Result Diversification

Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis

School of Computing Science,
University of Glasgow,
G12 8QQ, Glasgow, UK
{rodrygo,craigm,ounis}@dcs.gla.ac.uk

Abstract. Search result diversification has been effectively employed to tackle query ambiguity, particularly in the context of web search. However, ambiguity can manifest differently in different search verticals, with ambiguous queries spanning, e.g., multiple place names, content genres, or time periods. In this paper, we empirically investigate the need for diversity across four different verticals of a commercial search engine, including web, image, news, and product search. As a result, we introduce the problem of aggregated search result diversification as the task of satisfying multiple information needs across multiple search verticals. Moreover, we propose a probabilistic approach to tackle this problem, as a natural extension of state-of-the-art diversification approaches. Finally, we generalise standard diversity metrics, such as ERR-IA and α-nDCG, into a framework for evaluating diversity across multiple search verticals.

1 Introduction

Queries submitted to a web search engine are typically short and often ambiguous [28]. For instance, a user issuing the query '*amazon*' may be looking for the e-commerce company or the rainforest. Likewise, a user issuing a less ambiguous query such as '*amazon.com*' may be still interested in different aspects of this query, e.g., books, electronics, or digital music. To maximise the chance that different users will find at least one relevant search result to their particular information need, an effective approach is to diversify these results [13].

Existing diversification approaches have been deployed mostly in the context of web (e.g., [1,10,12,26,27]) and newswire (e.g., [6,9,32,33]) search, but there have also been approaches dedicated to diversifying image (e.g., [22,24]) and product (e.g., [19,31]) search results. Nevertheless, the nature of ambiguity can drastically vary across different search verticals. For instance, while query ambiguity arguably takes a more topical nature in a context such as web or image search, a news search query (e.g., '*olympics*') may give rise to temporal ambiguity (e.g., 2012? 2016?). In the same vein, a map search query (e.g., '*columbia*') may introduce geographical ambiguity (e.g., Maryland? Missouri? South Carolina?), while a blog search query (e.g., '*politics*') may entail social ambiguity (e.g., left-wing? neutral? right-wing?). Moreover, with the prevalence of aggregated search interfaces in modern web search [18,23], a search engine may be faced with the task of tackling query ambiguity spanning multiple search verticals.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 250–261, 2011.

In this paper, we introduce *aggregated search result diversification* as the problem of satisfying multiple possible information needs across multiple search verticals. To quantify the need for diversity in different search verticals, we investigate the nature of query ambiguity across four verticals of a commercial search engine, namely, web, image, news, and product search. Our investigation, based on queries from the TREC 2009 and 2010 Web tracks [10,12], shows that the ambiguity of a query varies considerably across different verticals, as do the likelihood of the different aspects underlying this query. Based upon this investigation, we propose a probabilistic approach for aggregated search result diversification. In particular, we extend state-of-the-art diversification approaches into a holistic approach to diversify the search results across multiple search verticals. Finally, we generalise standard diversity metrics into a framework for evaluating aggregated search result diversification. As a result, we extend the notion of whole-page relevance [3] to quantify the diversity of an entire result page.

Our major contributions are four-fold: (1) we introduce the problem of aggregated search result diversification; (2) we motivate this new problem through an empirical investigation using publicly available data from a commercial search engine; (3) we propose a probabilistic approach for tackling this problem; and (4) we introduce a general framework for evaluating approaches to this problem.

The remainder of this paper is organised as follows. Section 2 overviews related work in search result diversification and aggregated search. Section 3 investigates the nature of query ambiguity across four verticals of a commercial search engine, as a motivation for this work. Section 4 formalises the aggregated search result diversification problem. Section 5 describes our probabilistic approach for tackling the introduced problem. Section 6 proposes a framework for evaluating whole-page diversity. Lastly, Section 7 presents our conclusions.

2 Related Work

In this section, we describe diversification approaches that have been deployed in verticals such as web, newswire, image, and product search. We then review related work on aggregating results from multiple search verticals.

2.1 Search Result Diversification

The goal of search result diversification is to produce a ranking with maximum coverage and minimum redundancy with respect to the aspects underlying a query [13]. In recent years, several diversification approaches have been proposed, covering a range of search verticals such as web (e.g., [1,10,12,26,27]), newswire (e.g., [9,32,33]), image (e.g., [22,24]), and product (e.g., [19,31]) search.

In the context of web search, Agrawal et al. [1] proposed to diversify the search results with respect to a taxonomy of categories. Their approach focused on promoting search results with a high coverage of categories also covered by the query, but poorly covered by the other results. Rafiei et al. [26] proposed to favour search results that correlate lowly (in terms of content or received clicks) with the

other results, so as to promote novelty in the ranking. Santos et al. [27] proposed a probabilistic framework to rank the search results with respect to their coverage and novelty in light of multiple query aspects—represented by different query reformulations—as well as the relative importance of these aspects.

Newswire search result diversification was first tackled by Carbonell and Goldstein [6]. They proposed to rank the search results based on these results' estimated relevance to the query and their dissimilarity to the other results. Zhai et al. [33] extended this idea by comparing the search results in terms of the divergence of their language models, while Wang and Zhu [32] exploited relevance score correlations. Similarly, Chen and Karger [9] proposed to diversify the search results conditioned on the assumed irrelevance of the other results.

In the context of image search, van Leuken et al. [22] proposed to cluster the retrieved images using visual features, so that representative images from different clusters could form a diverse ranking. A similar approach was proposed by Deselaer et al. [15], however mixing both textual and visual features. In a different vein, Paramita et al. [24] proposed to diversify image search results spatially, by leveraging location information associated to every image.

Finally, in the context of product search, Vee et al. [31] proposed to diversify the search results for structured queries. From an initial ranking of products satisfying the query predicates, they devised tree-traversal algorithms to efficiently compare product pairs with respect to their attribute values. Similarly, Gollapudi and Sharma [19] deployed facility dispersion algorithms in order to promote diversity. Their approach compares the products retrieved for a query in terms of their categorical distance according to a given taxonomy.

2.2 Aggregated Search

Commercial web search engines often complement web search results with results from other search verticals, such as images, videos, and news articles [23]. As a modern instantiation of distributed information retrieval (DIR) [5], aggregated search involves the representation, selection, and aggregation of search results from multiple verticals. However, differently from traditional DIR problems, aggregated search deals with highly heterogeneous resources (i.e., search verticals) in a cooperative environment (i.e., verticals are usually run by the same company) and with abundance of usage data (e.g., vertical-specific query logs) [18]. Research in aggregated search has mostly focused on vertical selection, with fewer studies investigating the composition and evaluation of aggregated interfaces.

Vertical selection closely resembles resource selection in DIR [5]. While resource selection approaches typically focus on the contents of different resources (e.g., their size or their estimated number of relevant documents), modern vertical selection approaches leverage a wealth of available evidence from usage data. For instance, Diaz [16] proposed to predict the newsworthiness of web search queries by leveraging implicit user feedback (e.g., clicks and skips). Beitzel et al. [4] proposed a semi-supervised classification approach for automatically labelling queries with respect to 18 different verticals. A supervised approach was proposed by Arguello et al. [2] by exploiting evidence from vertical-specific query

logs and Wikipedia-induced vertical samples. Later, Diaz and Arguello [17] proposed to improve classification-based vertical selection by leveraging click data.

The composition of aggregated search interfaces was investigated by Ponnuswami et al. [25]. They improved click-through rates by learning to display results from already selected verticals in light of the displayed web search results. In terms of evaluation, Sushmita et al. [30] conducted a user study on factors affecting click-through rates on aggregated search. They observed a significant bias towards rank positions, but not towards any particular vertical. Finally, Bailey et al. [3] proposed a method for evaluating the relevance of a results page as a whole, as opposed to evaluating the relevance of individual search results.

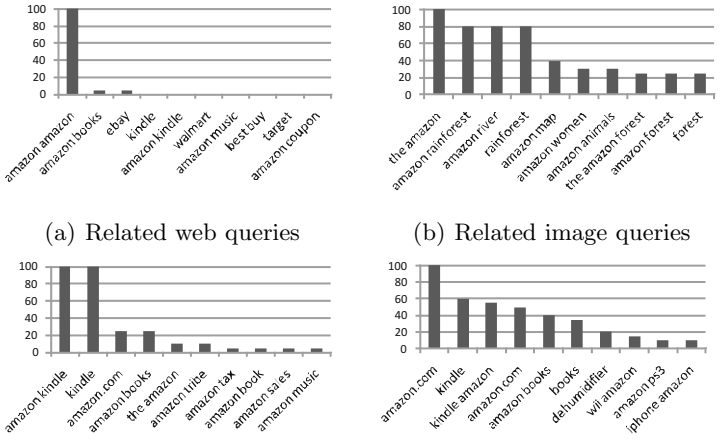
In the next section, we will bridge the gap between the search result diversification and the aggregated search problems, by investigating the nature of query ambiguity across multiple search verticals. This investigation will provide empirical motivation for the aggregated search result diversification problem, and the basis for modelling and evaluating approaches to this problem.

3 The Nature of Ambiguity

Aggregated search can be regarded as performing a surface-level diversification, in the sense that it tackles content-type ambiguity [14]. For instance, it is unclear whether a user issuing the query '*amazon*' to a web search engine would be satisfied with the e-commerce company's homepage (a standard web search result), its current stock performance (a financial search result), or the latest news about the company's recently announced music storage service (a news search result). Nevertheless, at the deeper level of individual verticals, the nature of the ambiguity of a single query can vary even further. To illustrate this observation, we use Google Insights for Search,[1] a service providing statistics of searchers' interest according to four Google verticals: web, image, news, and product search. For this particular study, we focus on related searches provided by each vertical for an input query, which can be regarded as representing the most likely information needs underlying this query in the context of a particular vertical. As an example, Fig. 1 shows the top 10 queries related to the query '*amazon*' in each of the four considered verticals, along with the normalised search volume associated to each of these queries. To ensure a uniform setting across the four considered verticals, we constrain our analysis to the US market. Additionally, we consider the total search volume generated within the period of January 2008 to March 2011, so as to attenuate seasonal or bursty interest fluctuations.

From Fig. 1(a), we observe that, in the web search vertical, the query '*amazon*' is likely to refer to some aspect of the e-commerce company, or similar companies related to it. However, in the image search vertical (Fig. 1(b)), the same query more likely refers to the Amazon rainforest. Likewise, this query may refer to the launch of new products and services or the discovery of a new tribe in the rainforest in the news vertical (Fig. 1(c)), or to the most popular products offered by the e-commerce company in the product vertical (Fig. 1(d)).

[1] http://www.google.com/insights/search



(a) Related web queries (b) Related image queries

(c) Related news queries (d) Related product queries

Fig. 1. Search volume for the queries most related to '*amazon*' in four verticals

To empirically quantify the nature of ambiguity across different verticals, we analyse the statistics provided by Google Insights for Search for all 100 queries from the TREC 2009 and 2010 Web tracks [10,12]. In particular, this set comprises queries of medium popularity sampled from the logs of a commercial search engine, hence providing a representative set of somewhat ambiguous yet not too popular web search queries. Limited by Google Insights for Search, we obtain up to the 50 most frequent queries related to each of the TREC Web track queries, along with their associated search volume, according to the aforementioned market and period constraints. Of the initial 100 queries, three do not have enough search volume in any of the four considered verticals, and are hence discarded from our analysis. Of the remaining 97 queries, 36 occur in only one vertical, 18 in two, 13 in three, and 30 queries occur in all four considered verticals. In cumulative terms, 61 ($\approx 63\%$) of the 97 considered ambiguous queries yield a non-negligible search volume in more than one vertical, which confirms the need to tackle query ambiguity spanning multiple search verticals.

To provide a consistent cross-vertical comparison, we further analyse the ambiguity of the 30 queries that occur in all four considered verticals. In particular, for each query q, let X be a categorical random variable with sample space $\mathcal{A}(q) = \{a_1, \ldots, a_k\}$, i.e., the set of all aspects underlying q, with each aspect represented by a query related to q, as obtained from all verticals. Likewise, let Y be a discrete random variable with sample space $\mathcal{V}(q) = \{v_1, \ldots, v_m\}$, i.e., the set of all verticals available for q. Lastly, let Z_v be a real-valued random variable with values $Z_v(a) = f_{X|Y}(X = a|Y = v)$, i.e., the frequency with which the aspect a is observed given that the vertical v was selected, as given by total search volume reported for the aspect a by the vertical v. We propose three metrics to contrast query ambiguity across the four considered verticals:

Ambiguity. The ambiguity of a query q quantifies the range of possible information needs underlying this query in light of a particular vertical v. We define it as the number of unique aspects related to q according to v:

$$ambiguity(q, v) = |\{a \in \mathcal{A}(q) : Z_v(a) > 0\}|. \quad (1)$$

Dominance. The definition of dominance complements our basic notion of ambiguity by showing how the interest for different information needs underlying a query is distributed. In particular, dominance quantifies the bias towards one or a few highly likely aspects of a query q in light of a particular vertical v. It is defined as the sample skewness g_1 [20] of the frequency distribution of the aspects related to q according to v:

$$dominance(q, v) = g_1(Z_v) = \frac{\sum_{a \in \mathcal{A}(q)} (Z_v(a) - \bar{Z}_v)^3}{(k-1)^3}, \quad (2)$$

where $\bar{Z}_v$ denotes the mean frequency over all $a \in \mathcal{A}(q)$ given v. A positive dominance indicates a bias towards frequent aspects, while a dominance approaching zero reveals a normal distribution around the mean frequency.

Agreement. The agreement of a pair of verticals with respect to a query quantifies the similarity of the distribution of information needs underlying this query across the two verticals. In other words, it measures the extent to which these verticals generate the same interest for the possible information needs underlying the query. We define the agreement between verticals v_i and v_j for a query q as the Czekanowski index[2] C [20] between the frequency distribution of aspects related to q according to v_i and v_j:

$$agreement(q, v_i, v_j) = C(Z_{v_i}, Z_{v_j}) = \frac{2 \sum_{a \in \mathcal{A}(q)} \min(Z_{v_i}(a), Z_{v_j}(a))}{\sum_{a \in \mathcal{A}(q)} (Z_{v_i}(a) + Z_{v_j}(a))}, \quad (3)$$

where the denominator performs a normalisation to enable the direct comparison of the agreement of different pairs of verticals. A value of one denotes total agreement, while a value of zero denotes total disagreement.

Table 1 shows the mean ambiguity, dominance, and agreement for the 30 TREC 2009 and 2010 Web track queries occurring in all four verticals, along with 95% confidence intervals for the means. From the table, we first note that, compared to news and product search, web and image search queries yield a significantly higher ambiguity. While the reason for this observation may be trivial (i.e., web and image search arguably receive a higher traffic), an important consequence is that diversification approaches should be aware of the varying number of possible aspects underlying the same query submitted to different verticals. In terms of dominance, all verticals show positive values, which indicate a moderate bias towards a few highly likely information needs. As an illustration (not shown in Table 1) of how a few aspects dominate the interest of the user population,

[2] For binary distributions, the Czekanowski index is equivalent to the Dice index.

Table 1. Mean ambiguity, dominance, and agreement across 30 TREC 2009 and 2010 Web track queries occurring in four Google verticals. A 95% confidence interval for the means according to the Student's t-distribution is also shown.

		web	image	news	product
ambiguity		45.567 $\pm$3.083	43.167 $\pm$4.198	21.233 $\pm$6.830	30.433 $\pm$7.114
dominance		6.295 $\pm$0.925	5.350 $\pm$0.720	7.741 $\pm$1.406	7.406 $\pm$1.276
		web	image	news	product
agreement	web	1.000 $\pm$0.000	0.372 $\pm$0.054	0.282 $\pm$0.060	0.285 $\pm$0.066
	image	–	1.000 $\pm$0.000	0.223 $\pm$0.065	0.204 $\pm$0.055
	news	–	–	1.000 $\pm$0.000	0.120 $\pm$0.041
	product	–	–	–	1.000 $\pm$0.000

to account for 70% of the search interest around all aspects of an ambiguous query, the web, image, news, and product verticals require, on average, only the top 35±3, 39±3, 48±10, and 46±8% most frequent aspects of this query, respectively. In absolute terms, the news search vertical shows the highest dominance, although not significantly higher than that of the other verticals. Finally, in terms of cross-vertical agreement, the highest non-trivial value observed is 0.372, when the web and image search verticals are compared. This observation quantitatively corroborates the illustrative example in Fig. 1, by showing that different verticals produce very dissimilar distributions of query aspects.

Overall, the results in this section highlight the specificities of query ambiguity in different search verticals, and the practical issues that must be considered when tackling it. Motivated by this investigation, in the next section, we formalise the problem of aggregated search result diversification.

4 Problem Formulation

Let $\mathcal{V}(q)$ denote the set of verticals v selected for a query q. Moreover, let $\mathcal{R}(q)$ denote the union of all search results r retrieved from these verticals. Finally, let $\mathcal{Q}(\cdot)$ denote the set of relevant aspects for a given input (a query or a result). For a rank cutoff $\tau > 0$, the goal of the aggregated search result diversification problem is to find a subset $\mathcal{S}(q) \subseteq \mathcal{R}(q)$, such that:

$$\mathcal{S}(q) = \underset{\mathcal{S}_i(q) \subseteq \mathcal{R}(q)}{\arg\max} \left| \cup_{\substack{v \in \mathcal{V}(q) \\ r \in \mathcal{S}_i(q)}} \mathcal{Q}(q|v) \cap \mathcal{Q}(r) \right|, \text{ s.t. } |\mathcal{S}_i(q)| \leq \tau. \tag{4}$$

From a search result diversification perspective, this formulation extends the diversification problem to account for query ambiguity across multiple search verticals. The key difference here is that the relevant aspects for a given query now depend on each individual vertical (i.e., $\mathcal{Q}(q|v)$), as motivated by our investigation in Section 3. From an aggregated search perspective, this formulation impacts the representation and selection of search verticals, which may benefit from accounting for the estimated diversity of the results provided by each vertical. Moreover, as we will show in Sections 5 and 6, this formulation impacts the criteria adopted for

Diversify$^{agg}(q, \mathcal{R}(q), \mathcal{V}(q), \tau)$
1 $\mathcal{S}(q) \leftarrow \emptyset$
2 **while** $|\mathcal{S}(q)| < \tau$ **do**
3 $r^* \leftarrow \arg\max_{r \in \mathcal{R}(q) \setminus \mathcal{S}(q)} f(r|q, \mathcal{S}(q), \mathcal{V}(q))$
4 $\mathcal{R}(q) \leftarrow \mathcal{R}(q) \setminus \{r^*\}$
5 $\mathcal{S}(q) \leftarrow \mathcal{S}(q) \cup \{r^*\}$
6 **end while**
7 **return** $\mathcal{S}(q)$

Alg. 1. Greedy aggregated diversification

aggregating results from multiple verticals and for evaluating this aggregation, as these results should ideally satisfy different information needs—as opposed to a single, precisely defined need—from different verticals.

5 Modelling Aggregated Diversification

By directly extending the basic diversification problem, the aggregated diversification problem also inherits its complexity. Indeed, both problems are instances of the maximum coverage problem, which is NP-hard [1]. Fortunately, there is a well-known greedy algorithm for this problem, which achieves a $(1 - 1/e)$-approximation. This is also the best possible worst-case approximation ratio achievable in polynomial time, unless NP $\subseteq$ DTIME$(n^{O(\log \log n)})$ [21].

In this section, we instantiate this greedy algorithm to tackle the aggregated diversification problem. In particular, Alg. 1 takes as input a query q, an initial ranking $\mathcal{R}(q)$ with $n = |\mathcal{R}(q)|$, a set of search verticals $\mathcal{V}(q)$, and an integer τ, with $0 < \tau \leq n$. It then iteratively constructs a re-ranking $\mathcal{S}(q)$, with $|\mathcal{S}(q)| \leq \tau$, by selecting, at each iteration, a search result $r \in \mathcal{R}(q) \setminus \mathcal{S}(q)$ that maximises the objective function f (line 3 in Alg. 1). This function evaluates a search result r given the query q, the results in $\mathcal{S}(q)$, selected in the previous iterations of the algorithm, and the considered verticals $\mathcal{V}(q)$. In this paper, we propose a probabilistic interpretation for the function f:

$$f(r|q, \mathcal{S}(q), \mathcal{V}(q)) \equiv \mathrm{P}(r|\mathcal{S}(q), q). \tag{5}$$

This formulation defines the diversity of a single result r as the probability of observing r conditioned on the observation of the query q and the already selected results in $\mathcal{S}(q)$.[3] In order to account for the available verticals, we marginalise the above probability over $\mathcal{V}(q)$ as a latent variable:

$$f \equiv \mathrm{P}(r|\mathcal{S}(q), q) = \sum_{v \in \mathcal{V}(q)} \mathrm{P}(v|q)\, \mathrm{P}(r|\mathcal{S}(q), q, v), \tag{6}$$

where $\mathrm{P}(v|q)$ is the probability of selecting the vertical v for the query q, and $\mathrm{P}(r|\mathcal{S}(q), q, v)$ denotes the probability of the search result r being relevant given

[3] Conditioning on $\mathcal{S}(q)$ is a typical mechanism for promoting novel results [9]—i.e., results different from those (assumed irrelevant) already in $\mathcal{S}(q)$.

the already selected results in $\mathcal{S}(q)$, the query q, and the vertical v. The latter probability is (in some form) at the core of most of the diversification approaches in the literature. In this work, we follow the state-of-the-art approaches [1,27], in order to explicitly account for the possible aspects underlying the query q in light of the vertical v. To do so, we further marginalise the probability $\mathrm{P}(r|\mathcal{S}(q), q, v)$ in Equation (6) over the set of aspects $\mathcal{A}(q|v)$ identified for q given v, as follows:

$$f \equiv \mathrm{P}(r|\mathcal{S}(q), q) = \sum_{v \in \mathcal{V}(q)} \mathrm{P}(v|q) \sum_{a \in \mathcal{A}(q|v)} \mathrm{P}(a|q, v)\, \mathrm{P}(r|\mathcal{S}(q), q, v, a), \qquad (7)$$

where $\mathrm{P}(a|q, v)$ denotes the likelihood of the aspect a given the query q and the vertical v, and $\mathrm{P}(r|\mathcal{S}(q), q, v, a)$ is the probability of the search result r being relevant given the already selected results in $\mathcal{S}(q)$, q, v, and a.

The problem is now reduced to estimating the various components in Equation 7. In particular, the set of verticals $\mathcal{V}(q)$ available for a query is normally fixed, while the probability $\mathrm{P}(v|q)$ can be estimated using any standard vertical selection approach, such as those described in Section 2.2. As for the set $\mathcal{A}(q|v)$ of query aspects identified from each vertical, as well as their likelihood $\mathrm{P}(a|q, v)$, one can deploy query log mining techniques to vertical-specific usage logs [27]. Alternatively, a taxonomy of categories appropriate to each individual vertical could be considered [1]. Finally, provided that the relevance estimation mechanism used by each considered vertical is available—which is the case in a typically cooperative aggregated search scenario—the probability $\mathrm{P}(r|\mathcal{S}(q), q, v, a)$ can be directly estimated by the state-of-the-art approaches in the literature [1,27].

Given the lack of a shared test collection for aggregated search evaluation, we leave the empirical validation of our proposed approach for the future. Such a collection could be constructed as part of a formal evaluation campaign (e.g., within the auspices of TREC) or as an individual or group effort (e.g., via crowdsourcing). Nevertheless, in the next section, we prepare the grounds for such an evaluation by proposing a suitable framework for this purpose.

6 Evaluating Aggregated Diversification

Traditional information retrieval evaluation metrics assume that the query unambiguously defines a user's information need. However, this assumption may not hold true in a real search scenario, when there is uncertainty regarding which aspect of the query the user is interested in [29]. Cascade metrics, such as ERR [8] and α-DCG [13], partially address this problem, by modelling a user who stops inspecting the result list as soon as a relevant result is found, hence rewarding novelty. To ensure that a high coverage of the possible aspects underlying the query is also rewarded, a possible solution is to extend existing metrics and compute their expected value given the likelihood of different aspects. This is precisely the idea behind the so-called *intent-aware* metrics for diversity evaluation [1]. In particular, given a ranking of documents $\mathcal{R}(q)$ and a set of relevant aspects $\mathcal{Q}(q)$ for a query q, a traditional evaluation metric $\mathit{Eval}(\mathcal{R}(q))$ can be cast into an intent-aware metric according to:

$$Eval\text{-IA} \equiv \sum_{a \in \mathcal{Q}(q)} \mathrm{P}^*(a|q)\, Eval(\mathcal{R}(q)|a), \tag{8}$$

where $\mathrm{P}^*(a|q)$ is the 'true' probability of observing a relevant aspect $a \in \mathcal{Q}(q)$, while $Eval(\mathcal{R}(q)|a)$ evaluates the ranking $\mathcal{R}(q)$ with respect to this aspect.

Despite being well established and validated [11], diversity metrics assume that a ranking of homogeneous search results is used as input. To cope with the presence of heterogeneous results (e.g., documents, images, videos, maps) in the increasingly prevalent aggregated search interfaces of modern search engines, we propose to generalise intent-aware metrics into a framework for evaluating diversity across multiple search verticals. In particular, we define an *aggregated intent-aware* (AIA) metric as the expected value of the corresponding intent-aware metric across multiple verticals, according to:

$$Eval\text{-AIA} \equiv \sum_{v \in \mathcal{V}(q)} \mathrm{P}^*(v|q) \sum_{a \in \mathcal{Q}(q|v)} \mathrm{P}^*(a|q,v)\, Eval(\mathcal{R}(q)|a,v), \tag{9}$$

where $\mathrm{P}^*(v|q)$ is the 'true' probability of observing the vertical v given the query q, $\mathrm{P}^*(a|q,v)$ is the 'true' probability of observing a relevant aspect $a \in \mathcal{Q}(q|v)$, and $Eval(\mathcal{R}(q)|a,v)$ now evaluates the ranking $\mathcal{R}(q)$ with respect to each vertical v and each aspect a identified in light of v. This formulation provides a framework for leveraging a wealth of existing evaluation metrics in order to synthesise the relevance and diversity of a whole page of results [3]. As concrete instantiations of Equation (9), we introduce aggregated intent-aware versions of the most widely used metrics for diversity evaluation, namely, ERR-IA [8] and α-DCG [13]:

$$\text{ERR-AIA} \equiv \sum_{v \in \mathcal{V}(q)} \mathrm{P}^*(v|q) \sum_{a \in \mathcal{Q}(q|v)} \mathrm{P}^*(a|q,v)\, \mathrm{ERR}(\mathcal{R}(q)|a,v), \tag{10}$$

$$\alpha\text{-DCG-AIA} \equiv \sum_{v \in \mathcal{V}(q)} \mathrm{P}^*(v|q) \sum_{a \in \mathcal{Q}(q|v)} \mathrm{P}^*(a|q,v)\, \alpha\text{-DCG}(\mathcal{R}(q)|a,v). \tag{11}$$

Note that the normalised version of both ERR-AIA and α-DCG-AIA (i.e., nERR-AIA and α-nDCG-AIA, respectively) requires producing an optimal re-ranking of $\mathcal{R}(q)$, which is an NP-hard problem, as discussed in Section 5. Nevertheless, the greedy approach in Alg. 1 can be used for this purpose, without noticeable loss in practice [7]. Also note that Equations (10) and (11) only penalise redundancy within each individual vertical, but not across multiple verticals. In practice, we assume that similar results of the same type (e.g., two videos about the same event) may be redundant, but similar results of different types (e.g., a video and a news story covering the same event) may be actually complementary.

In order to produce a realistic test collection for aggregated search result diversification, 'true' estimations of the likelihood of verticals could be derived from a large sample of the query logs of an aggregated search engine, while the likelihood of different aspects could be estimated from the logs of individual verticals. Lastly, the relevance of a search result could be judged with respect to the 'true' aspects

identified from the vertical providing this result. Besides enabling the evaluation of this newly proposed problem, such a collection would benefit ongoing research on both search result diversification and aggregated search.

7 Conclusions

We have proposed the aggregated search result diversification problem, with the aim of satisfying multiple information needs across multiple search verticals. To empirically motivate this new problem, we have analysed the ambiguity of real search queries submitted to four different verticals of a commercial search engine. Our results support the need for aggregated diversification, by showing that the nature of query ambiguity varies considerably across different verticals. Moreover, we have proposed a probabilistic approach for aggregated diversification, by extending current state-of-the-art diversification approaches to tackle query ambiguity in multiple search verticals. Lastly, we have proposed an evaluation framework for this new problem, by generalising existing metrics.

By laying the foundations of aggregated search result diversification, we have bridged current research in the vigorous fields of search result diversification and aggregated search. Nevertheless, we believe we have only scratched the surface of a very promising new field. With the availability of suitable shared test collections, this work can be further expanded in several directions, encompassing alternatives for modelling and evaluating approaches to this new problem.

References

1. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM, pp. 5–14 (2009)
2. Arguello, J., Diaz, F., Callan, J., Crespo, J.F.: Sources of evidence for vertical selection. In: SIGIR, pp. 315–322 (2009)
3. Bailey, P., Craswell, N., White, R.W., Chen, L., Satyanarayana, A., Tahaghoghi, S.: Evaluating whole-page relevance. In: SIGIR, pp. 767–768 (2010)
4. 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 Trans. Inf. Syst. 25(9) (2007)
5. Callan, J.: Distributed information retrieval. In: Croft, W.B. (ed.) Advances in Information Retrieval, ch. 5, pp. 127–150. Kluwer Academic Publishers, Dordrecht (2000)
6. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR, pp. 335–336 (1998)
7. Carterette, B.: An analysis of NP-completeness in novelty and diversity ranking. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 200–211. Springer, Heidelberg (2009)
8. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: CIKM, pp. 621–630 (2009)
9. Chen, H., Karger, D.R.: Less is more: probabilistic models for retrieving fewer relevant documents. In: SIGIR, pp. 429–436 (2006)
10. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 Web track. In: TREC (2009)

11. Clarke, C.L.A., Craswell, N., Soboroff, I., Ashkan, A.: A comparative analysis of cascade measures for novelty and diversity. In: WSDM, pp. 75–84 (2011)
12. Clarke, C.L.A., Craswell, N., Soboroff, I., Cormack, G.V.: Overview of the TREC 2010 Web track. In: TREC (2010)
13. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR, pp. 659–666 (2008)
14. Damak, F., Kopliku, A., Pinel-Sauvagnat, K., Boughanem, M.: A user study to evaluate the utility of verticality and diversity in aggregated search. Tech. Rep. 2, IRIT (2010)
15. Deselaers, T., Gass, T., Dreuw, P., Ney, H.: Jointly optimising relevance and diversity in image retrieval. In: CIVR, pp. 1–8 (2009)
16. Diaz, F.: Integration of news content into web results. In: WSDM, pp. 182–191 (2009)
17. Diaz, F., Arguello, J.: Adaptation of offline vertical selection predictions in the presence of user feedback. In: SIGIR, pp. 323–330 (2009)
18. Diaz, F., Lalmas, M., Shokouhi, M.: From federated to aggregated search. In: SIGIR, p. 910 (2010)
19. Gollapudi, S., Sharma, A.: An axiomatic approach for result diversification. In: WWW, pp. 381–390 (2009)
20. Hand, D.J., Smyth, P., Mannila, H.: Principles of data mining. MIT Press, Cambridge (2001)
21. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Inf. Proc. Lett. 70(1), 39–45 (1999)
22. van Leuken, R.H., Garcia, L., Olivares, X., van Zwol, R.: Visual diversification of image search results. In: WWW, pp. 341–350 (2009)
23. Murdock, V., Lalmas, M.: Workshop on aggregated search. SIGIR Forum 42, 80–83 (2008)
24. Paramita, M.L., Tang, J., Sanderson, M.: Generic and spatial approaches to image search results diversification. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 603–610. Springer, Heidelberg (2009)
25. Ponnuswami, A.K., Pattabiraman, K., Wu, Q., Gilad-Bachrach, R., Kanungo, T.: On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals. In: WSDM, pp. 715–724 (2011)
26. Rafiei, D., Bharat, K., Shukla, A.: Diversifying Web search results. In: WWW, pp. 781–790 (2010)
27. Santos, R.L.T., Macdonald, C., Ounis, I.: Exploiting query reformulations for Web search result diversification. In: WWW, pp. 881–890 (2010)
28. Song, R., Luo, Z., Nie, J.Y., Yu, Y., Hon, H.W.: Identification of ambiguous queries in Web search. Inf. Process. Manage. 45(2), 216–229 (2009)
29. Spärck-Jones, K., Robertson, S.E., Sanderson, M.: Ambiguous requests: implications for retrieval tests, systems and theories. SIGIR Forum 41(2), 8–17 (2007)
30. Sushmita, S., Joho, H., Lalmas, M., Villa, R.: Factors affecting click-through behavior in aggregated search interfaces. In: CIKM, pp. 519–528 (2010)
31. Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., Yahia, S.A.: Efficient computation of diverse query results. In: ICDE, pp. 228–236 (2008)
32. Wang, J., Zhu, J.: Portfolio theory of information retrieval. In: SIGIR, pp. 115–122 (2009)
33. Zhai, C., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR, pp. 10–17 (2003)

Topical Categorization of Search Results Based on a Domain Ontology

Silvia Calegari, Fabio Farina*, and Gabriella Pasi

Department of Informatics, Systems and Communication (DISCo),
University of Milano-Bicocca,
v.le Sarca 336/14, 20126 Milano, Italy
{calegari,farina,pasi}@disco.unimib.it

Abstract. This paper presents an approach to the categorization of Web search results based on a domain ontology that represents specific long term user's interests. The idea is to leverage the information in the considered ontology to classify results related to queries formulated in the topical context represented by the ontology. To efficiently manage the knowledge represented in a domain ontology for a categorizaton task, we propose a methodology to convert any domain ontology into its granular (taxonomy like) representation. In this paper, both the categorization process based on granular ontologies, and some evaluations that show the effectiveness the approach are presented.

Keywords: Granular Ontologies, Categorization of Web Results.

1 Introduction

The research reported in this paper is related to the problem of categorizing the results produced by search engines in response to users' queries. This is a well know research problem that has been addressed by several authors; to the aim of search results categorization both unsupervised and supervised techniques have been proposed [4,14]. More recently, the use of knowledge resources such as the ODP and WordNet has been considered to the aim of the categorization task [15,20,9]. Also some ontologies-based approaches to text classification have been proposed [19,21,16,5,12,6]. The advantage of using external knowledge resources is to effectively address semantic issues (through an explicit representation of concepts and relations among them), which enhance the classification process. The research reported in this paper is related to this last approach (use of external knowledge resources), but from a slightly different perspective. While the knowledge resources typically used in the categorization task are usually general taxonomies, we address the use of domain specific ontologies. The rationale behind this choice is twofold. First, people often use search engines to formulate queries related to specific domains of knowledge (e.g. related to their professional activity, or to their hobbies), and this kind of queries witnesses long term users'

* Now at Consortium GARR, the Italian NREN.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 262–273, 2011.

interests. Second, the domains of knowledge for which domain ontologies are available and publicly accessible are increasing. Organizing the results produced by a search engine in response to a domain dependent query, gives users' the possibility to more easily identify the web pages relevant to their needs, in a more focused way then by adopting a domain independent classification technique.

Following the above research direction in this paper we propose an approach that leverages the information in a domain ontology to classify results related to queries formulated in the topical context represented by the considered ontology.

However, as also outlined in the literature, to define effective classification algorithms based on the use of an ontology is not an easy task, mainly due to the complexity of the formal representation of ontologies. To make it possible to efficiently and effectively manage the knowledge represented in a domain ontology in a categorizaton task, we adopt a methodology to convert any domain ontology into its granular (taxonomy like) representation [2]. A granular representation of an ontology is a simplified and more compact representation, where only the hierarchical relation is considered, and where the concepts are grouped on the basis of their common properties. Based on the hierarchical organization of granules in the granular representation of the considered ontology, the proposed categorization strategy is hierarchical and multi-label [10].

In [3] the rationale behind the proposed approach was introduced; in this paper we present an extended formalization of the approach, as well as its implementation and preliminary evaluations. To evaluate the effectiveness of the proposed approach we have compared the results produced by the categorization obtained based on the granular representation of the considered domain ontology with respect to the categorization based on the original ontology.

The paper is organized as follows: Section 2 shortly introduces the related research; in Section 3 the method to reduce an ontology into a granular representation is shortly described. Section 4 explains the proposed method, while in Section 5 the evaluations are presented.

2 Related Work

In [10] the authors present how ontologies can be used to classification purposes. The advantage of defining classification algorithms based on an ontology is that a training set is not needed, as the classification relies on the entities represented in the ontology as well as on their relations [7]. Performing text classification based on an ontology means to associate the textual documents with some corresponding concepts of the ontology. In [21,6,17,16,5,12] a domain ontology has been used to classify Web documents and news from CNN and China News Agencies. A Web document is represented by a set of weighted terms, whereas categories are represented by a set of concepts from the domain ontology. The assignment of a Web document to a category is based on the similarity score between the document and the category: the higher the score is, the more likely the document belongs to this category.

Instead of analyzing the content of Web documents (as in the above works), we are interested in approaches where search results are categorized (usually by

analyzing their title and snippet). In [15] a system that makes use of document classification techniques by the multinomial Naïve Bayes classifier to organize search results into a hierarchy of topics has been proposed. The hierarchical structure considered in the above paper is the one provided by the ODP[1] Web directory. The classification task based on the ODP implies the analysis of more than 100.000 categories. This problem has been faced in [9,20] where a two-stage algorithm is used based on a search stage and a classification stage, respectively. In the search stage a subset of categories from the considered Web directory is identified, which is related to the given Web document. Then the classification process is performed on a smaller set of categories by improving the performance of the proposed classification strategy with respect to consider the whole hierarchy. The method presented in [9] outperforms the one defined in [20]. The major difference between these approaches is related to the considered strategy of classifiers selection: trigram and language model in [20], and a naive Bayes combination of local and global information in [9], respectively.

Also the approach presented in this paper organizes Web results according to a taxonomy (generated by a domain ontology); by considering a domain ontology, the categorization process works on a small set of candidate categories, so the first phase to prune irrelevant categories is not necessary.

In [13] one of the pioneer works reported in the literature for organizing hierarchically the content of texts is presented for purposes of documents classification. This step is performed with some rules that allow to semi-automatically define a hierarchy of concepts from the analysis of texts.

Ontologies are a power tool for knowledge definition and representation, and in last years their use as an external support for assigning semantics to the terms is increased. In [11] a general purpose ontology, named YAGO [18], has been used to categorize each search result into one category (single label classification) of the ontology. As the use of a domain ontology offers a more accurate representation of the knowledge related to a specific domain, and as often users formulate queries in relation to specific topical contexts, in this paper we propose a multi-label classification process based on domain ontologies, as it will explained in Section 4.

3 Definition of a Topical Granular Taxonomy from a Domain Ontology

As outlined in the Introduction, the first step of the approach proposed in this paper consists in simplifying an ontology-based representation into a conceptual taxonomy. To this aim we adopt the methodology originally proposed in [2], and finalized at generating a granular representation of an ontology. In this section this approach is shortly described; for additional information see [2].

Let $\mathbf{O}$ be an ontology: $\mathbf{O} = \{\mathbf{E}, \mathbf{R}, \mathbf{F}, \mathbf{A}\}$ [8], where $\mathbf{E}$ is the set of entities (i.e., concepts plus instances), $\mathbf{R}$ is the set of relations (taxonomic and non-taxonomic), $\mathbf{F}$ is the set of functions, and $\mathbf{A}$ is the set of axioms. $\mathbf{P}$ is a subset of

[1] ODP: Open Directory Project, (http://dmoz.org)

$\mathbf{R}$, $\mathbf{P} \subset \mathbf{R}$, which defines the properties of entities. Each property $P \in \mathbf{P}$ of $\mathbf{E}$ is defined as $P : \mathbf{E} \mapsto P_{val}$, where P_{val} is the set of all the values assumed by P. For example, the property *color* on the *wine* concept is defined as $color : wine \mapsto \{red, white, rosé\}$. The notion of granular representation of $\mathbf{O}$ is formally defined as a pair: $\mathbf{G_O} = \{\mathcal{G}, \mathbf{R_{IS-A}}\}$, where $\mathcal{G}$ is a set of granules, and $\mathbf{R_{IS-A}}$ is the subsumption (IS-A) relation defined on the set of granules $\mathcal{G}$.

To generate a granular representation of an ontology means to group into granules the entities of $\mathbf{O}$ that share some properties. From $\mathbf{O}$ we only consider instances linked by the IS-A relation and the properties defined on them. Let $P_{IS-A} = \{Color, Sugar, Flavor, Body\}$ be a subset of properties in $\mathbf{P}$ defined on the entities linked by the IS-A relation in $\mathbf{O}$ (such entities belong to the subset $\mathbf{E}_{IS-A}$ of $\mathbf{E}$). Then for each instance $e \in \mathbf{E}_{IS-A}$ we consider the values assumed for each property $p \in P_{IS-A}$. F is the function that assigns to a pair (e, p) the value assumed by the instance e for the property p, i.e. $F(Marietta\ Zinfandel, Color) = Red$.

Formally, given two instances $e_1, e_2 \in \mathbf{E}_{IS-A}$, e_1 is similar to e_2 with respect to P_{IS-A}, iff

$$\frac{|\{p_j \in P_{IS-A} | F(e_1, p_j) = F(e_2, p_j)\}|}{|P_{IS-A}|} \geq \epsilon \tag{1}$$

where $\epsilon \in [0, 1]$.

Formula 1 says that e_1, e_2 are similar if they have at least $\epsilon \cdot |P_{IS-A}|$ properties with the same value. As a consequence, we can assign e_1 and e_2 to the same granule.

An Example. In this paragraph, a simple and illustrative example is given to show the application of the granulation process on a specific domain ontology. To this aim we consider the Wine Ontology defined by the Stanford University [2]. Let us focus on a small portion of the Wine ontology where the considered set of properties is $P_{IS-A} := \{Color, Sugar, Flavor, Body\}$, and the set of instances is $\mathbf{E}_{IS-A} = \{Mountadan\ Pinot\ Noir, Marietta\ Zinfandel, Lane\ Tanner\ Pinot\ Noir\}$. For each instance in $\mathbf{E}_{IS-A}$ Table 1 reports the values associated with the considered properties.

Table 1. A tabular representation of a subpart of the Wine Ontology

Instances	Color	Sugar	Flavor	Body
`Marietta Zinfandel`	Red	Dry	Strong	Medium
`Mountadan Pinot Noir`	Red	Dry	Moderate	Medium
`Lane Tanner Pinot Noir`	Red	Dry	Moderate	Light

The problem we address is how to define coarser granules based on the considered information. To this aim on Table 1 we identify the properties where the instances share more values. As all instances share the same values for the properties *Color* and *Sugar*, we select one of the two properties to perform the first granulation step. We consider *Color* to generate the first granular

[2] http://www.w3.org/2001/sw/WebOnt/guide-src/wine

level (this choice is arbitrary, as we could first select $Sugar$). At step 2 the set P_{IS-A} will be reduced at $P_{IS-A} := \{Sugar, Flavor, Body\}$ with $|P_{IS-A}| = 3$. By applying Formula 1, we obtain that two instances belong to the same granule having two properties (over three) with the same value. Thus, (see Table 1) we can define two granules, one for the instances $Marietta\,Zanfandel$ and $Mountadam\,Pinot\,Noir$, and the other one for the instances $Mountadam\,Pinot\,Noir$ and $Lane\,Tanner\,Pinot\,Noir$, respectively. Let us notice that the instance $Mountadam\,Pinot\,Noir$ belongs to both granules.

When the process of granular representation is completed, then a domain expert can assign a meaningful name to each granule, according to the properties values characterizing the definition of the granule itself. Figure 1 shows a graphical representation of this example, where the circles are the properties values and the squares are the instances. On the left hand side of Figure 1 the considered subpart of the ontology **O** is sketched. On the right hand side the corresponding granular representation $\mathbf{G_O}$ is sketched. As it may be noticed, a granular taxonomy has less nodes than the original ontology, with the consequence that a topical categorization of search results is faster and simpler. In fact, the knowledge expressed by a domain ontology can be complex, i.e. rich of several concepts and relations.

By applying the previously granulation strategy explained to the Wine ontology, the original 219 entities have been reduced to 162 granules.

It is important to notice that the generated granules have a different semantics with respect to the original entities as they group instances which

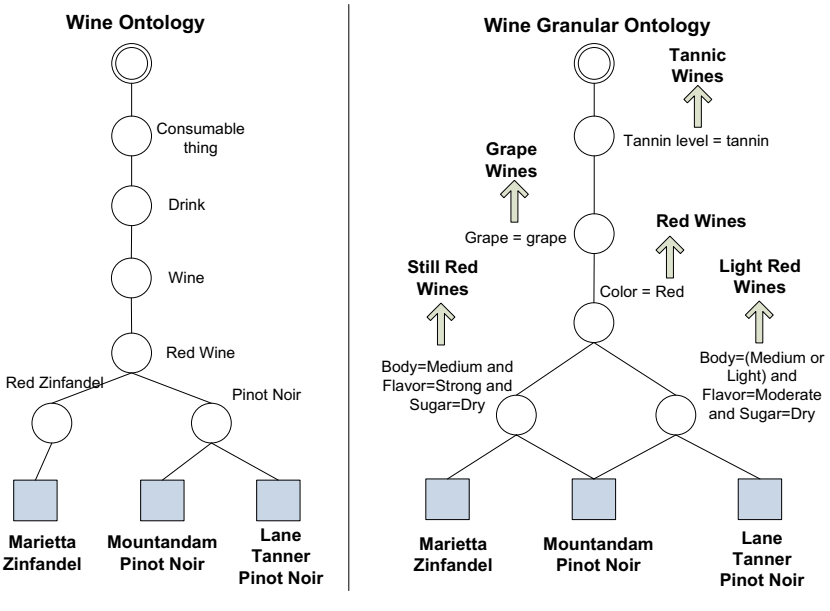


Fig. 1. On the left hand side the considered subpart of the Wine ontology **O** is sketched. On the right hand side the corresponding granular representation is sketched.

share some properties. Furthermore, $\mathbf{O_G}$ allows to discover new associations among instances not predictable a priori by analyzing $\mathbf{O}$, as it is happened for the *Mountandan Pinot Noir* wine that has similar features to the *Marietta Zinfandel* wine.

Two additional observations related to this example can be made: 1) the *red wine* node appears in the original taxonomy at a deeper level than in the granular one, and 2) the knowledge expressed in the granular taxonomy offers an effective synthesis of the domain knowledge.

4 The Proposed Method

In this section the method to categorize search results based on the granular representation of a domain ontology is described. The granular representation is first defined based on the method described in Section 3, then the proposed categorization method associates each search result with one or more topical granules.

Generally, in search engines the evaluation of a user's query produces a ranked list of results. The categorization of each search result is performed by locating in the granular representation of the ontology the appropriate granules with which it may be associated. Figure 2 shows the general structure of the approach where the results (left-hand side of Figure 2) are re-organized by the categorization method (right-hand side of Figure 2). The categorization process is described in Section 4.1.

4.1 The Categorization Process

Let $\mathbf{G_O}$ be the considered granular representation of an ontology $\mathbf{O}$, $\mathcal{G}$ the set of granules, and Res the set of search results. Each granule represents a meaningful concept related to the topical content of the ontology $\mathbf{O}$.

The association of each Web search result $R_i \in Res$ with one or more granules $g_k \in \mathcal{G}$ is performed in two steps:

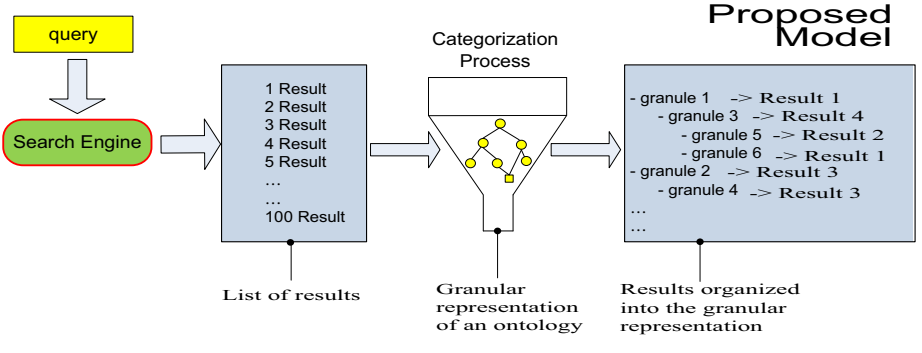


Fig. 2. The proposed model

- **Step 1:** "Search results conceptual indexing". To formally represent the topical content of a result R_i, we index its title and snippet via the controlled vocabulary represented by $\mathbf{G_O}$. We denote by $Rep(R_i)$ the set of granules extracted from title and snippet, and we define it as $Rep(R_i) := Title(R_i) \cup Snippet(R_i)$, where $Title(R_i)$ and $Snippet(R_i)$ are the set of granules extracted from title and snippet respectively, and belonging to the considered granular ontology. $Rep(R_i)$ is then a subset of information granules, i.e. $Rep(R_i) \subseteq \mathcal{G}$.
- **Step 2:** "Association of search results R_i with granule g_k". We denote by $Assoc(g_k)$ the search results associated with the granule g_k:

$$Assoc(g_k) = \{R_i | g_k \in Rep(R_i)\} \cup \left\{\cup_{g_c \in sub(g_k)} Assoc(g_c)\right\}$$

where $sub(g_k)$ is the set of the successors of g_k in the hierarchy, and consequently $\left\{\cup_{g_c \in sub(g_k)} Assoc(g_c)\right\}$ is the set of results associated with all the sub-granules (children nodes) of g_k.

A Simple Example. Let us consider the same vocabulary and structure of the Wine Ontology described in Section 3. The related set of granules is $\mathcal{G} :=$ {*Marietta Zinfandel, Mountadan Pinot Noir, Lane Tanner Pinot Noir, Red Wines, Tannic Wines, Grape Wines, Still Red Wines, Light Red Wines*}. During a search session a user is interested in finding, for instance, information about *red wines* and he/she writes the following short query q=*"red wines in*

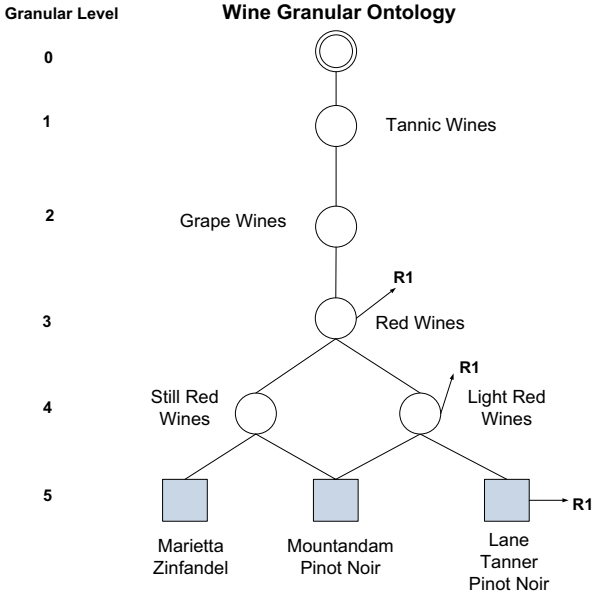


Fig. 3. Classification of the search result R_1 provided by the granular taxonomy

France", and a list of results is displayed. By analysing the first result R_1, we have: *Title="Wines of France-A guide to French wines"* and *Snippet="Discover the wines of France, their varieties, history and regions;... Lane Tanner Pinot Noir is a very famous red wine produced in..."*. From these two short texts, we index R_1 by the granules of the ontology. We obtain $Title(R_1) = \emptyset$ and $Snippet(R_1) = \{Lane\ Tanner\ Pinot\ Noir, Red\ Wine\}$. Thus, $Res(R_1) = \{Lane\ Tanner\ Pinot\ Noir, Red\ Wines\}$, i.e. $Rep(R_1) = Title(R_1) \cup Snippet(R_1)$. Figure 3 depicts the situation after the application of Step 2 where the result $\{R_1\}$ has been categorized into the granules g_1, g_3 and g_6. These phases are applied to the whole set of Web results *Res* obtained by the evaluation of a user query. Moreover, a granule (named *Unclassified*) has been added to the same level of the root node in order to classify all the results that are not associated with the considered set of granules $\mathcal{G}$. By considering the same example depicted in Figure 4, the cardinality associated with the *Unclassified* granule is equal to zero.

5 Preliminary Experiments

The presented approach has been implemented as a standalone service that interacts with the Yahoo! Search Engine, and returns to the user the classified results. It consists of two main components: (1) the search result categorization module, and (2) a user interface that presents the search results according to the category structure sketched in Figure 4. We have taken inspiration from *Clusty*[3] where the Web-page structure is split into three parts: 1) a query formulation box (according to the used search engine), 2) a graphical representation of the granular ontology, and 3) an area devoted to the visualization of search results. Figure 4 reports an example where the granular representation of the Wine Ontology is used to classify the obtained results produced by the evaluation of the query *red wines in France*.

To the aim of a preliminary evaluation of the approach, we have considered the Wine Ontology defined by the Stanford University[4], as well as a set of queries related to the wine topical interests. A granular representation of the Wine Ontology has been defined by the methodology reported in Section 3.

To measure the effectiveness of the proposed categorization strategy we have adopted different metrics: the precision and recall measures, and the agreement between the classification proposed by our model with the one provided by human experts for a given set of queries. In detail, we have asked to four wine experts to use our Yahoo! wrapper application for formulating 11 queries each, and for each query to analyze the produced results by identifying the more appropriated categories. We have compared the categorization process performed by the granular ontology with the original ontology (by only considering entities connected by the IS-A relation). A vocabulary defined on both set of granules ($\mathcal{G}$) and the entities ($\mathbf{E}_{IS-A}$) from the two taxonomies has been defined (where the granular taxonomy is made up of 162 granules, whereas the original ontology

[3] (http://clusty.com/)
[4] http://www.w3.org/2001/sw/WebOnt/guide-src/wine

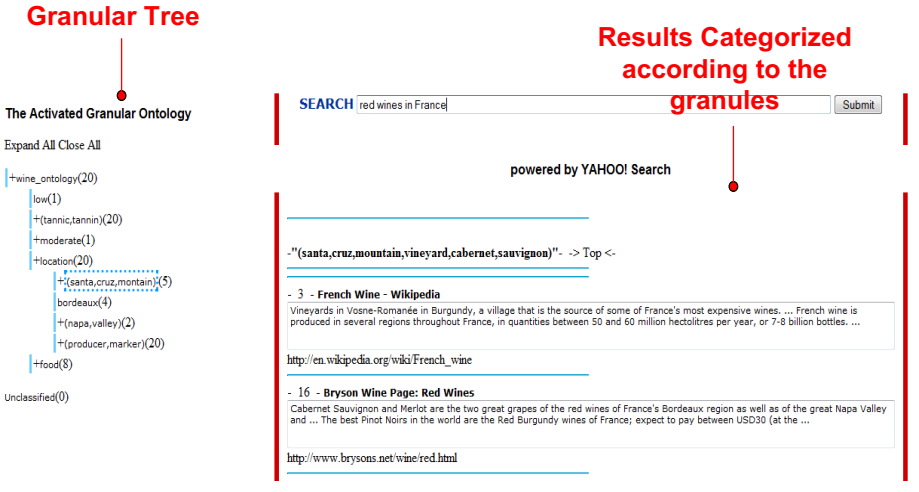


Fig. 4. Web pages returned by the proposed prototype implementation

is made up of 219 entities). The four experts decided which categories a search result belongs to by identifying the terms more appropriate from the vocabulary. In both cases a hierarchical multi-label classification has been performed.

To evaluate precision and recall we have used the definitions proposed in [11]:

$$Prec(q) := \sum_{g_k \in \mathcal{G}_q} \left(p(g_k) \cdot \frac{|Assoc_{ontology}(g_k)|}{|\bigcup_{g_j \in \mathcal{G}_q} Assoc_{ontology}(g_j)|} \right) \tag{2}$$

$$Rec(q) := \sum_{g_k \in \mathcal{G}_q} \left(r(g_k) \cdot \frac{|Assoc_{ontology}(g_k)|}{|\bigcup_{g_j \in \mathcal{G}_q} Assoc_{ontology}(g_j)|} \right) \tag{3}$$

where $\mathcal{G}_q$ is the set of granules that contain at least one search result for the query q, the functions $p(g_k)$ and $r(g_k)$ evaluate the precision and the recall of the granule g_k for the query q. These are defined as $p(g_k) := |Assoc_{ontology}(g_k) \cap Assoc_{expert}(g_k)|/|Assoc_{ontology}(g_k)|$ and

$$r(g_k) := |Assoc_{ontology}(g_k) \cap Assoc_{expert}(g_k)|/|Assoc_{expert}(g_k)|,$$

where $Assoc_{ontology}(g_k)$ is the set of results associated with the granule g_k according to the considered taxonomies and the expert's judgement, respectively. The values of $p(g_k)$ and $r(g_k)$ are weighted and normalized according to the cardinality of $Assoc_{ontology}(g_k)$ with respect to the total number of results associated with any other granule. Precision indicates the fraction of search results correctly categorized with respect to the classification provided by the human experts for each granule. Recall states the percentage of the number of search results correctly categorized in comparison to the total number of search results that should be categorized into this category according to the experts expectation.

The agreement is a measure that we adopt to evaluate how much the association obtained with the proposed approach overlaps with respect to an association provided by the domain expert for a specific query. The agreement focuses on the categories of the search results rather than the results themselves. Higher values of the measure indicate more common associations between the experts and the system. Formally the agreement on the results obtained for a query q is defined as:

$$Agr(q) := \left(\sum_{R_i \in Res_q} \frac{|C_{ontology}(R_i) \cap C_{expert}(R_i)|}{|C_{expert}(R_i)|} \right) / |Res_q| \tag{4}$$

where $C_\alpha(R_i) := \{g_j | R_i \in Assoc_\alpha(g_j), g_j \in \mathcal{G}_q\}$ is the set of the granules associated with a given Web result according to the criterion adopted either by experts or by the ontology.

For evaluations we have considered, for the 44 queries, the top 20 results at different cuts: @5, @10, @15 and @20 (as indicated in [1]). Figure 5 reports the average values for $Agr(q)$, $Prec(q)$ and $Rec(q)$ over all users and queries. The $Agr(q)$ values, always above 0.8, denote that our approach is in high accordance with the one proposed by experts. $Agr(q)$ for the granular ontology is almost twice than the one measure for the original taxonomy at every cut. This confirms that a good hierarchical classification can be obtained with the use of a simpler representation of a domain ontology.

For the same reason we observe that precision and recall are always higher for the granular ontology. The difference with respect to the agreement measure is that the agreement considers the superimposition of the granules involved by both experts and the methodology. Instead the precision takes into account the single

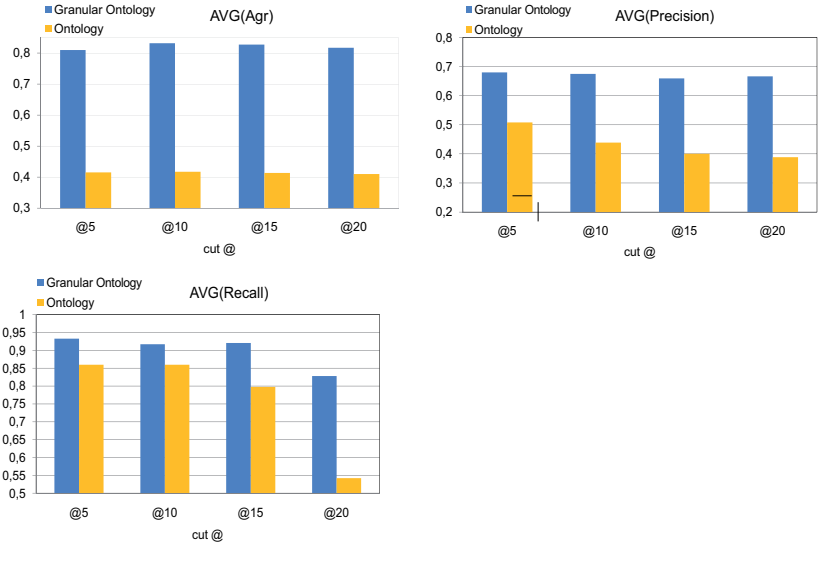


Fig. 5. Agreement, Precision and Recall for multi-label classification

search results. Higher precision with respect to the ontology denotes that the granular ontology is able to identify a large portion of the relevant results although the granular ontology is defined with less concepts.

6 Conclusions and Future Work

In this paper we have studied the problem of classification of search results when search refers to a specific domain represented by a granular representation of a domain ontology. The granular representation has been obtained by considering the methodology presented in [2], but in this work its formal extension has been proposed. The proposed method offers a semantic support to the categorization task in domain-dependent IR.

To evaluate the effectiveness of our approach we used different metrics, such as precision, recall and agreement measures. Given an ontology domain, we have compared the effectiveness of its granular representation with respect to the original taxonomic relation. The granular ontology exhibits a better behavior than the original ontology for all the considered measures.

In the prosecution of this research activity will evaluate our hierarchical approach with standard hierarchical classifiers (e.g., C4.5).

References

1. Ansen, B.J., Spink, A., Pedersen, J.: A temporal comparison of altavista web searching. Journal of the American Society for Information Science & Technology 56(6), 559–570 (2005)
2. Calegari, S., Ciucci, D.: Granular computing applied to ontologies. Int. J. Approx. Reasoning 51(4), 391–409 (2010)
3. Calegari, S., Pasi, G.: Gronto: A granular ontology for diversifying search results. In: Melucci, M., Mizzaro, S., Pasi, G. (eds.) IIR. CEUR Workshop Proceedings, vol. 560, pp. 59–63. CEUR-WS.org (2010)
4. Carpineto, C., Osinski, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Computing Surveys 41(3), 17:1–17:17 (2009)
5. Fang, J., Guo, L., Wang, X., Yang, N.: Ontology-based automatic classification and ranking for web documents. In: FSKD 2007: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 627–631. IEEE Computer Society, Washington, DC, USA (2007)
6. Gu, H.Z., Zhou, K.J.: Text classification based on domain ontology. Journal of Communication and Computer 3(5), 29–32 (2006)
7. Janik, M., Kochut, K.: Training-less ontology-based text categorization. In: Workshop on Exploiting Semantic Annotations in Information Retrieval at the 30th European Conference on Information Retrieval (ECIR 2008) (March 2008)
8. Maedche, A., Staab, S.: Ontology learning for the Semantic Web. IEEE Intelligent Systems 16(2), 72–79 (2001)
9. Oh, H.S., Choi, Y., Myaeng, S.H.: Combining global and local information for enhanced deep classification. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC 2010, pp. 1760–1767. ACM, New York (2010)

10. Qi, X., Davison, B.D.: Web page classification: Features and algorithms. ACM Comput. Surv. 41(2), 1–31 (2009)
11. Ren, A., Du, X., Wang, P.: Ontology-based categorization of web search results using YAGO. In: Int. Joint Conference on Computational Sciences and Optimization, pp. 800–804. IEEE, Los Alamitos (2009)
12. Rudy, P., Mike, J., Peter, B., Heinz-Dieter, K.: Ontology-based automatic classification for web pages: design, implementation and evaluation. In: Proceedings of the Third International Conference on Web Information Systems Engineering, WISE 2002, pp. 182–191 (2002)
13. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: SIGIR 1999: Proc. of the 22nd Annual International ACM SIGIR Conference, pp. 206–213. ACM Press, New York (1999)
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
15. Singh, A., Nakata, K.: Hierarchical classification of web search results using personalized ontologies. In: Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction, pp. 1–10 (2005)
16. Song, M.H., Lim, S.Y., Kang, D.J., Lee, S.J.: Automatic classification of web pages based on the concept of domain ontology. In: APSEC 2005, pp. 645–651. IEEE Computer Society, Washington, DC, USA (2005)
17. Song, M., Lim, S., Kang, D., Lee, S.: Ontology-based automatic classification of web documents. In: Huang, D.S., Li, K., Irwin, G. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 690–700. Springer, Heidelberg (2006)
18. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a large ontology from wikipedia and wordnet. Journal of Web Semantics pp. 1–21 (2008)
19. Wu, S.H., Tsai, T.H., Hsu, W.L.: Text categorization using automatically acquired domain ontology. In: Proceedings of the 6th Inter. Workshop on IR with Asian Languages, pp. 138–145. Assoc. Comput. Linguistics, Morristown (2003)
20. Xue, G.R., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: Proc. of the 31st Annual International ACM SIGIR Conference, SIGIR 2008, pp. 619–626. ACM, New York (2008)
21. Yang, X.q., Sun, N., Zhang, Y., Kong, D.r.: General framework for text classification based on domain ontology. In: SMAP 2008, pp. 147–152. IEEE Computer Society, Washington, DC, USA (2008)

Towards Semantic Category Verification with Arbitrary Precision

Dmitri Roussinov

Department of Computer and Information Sciences,
University of Strathclyde,
LT1428b LIVINGSTONE TOWER,
26 Richmond Street, Glasgow, UK G1 1XQ
dmitri.roussinov@cis.strath.ac.uk

Abstract. Many tasks related to or supporting information retrieval, such as query expansion, automated question answering, reasoning, or heterogeneous database integration, involve verification of a semantic category (e.g. "coffee" is a drink, "red" is a color, while "steak" is not a drink and "big" is not a color). We present a novel framework to automatically validate a membership in an arbitrary, not a trained a priori semantic category up to a desired level of accuracy. Our approach does not rely on any manually codified knowledge but instead capitalizes on the diversity of topics and word usage in a large corpus (e.g. World Wide Web). Using TREC factoid questions that expect the answer to belong to a specific semantic category, we show that a very high level of accuracy can be reached by automatically identifying more training seeds and more training patterns when needed. We develop a specific quantitative validation model that takes uncertainty and redundancy in the training data into consideration. We empirically confirm the important aspects of our model through ablation studies.

Keywords: information extraction, question answering, ontologies, natural language processing.

1 Introduction

Semantic verification is the task of automated validation of the membership in a given category, e.g. *red* is a *color*, *coffee* is a *drink*, but *red* is not a *drink*. The problems arises in many tasks related to or supporting information retrieval including 1) **Automated question answering**. For example, the correct answer to the question *What soft drink has most caffeine?* should belong to the category "soft drink." 2) **Query expansion** by adding, for example, specific instances of a category like *international* → *france, UK, germany* 3) **Database federation**, where the automated integration of several heterogeneous databases requires matching an attribute in one database (e.g. having such values as *red, green,* and *purple*) to an attribute (e.g. *colour*) in another database. 4) **Automated reasoning**, where the rules are propagated to all the subclasses of the superclass. 5) **Spellchecking** or **oddity detection**

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 274–284, 2011.

(Fong et al., 2008), where the substitution of a word with its hypernym (superclass) or hyponym (subclass) is considered legitimate while many other types of substitutions are not.

The most precise approaches are through manual or semi-automated development of an extensive taxonomy of possible semantic categories (Harabagiu et al., 2001). However, this requires the anticipation of all the possible categories of interest and substantial "knowledge-engineering." Moreover, those approaches pose significant limitations and do not work well for several types of categories, for example for 1) **relatively rare categories,** e.g. "*American revolutionary general"* 2) **logically complex categories,** e.g. "*a city in Eastern Germany"* 3) **vaguely defined categories,** e.g. "*tourist attraction."*

While fully automated training approaches also exist (e.g., Igo & Riloff, 2009; Huang & Riloff, 2010; Wang & Cohen, 2009; Schlobach et al., 2004) they still rely on manually identified seeds for bootstrapping, and thus, again limit the applications to pre-anticipated categories. The work by Roussinov and Turetken (2009) explored how an *arbitrary* (not pre-anticipated) category can be validated, but its reported performance was below the "knowledge-engineering" approaches or those based on manually specified seeds.

This work presents a theoretical framework aiming to close the gap in performance between those two types of approaches by suggesting a novel model that can identify the training seeds automatically. We empirically validate that the automatically discovered seeds improve the overall verification accuracy to the levels well above those in the prior research.

The next section overviews the prior related work. It is followed by the description of our novel framework suggested here, followed, in turn, by empirical results. The "Conclusions" section summarizes our findings.

2 Prior Work

To avoid laborious creation of large ontologies, researchers have been actively trying automated or semi-automated data driven semantic verification techniques. The idea to count the number of matches to certain simple patterns (e.g. *colors such as red, blue or green; soft-drinks including Pepsi and Sprite* etc.) in order to automatically discover hyponym relationships is typically attributed to Hearst (1992) and was tested on Grolier's American Academic Encyclopedia using WordNet as gold standard. Variations of the idea of Hearst's patterns have been adopted by other researchers: in specific domains (Ahmad, 2003), for anaphora resolution (Poesio, 2002), for discovery of part-of (Charniak, 1999) and causation relations (Girju, 2002).

These approaches are known for a relatively high (50%+) precision, but a very low recall due to the fact that the occurrence of patterns in a closed corpora are typically rare. To overcome this data sparseness problem, researchers resorted to the World Wide Web: Hearst patterns were matched using Google API for the purpose of anaphoric resolution in Markert (2003) or enriching a given ontology in Agirre (2000). It was also the general idea underlying the Armadillo system (Ciravegna, 2003). Earlier work by Brin (one of the founders of Google search portal) (Brin, 1998) presented a bootstrapping approach in

which the system starts with a few patterns, and then tries to induce new patterns using the results of the application of the seed patterns as training dataset.

Cimiano et al. (2005) used Google API to match Hearst-like patterns on the Web in order to find the best concept for an unknown instance, and for finding the appropriate superconcept for a certain concept in a given ontology. SemTag system (Dill, 2003) automatically annotated web pages by disambiguating appearing entities while relying on the TAP lexicon to provide all the possible concepts, senses or meanings of a given entity (Dill, 2003). The systems participating in the Message Understanding Conferences (MUC) achieved accuracies of well above 90% in the task of tagging named entities with respect to their class labels, but the latter were limited to three classes: *organization*, *person* and *location*. The works by Alfonseca and Manandhar (2002) also addressed the problem of assigning the correct ontological class to unknown words by building on the *distributional hypothesis*, i.e. that words are similar to the extent to which they share linguistic contexts. They adopted a vector-space model and exploited verb/object dependencies as features .

Somewhat similar to the methods and the purpose were the approaches within the KnowItAll project (Downey & Etzioni, 2005), which automatically collected thousands of relationships, including the hyponymic ("is-a") ones, from the Web. Within KnowItAll, Etzioni et al. developed a probabilistic model by building on the classic balls-and-urns problem from combinatorics. They established that the urns model outperforms those based on pointwise mutual information metrics used earlier within prior KnowItAll studies (Downey & Etzioni, 2005) or by other researchers, which captures non-randomness of the occurrence of a certain pattern (word sequence). However, the model provides the estimate of the probability of the categorical membership only in the case of supervised learning (anticipated category and manually labeled training examples). It provides *only rank-ordering in the unsupervised case*, and was evaluated by recall and precision on four relations only: `Corporations`, `Countries`, `CEO of a company`, and `Capital Of a Country`.

Schlobach et al. (2004) studied semantic verification for a larger number of categories, but their categories were limited to the geography domain. They also used knowledge intensive methods in addition to pattern matching statistics. Igo & Riloff (2009) combined corpus-based semantic lexicon induction with statistics acquired from the Web to improve the accuracy of automatically acquired domain-specific dictionaries. They used a weakly supervised bootstrapping algorithm to induce a semantic lexicon from a text corpus, and then issued Web queries to generate co-occurrence statistics between each lexicon entry and semantically related terms. The Web statistics captured by pointwise mutual information were used to confirm, or disconfirm, that a word belongs to the intended semantic category. The approach was evaluated on *7* semantic categories representing two domains, and still required "a small" set of seed words for each semantic category.

Huang and Riloff (2010) went further by using positive instances for one class as negative training instances for the others. They noted that "Despite widespread interest in semantic tagging, nearly all semantic taggers ... still rely on supervised learning, which requires annotated data for training," and that iterative self-training often has difficulty "sustaining momentum or it succumbs to *semantic drift*." In that particular work, they evaluated their approach by creating six semantic taggers using a collection of message board posts in the domain of veterinary medicine.

Wang and Cohen (2009) presented a system named ASIA (Automatic Set Instance Acquirer), which takes in the name of a semantic class as input (e.g., "car makers") and automatically outputs its instances (e.g., "ford", "nissan", "toyota"), which is a related but different task. Their approach and those mentioned in their review of the prior work also relied on manually identified seeds. They used two benchmark sets of 4 and 12 categories accordingly.

3 Framework

3.1 Seed Identification

This section presents our suggested framework and the specific implementations followed in this study. The general idea beyond our framework, as already presented above, is to automatically and iteratively identify additional seeds and to train additional verification patterns using those seeds. Thus, we need to 1) create a **model**, that can identify seeds across any of the targeted categories, and 2) provide a **mechanism** to ensure each category will get sufficient number of high quality seeds.

In order to know when to terminate this iterative identification-training process, it is necessary to define a *stopping condition*. As a first step in this direction, here we simply terminate the process when *a desired number of seeds has been identified* leaving more fine-grained accuracy estimation models for future. In order to know how many seeds we have at each iteration and to identify the seeds themselves, the model needs the following two crucial components: 1) A **sampling mechanism** that provides more candidates for seeds as needed. 2) A **binary classifier** that can tell if a given candidate can serve as a seed or not.

In order to define what is the range of possible categories to target and to estimate the performance empirically, a specific background task is needed. Here, as a first stop in this direction of research, we considered automated question answering (QA) as it was defined by TREC conferences (Voorhees & Buckland, 2005), while our introduction above listed more applications. For QA, it is generally sufficient for the sampling mechanism to consider all the word *n*-grams (sequences) up to a certain length (e.g. *4* here) from a certain sample of documents with high likelihood of mentioning both the category in question (e.g. *record company*) and its instances (e.g. *EMI, Interscope, Columbia*, etc). We create such a sample from the merger of snippets returned by a search engine portal (Microsoft's Bing in this study) with response to the two queries: 1) a query consisting of the **category** itself (e.g. *"record company"*) and 2) the query consisting of the **entire question** (e.g. *What record company is Fred Durst with?*). At each iteration step, the sample is extended by the next 100 snippets, starting from the very top of the search results. The maximum depth is *1000* snippets, at which the algorithm terminates even if no desired number of seeds is found.

As the prior work (e.g. Roussinov & Turetken, 2009), we validate membership in an arbitrary category through a *quantitative model* that combines various metrics derived from the number of matches to certain patterns in a large corpus (Entire WWW indexed by Bing). Guided by the prior experience and our preliminary empirical investigations, we decided to primarily use the metrics defined by pointwise mutual

information (Downey & Etzioni, 2005), which captures non-randomness of the occurrence of a certain pattern. Specifically, for each validation pattern *a+b*, we define

$$PMI(a+b) = \frac{\#(a+b)}{\#(a)\cdot\#(b)}\ , \tag{1}$$

where *a* and *b* are the constituent parts of the pattern. E.g., to validate that *Microsoft* is a *company,* a pattern *company such as Microsoft* can be segmented as *company* + *such as Microsoft. DF(p)* is the number of matches to the pattern *p* in the corpus, e.g., here *DF(a)* is the number of times the word *company* occurs in the corpus. When there are several possible ways to segment a pattern into its constituents (e.g. *company such as Microsoft* = *company such as* + *Microsoft*) the algorithm takes the *max* PMI out of all of the possible segmentations. We only use segmentations into two constituents and limit the total number of words in a pattern to 3, not counting the candidate and the category, in order to decrease computational burden.

While several prior works involved more powerful pattern languages, for the sake of generality, in this study, we desired to use the simplest language possible, e.g. as it was in Roussinov & Turetken (2009): our and their pattern language does not involve any information on the part of speech, dependency, grammatical or semantic parsing, nor any other linguistic resources. There are no wildcards in the pattern language, thus, each pattern match is simply an exact string match.

When matching in a limited size corpus, even such a large one as the entire indexed part of the World Wide Web, many patterns do not produce any matches. This results in some undefined PMI metrics. In order to deal with this type of undefined data, *our model operates with the estimated upper and lower bounds of PMI metrics rather than with the metrics themselves* as defined in the following.

First, if we assume that *DF(p)* approximately follows Poisson distribution, we can estimate its standard deviation as its square root:

$$sttdv(\#(p)) = \sqrt{\#(p)}\ . \tag{2}$$

Next, we define the *upper bound estimate* as following:

$$\overline{\#(p)} = \begin{cases} \#(p) + \sqrt{\#(p)}, if\ \ p > 0 \\ 1, if\ \ p = 0 \end{cases} \tag{3}$$

"Flooring" at *1* is desirable in order to define the upper bound for the patterns with no matches to be *1*. The *lower bound estimate* for the *DF* is defined similarly, while the correction is made in the opposite direction:

$$\underline{\#(p)} = \#(p) - \sqrt{\#(p)}\ . \tag{4}$$

Now, we define the upper and low bound estimates for the PMI metric as following:

$$\overline{PMI(a+b)} = \frac{\overline{\#(a)}}{\underline{\#(a)}\cdot\underline{\#(b)}}, \underline{PMI(a+b)} = \frac{\#(a+b)}{\overline{\#(a)}\cdot\overline{\#(b)}}\ . \tag{5}$$

This allows $\overline{PMI(a+b)}$ to be infinity. Again, when more than one segmentation of a pattern into *a* and *b* exist, we take the *max* of their PMIs for the upper and *min* for the lower estimate accordingly. A low value for the estimate of the upper bound $\underline{PMI(p)}$ serves as a signal that a certain pattern likely occurs only due to a random chance and, thus, the category membership is unlikely. Conversely, a high value estimate of the lower bound $\underline{PMI(p)}$ signals that the non randomness of occurrence is strong and the membership is very likely. We believe *modeling the uncertainty and limited corpus size in this way is crucial to achieve empirical advantage over the models reported in the prior works*, along with the other important modeling step: we convert the numerical values of the estimated low and upper bounds of *PMI-s* into boolean features the following way. A boolean feature is defined for each *PMI* metric that shows if it is below or above certain threshold *T*. The threshold *T*, in turn, is determined by the background value of the *PMI* metric for the same pattern, averaged across randomly selected candidates plus its standard deviation:

$$T(p) = <PMI(p)> + stdev(PMI(p)) . \quad (6)$$

Thus, with each pattern *p* we define and only use the following two boolean features:

$$\overline{PMI(p)} < T(p) , \ \underline{PMI(p)} > T(p) . \quad (7)$$

To summarize, our model operates with a large number of boolean features, each evaluating a certain occurrence pattern to check if that occurrence is *very likely* to be more than the random chance occurrence ($\underline{PMI(p)} > T$, as positive evidence) or, conversely, the occurrence is *not likely* to be above the random occurrence ($\overline{PMI(p)} < T$, as negative evidence).

Once the features are obtained, the classification model can be trained in a supervised way or fitted manually on a certain set of categories. The important property of the model is that it needs to generalize well to the categories previously unseen by the algorithm, which is tested here by 10 fold cross-validation. In our empirical tests, we used a logistic regression to train our classifier. Thus, a final model essentially assigns different weights to different patterns.

3.2 Training Using the Seeds

The classifier-based model described in the preceding section can, thus, identify new seed instances for any previously unseen category after evaluating all the candidates in the automatically created sample. The seeds are, in turn, used to automatically identify and train category specific validation patterns as described in this section. First, for each seed, a sample corpus is obtained consisting of the top *1000* snippets returned by the search portal with response to the query consisting of the union of the category and the seed, e.g. *record company EMI*. To identify category-specific patterns, all the occurrences of the seed in the sample are replaced with the special marker (\A), and all the occurrences of the category are replaced with another special marker (\C). All the *n*-grams containing both \A and \C that occur more than twice in

the sample, form the *category specific set of validation patterns* along with the (category-neutral) patterns that are used to identify the seeds. We limited *n* to *4* + the number of words in the category + the number of words in the seed. Once the set of validation patterns has been identified for each category, the logistic regression classifier is trained on the same boolean features as described in the preceding section, thus the weights are specific to each category.

3.3 Answering the Question

Since our primary focus was on seed identification, for the sake of generality, we sought to test its application within as simple framework as possible. While we applied our semantic verification to factoid question answering (QA), for the sake of generality, we decided not to involve any elaborate QA algorithms from prior research (e.g. Lin, 2005; Brill et al., 2002; Roussinov & Turetken 2009) since many of them were optimized for a specific purpose, e.g. TREC competitions, and are hard to replicate without involving a large number of heuristics, which would confound the empirical results.

Fortunately, we were able to design and apply here, a very simple formula to score the candidate answers, which provided the baseline performance comparable (55% correctness and above) with that reported in the prior works involving redundancy-based approaches (Brill et al., 2002; Ravichandran & Hovy, 2002; Roussinov & Turetken, 2009). Each question in our data set has a so-called "target" defined, e.g. *Fred Durst* in the question *What record company is Fred Durst with?* We choose the answer to the question as the candidate with the largest PMI score with the target, while still validating positively into the expected semantic category (*record company* in this example). The pool of candidates was determined by the first 1000 snippets returned by the search portal as a response to the query consisting of the question. In a more general case when the target is not explicitly defined, the PMI scores with each word in the questions can be combined instead or a special target identification mechanism involved.

4 Empirical Evaluation

4.1 Data Sets

Since there are no standard benchmarks with respect to semantic verification, especially for a large number of arbitrary, not anticipated a priori categories, we followed the route similar to Roussinov & Turetken (2009). Thus, we have evaluated semantic verification having a specific application in mind, which in both cases, is automated question answering. Our tested categories were taken from the TREC 2000 and 2003 competition-format conference, the track on Question Answering. The systems participating in the track had to identify exact answers to factual questions, some of them were expected to belong to a certain semantic category. As in Roussinov & Turetken (2009), we considered all the questions that *explicitly* stated a semantic category of the expected answer. For the sake of generality and diversity of categories, we did not use the questions that only implied the expected category. For example, there are many questions that start with "who is" and typically imply *person* as the answer,

questions expecting locations (*where*) or dates (*when*). The target semantic category was identified by applying simple regular expressions to the questions. For example, the regular expression "(What|Which) (.+) (do|does|did|is|was|are|were)" would match a question "What tourist attractions are there in Reims?" and identify "tourist attraction" as a semantic category. This resulted in 109 test questions with 51 unique categories, a much larger sample than used in any prior research.

Comprehensive manual labeling of all the possible candidate instances for the membership in the expected category (e.g. a *city*) would be a daunting task. However, since our focus here was on identifying seeds, it was sufficient to label only the top (according to the algorithm) candidates and to compare the relative performance of various models studied here based on those top candidates -- the idea similar to "pooling" widely accepted for information retrieval evaluation. Here, we performed labeling in an iterative way: only the 10 top candidates within each category were labeled (unless they were labeled previously) then the model was re-trained and the new top candidates obtained. We stopped the iterations when no unlabeled top candidates were produced.

The labeling took approximately 40 man-hours, however was unavoidable considering that no clean benchmarking sets exist. Apparently, to evaluate the accuracy of seed selection, highly comprehensive lists of items are needed to avoid the impact of false negatives in the evaluation. While approximately half of our categories exist in WordNet, the recall of instances of each category is only around 20-30%. While we have found Wikipedia lists to generally provide a higher (30-40%) recall, it still only covers 20% of the categories under consideration. Besides, a special mechanism to resolve naming variations would be needed (e.g. *confederate general* vs. *confederate states general*) if using Wikipedia as a gold standard. This is not surprising since Wikipedia is indented for reading by humans rather than for running automated benchmark tests.

4.2 Results and Discussion

Table 1 presents the results of the comparison of various configurations of our seed identification algorithm using 10-fold cross validation. The "Average Precision" column shows the average across all the categories precision of the top 10 seeds. When the algorithm identified fewer than 10 seeds, the remaining 10 were assumed to be incorrect in the precision calculations.

Table 1. Identifying seeds using various configurations

Configuration	*Average Precision*
All patterns	94%
Logit models and patterns as in Roussinov & Turetken (2009)	65%
Logit models from Roussinov & Turetken (2009), our patterns	70%
4 Best patterns	82%
10 Best patterns	85%
All left patterns only	83%
All right patterns only	79%
Using only PMI high	74%
Using only PMI low	79%

The complete configuration used all the 52 possible patterns. *91%* of categories received at least one correct seed. *65%* of categories had all their seeds correct. The top average precision is remarkably high and well above the typical accuracy of verification (not specifically targeting seed identification), e.g. *50-70%* reported in Roussinov & Turetken (2009), which used smaller number of patterns and logistic regression models combining normalized numbers of matches, rather than the boolean PMI-based features suggested here. The second (from top to bottom) row shows the results using their model and the set of patterns, which is much less precise in identifying seeds. While increasing number of patterns, as shown by the following row, improves the accuracy, the latter still remains well below the models studied here, and is still unlikely to be high enough to prevent "semantic drift." The results in the next two rows ("4 Best patterns" and "10 Best patterns") illustrate that a large number of patterns is needed for good coverage of categories, which is not surprising considering that different patterns are effective for different categories.

The next few rows illustrate the importance of using both the "left patterns" (where *\C* is to the left of \A, e.g. *record companies such as \A*) and the "right patterns" (where *\C* is to the right of *\A*, e.g. *\A as his record company*). Using both left and right patterns in the model suggested here prevents the substrings of the correct instances to become false positives (e.g. *New* from *New York City* as an instance of *city*). Finally, the bottom rows indicate the importance of using the upper and low estimates of the PMI metrics rather than the PMI-s directly to adjust for undefined values and the limited corpus size.

Table 2 presents the results of the comparison of various configurations of the impact of training using the automatically identified seeds on the question answering accuracy. It can be seen that the overall accuracy improves from the ranges typical for the "knowledge-light" redundancy based approaches (55%) to the ranges approaching those of the best "knowledge-engineering" systems (75%, from Voorhees & Buckland, 2005), thus the application of automated seed selection to train semantic validation seems to be extremely promising. It can be also seen that the relative impacts under various configurations are consistent with those presented in Table 1. The result in the row "Using only category neutral patterns" shows that simply using the seeds to train the model using the same, identical for all categories patterns improves the answer accuracy, but that category-specific patterns are still needed to get the best performance.

Table 2. The impact of identifying seeds and training patterns using the seeds on the accuracy of question answering

Configuration	*Answer Accuracy*
Trained on the automatically identified seeds	74%
Not using any automatically identified seeds (baseline)	55%
Using only category neutral patterns	59%
Using only PMI high	61%
Using only PMI low	63%
Using only left patterns	66%
Using only right patterns	63%

5 Conclusions

Many artificial intelligence tasks, such as automated question answering, reasoning, or heterogeneous database integration, involve verification of a semantic category (e.g. "coffee" is a drink, "red" is a color, while "steak" is not a drink and "big" is not a color). We have presented a novel framework to automatically validate a membership in an arbitrary, not a trained a priori semantic category, which may iteratively achieve higher levels of accuracy. Our approach does not rely on any manually codified knowledge but instead capitalizes on the diversity of topics and word usage in a large corpus (e.g. World Wide Web). Using TREC factoid questions that expect the answer to belong to a specific semantic category, we have shown that a very high level of accuracy can be reached by automatically identifying more training seeds and more training patterns when needed. We have developed a specific quantitative validation model that takes uncertainty and redundancy in the training data into consideration. We have also empirically confirmed the important aspects of our model through ablation studies.

Future studies may proceed along more accurate prediction of the accuracy achieved, designing better models or trying other applications.

References

1. Agirre, E., Ansa, O., Hovy, E., Martinez, D.: Enriching very large ontologies using the WWW. In: Proceedings of the ECAI Ontology Learning Workshop (2000)
2. Ahmad, K., Tariq, M., Vrusias, B., Handy, C.: Corpus-based thesaurus construction for image retrieval in specialist domains. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 502–510. Springer, Heidelberg (2003)
3. Alfonseca, E., Manandhar, S.: Extending a lexical ontology by a combination of distributional semantics signatures. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 1–7. Springer, Heidelberg (2002)
4. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics 21(4), 543–566 (1995)
5. Brill, E., Dumais, S., Banko, M.: An Analysis of the AskMSR Question-Answering System. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA (July 6-7, 2002)
6. Brin, S.: Extracting patterns and relations from the World Wide Web. In: Proceedings of the WebDB Workshop at EDBT 1998 (1998)
7. Charniak, E., Berland, M.: Finding parts in very large corpora. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 57–64 (1999)
8. Cimiano, P., Ladwig, G., Staab, S.: Gimme' the context: Context-driven automatic semantic annotation with C-PANKOW. In: Proceedings of the 14th World Wide Web Conference (2005)
9. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: Proceedings of the 13th World Wide Web Conference, pp. 462–471 (2004)
10. Ciravegna, F., Dingli, A., Guthrie, D., Wilks, Y.: Integrating Information to Bootstrap Information Extraction from Web Sites. In: Proceedings of the IJCAI Workshop on Information Integration on the Web, pp. 9–14 (2003)

11. Wang, R.C., Cohen, W.W.: Automatic Set Instance Extraction using the Web ACL 2009 (2009)
12. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the 12th International World Wide Web Conference, pp. 178–186. ACM Press, New York (2003)
13. Downey, D., Etzioni, O., Soderland, S.: A Probabilistic Model of Redundancy in Information Extraction. In: IJCAI 2005 (2005)
14. Dumais, S., Banko, M., Brill, E., Lin, J., Ng, A.: Web Question Answering: Is More Always Better? In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland (August 11-15, 2002)
15. Fong, S.W., Roussinov, D., Skillicorn, D.B.: Detecting Word Substitutions in Text. IEEE Transactions on Knowledge and Data Engineering 20(8), 1067–1076 (2008)
16. Girju, R., Moldovan, M.: Text mining for causal relations. In: Proceedings of the FLAIRS Conference, pp. 360–364 (2002)
17. Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., Morarescu, P.: The Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering. In: Proceedings of the Association for Computational Linguistics, pp. 274–281 (July 2001)
18. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics (1992)
19. Lin, J.: Evaluation of Resources for Question Answering Evaluation. In: Proceedings of ACM Conference on Research and Development in Information Retrieval (2005)
20. Markert, K.K., Modjeska, N., Nissim, M.: Using the web for nominal anaphora resolution. In: EACL Workshop on the Computational Treatment of Anaphora (2003)
21. Moldovan, D., Pasca, M., Harabagiu, S., Surdeanu, M.: Performance Issues and Error Analysis in an Open Domain Question Answering System. In: Proceedings of ACL 2002, pp. 33–40 (2002)
22. Poesio, M., Ishikawa, T., Schulte im Walde, S., Viera, R.: Acquiring lexical knowledge for anaphora resolution. In: Proceedings of the 3rd Conference on Language Resources and Evaluation, LREC (2002)
23. Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: Proceedings of ACL 2002 (2002)
24. Roussinov, D., Turetken, O.: Semantic Verification in an Online Fact Seeking Environment. In: ACM Conference on Information and Knowledge Management, Lisbon, Portugal, pp. 71–78 (2007)
25. Igo, S.P., Riloff, E.: Corpus-based Semantic Lexicon Induction with Web-based Corroboration. In: NAACL 2009 Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics (2009)
26. Huang, R., Riloff, E.: Inducing Domain-specific Semantic Class Taggers from (Almost) Nothing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL) (2010)
27. Schlobach, S., Olsthoorn, M., de Rijke, M.: Type Checking in Open-Domain Question Answering (Extended Abstract). In: Verbrugge, R., Taatgen, N., Schomaker, L. (eds.) Proceedings BNAIC 2004, pp. 367–368 (2004)
28. Voorhees, E., Buckland, L.P. (eds.): Proceedings of the Eleventh Text Retrieval Conference TREC 2004, Gaithersburg, Maryland (November 2004)
29. Voorhees, E., Buckland, L.P. (eds.): Proceedings of the Eleventh Text Retrieval Conference TREC 2005, Gaithersburg, Maryland (November 2005)

Negation for Document Re-ranking in Ad-hoc Retrieval

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro

Dept. of Computer Science - University of Bari "Aldo Moro"
Via Orabona, 4 - I-70125, Bari, Italy
{basilepp,acaputo,semeraro}@di.uniba.it

Abstract. Information about top-ranked documents plays a key role to improve retrieval performance. One of the most common strategies which exploits this kind of information is relevance feedback. Few works have investigated the role of *negative* feedback on retrieval performance. This is probably due to the difficulty of dealing with the concept of non-relevant document. This paper proposes a novel approach to document re-ranking, which relies on the concept of negative feedback represented by non-relevant documents. In our model the concept of non-relevance is defined as a quantum operator in both the classical *Vector Space Model* and a *Semantic Document Space*. The latter is induced from the original document space using a distributional approach based on Random Indexing. The evaluation carried out on a standard document collection shows the effectiveness of the proposed approach and opens new perspectives to address the problem of quantifying the concept of non-relevance.

1 Introduction

Following the cluster-based re-ranking paradigm, several re-ranking techniques aim to improve the performance of the initial search by exploiting top-ranked results. This paradigm is founded on the cluster hypothesis [14] according to which similar documents tend to be relevant to the same request. Moreover, top-ranked documents are also supposed to be the most relevant ones. Hence, it is reasonable to think that also non-relevant documents could improve performance in document re-ranking. Whilst relevant documents have been successfully employed in several approaches to improve Information Retrieval (IR) performance, non-relevant ones seem not to arouse researchers' interest. Singhal et al. [19] achieved an interesting result for the learning routing query problem: they showed that using non-relevant documents close to the query, in place of those in the whole collection, is more effective. An early attempt to model terms negation in pseudo-relevance feedback by quantum logic operators is due to Widdows [23]. In his work, Widdows has shown that negation in quantum logic is able to remove, from the result set, not only unwanted terms but also their related meaning. The concept of vectors orthogonality is exploited to express queries like "Retrieve documents that contain term A NOT term B". Widdows suggests that vectors which represent unrelated concepts should be orthogonal

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 285–296, 2011.

to each other. Indeed, orthogonality prevents vectors from sharing common features. Among more recent works, a successful use of non-relevant documents for *negative* pseudo-relevance feedback has been carried out in [21], where authors point out the effectiveness of their approach with poorly performing queries.

This work investigates the role of non-relevant documents in document re-ranking. In particular, our re-ranking strategy is based on a pseudo-relevance feedback approach which takes into account both relevant and non-relevant documents in the initial document ranking. The key idea behind our approach is to build an *ideal document* which fits the user's need, and then re-rank documents on the ground of their similarity with respect to the ideal document. Generally, standard relevance feedback methods are able to handle negative feedback by subtracting "information" from the original query. The key issue of this approach is to quantify the side effect caused by information loss. To deal with this effect, we propose a negative feedback based on quantum negation that is able to remove only the unwanted aspects pertaining to non-relevant documents. In our approach, documents are represented as vectors in a geometric space in which similar documents are represented close to each other. This space can be the classical *Vector Space Model* (VSM) or a *Semantic Document Space* (SDS) induced by a distributional approach. Moreover, we compare our strategy with a classical strategy based on "information subtraction".

How to identify non-relevant documents is an open question. We propose two distinct approaches in our work: the former exploits documents at the bottom of the rank, while the latter takes the non-relevant documents directly from relevance judgments. These approaches are thoroughly described in Section 5. We want to underline here that how to identify non-relevant documents is out of the scope of this paper.

The paper is structured as follows. Section 2 describes the proposed strategy for re-ranking, while *Semantic Document Space* is presented in Section 3. Section 4 gives details about quantum negation. Experiments performed for evaluating these methods are presented in Section 5. Related work are briefly analyzed in Section 6, while the last section reports some final observations.

2 A Re-ranking Method Based on Non-relevant Documents

This section describes our re-ranking strategy based on non-relevant documents.

The main idea is to build a document vector which attempts to model the *ideal document* in response to a user query, and then exploits this vector to re-rank the initial set of ranked documents D_{init}. The ideal document vector d^* should fit the *concepts* in the set of relevant documents D^+, while skipping *concepts* in the set D^- of non-relevant ones. Identifying relevant documents is quite straightforward: we assume the top ranked documents in D_{init} as relevant, whereas identifying non-relevant documents is not trivial. To this purpose, we propose two strategies: the former relies on documents at the bottom of D_{init}, while the latter needs relevance judgments. The ideal document vector d^* is

exploited to re-rank documents in D_{init} on the ground of the similarity between d^* and each document in D_{init} in the geometrical space they are defined on.

From now on two geometrical spaces will be investigated: the classical *Vector Space Model* (VSM) and a *Semantic Document Space* (SDS) built using a distributional approach presented in Section 3.

Formally, a new relevance score is computed for each document $d_i \in D_{init}$ according to the following equation:

$$S(d_i) = \alpha * S_{D_{init}}(d_i) + (1 - \alpha) * sim(d_i, d^*) \quad (1)$$

where $S_{D_{init}}(d_i)$ is the score of d_i in the initial rank D_{init}, while $sim(d_i, d^*)$ is the similarity degree between the document vector d_i and the ideal document vector d^* computed by cosine similarity. The outcome of the process is a list of documents ranked according to the new scores computed using Equation 1.

To perform re-ranking using a classical "information subtraction" strategy, we assume that documents are represented by classical bag-of-words. Given the subset of relevant documents D^+ and the subset of non-relevant documents D^-, both of them computed starting from D_{init}, the ideal document d^*_C is defined as follows:

$$d^*_C = \frac{1}{|D^+|} \sum_{i \in D^+} d_i - \frac{1}{|D^-|} \sum_{j \in D^-} d_j \quad (2)$$

This formula is based on Rocchio algorithm for relevance feedback [18], but the proposed approach differs in two points:

1. the goal is document re-ranking and not query expansion. Our idea is not to add or subtract terms to/from the original query, but rather to re-rank documents using inter-document similarities;
2. we add and subtract vector documents without weighing differently relevant and non-relevant documents. This strategy assigns the same significance to both relevant and non-relevant documents.

Finally, re-ranking is performed as previously described in Equation 1.

3 Semantic Document Space

The strategy used to re-rank documents in a semantic space relies on the distributional approach used to build the *Semantic Document Space*. This approach represents documents as vectors in a high dimensional space, such as `WordSpace` [17].

The core idea behind `WordSpace` is that words and concepts are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space (geometric metaphor of meaning). Replacing words with documents results in a high dimensional space where similar documents are represented close. Therefore, semantic similarity between documents can be represented as proximity in that n-dimensional space. The main characteristic of

the geometric metaphor of meaning is not that meanings are represented as locations in a semantic space, but rather that similarity between documents can be expressed in spatial terms, as proximity in a high-dimensional space. One of the great virtues of the distributional approach is that document spaces can be built using entirely unsupervised analysis of free text. According to the *distributional hypothesis* [7], the meaning of a word is determined by the rules of its usage in the context of ordinary and concrete language behavior. This means that words are semantically similar to the extent that they share *contexts* (surrounding words). If "green" and "yellow" frequently occur in the same context, for example near the word "color", the hypothesis states that they are semantically related or similar. Co-occurrence is defined with respect to a context, for example a window of terms of fixed length, or a document. In our strategy the role of contexts is played by words, while the role of words is played by documents. Hence, documents are similar if they have the same contexts, that is to say, they are similar if they share the same words. It is important to underline here that a word is represented by a vector in a high dimensional space. Since these techniques are expected to handle efficiently high dimensional vectors, a common choice is to adopt *dimensionality reduction* algorithms that allow for representing high-dimensional data in a lower-dimensional space without losing information. *Latent Semantic Analysis (LSA)* [12] collects the text data in a co-occurrence matrix, which is then decomposed into smaller matrices with Singular-Value Decomposition (SVD), by capturing latent semantic structures in the text data. The main drawback of SVD is scalability. Differently from LSA, *Random Indexing* (RI) [8] targets the problem of dimensionality reduction by removing the need for matrix decomposition or factorization. RI incrementally accumulates context vectors, which can be later assembled into a new space, thus it offers a novel way of conceptualizing the construction of context vectors. In this space it is possible to define "negation" using the orthogonal complement operator, as proposed in an early work about quantum logic [2].

We exploit both RI and quantum negation to implement our re-ranking method in SDS. In particular, we adopt RI to build the SDS, while quantum negation is useful to build the ideal document d^* as a vector which represents the disjunction of relevant documents (D^+) and the negation of non-relevant ones (D^-) in the SDS. Moreover, we also apply quantum negation in VSM, in order to compare the two spaces.

RI is based on the concept of Random Projection [5]: the idea is that high dimensional vectors chosen randomly are "nearly orthogonal". This yields a result that is comparable to orthogonalization methods, such as SVD, but saving computational resources. Specifically, RI creates the *Semantic Document Space* in two steps:

1. a context vector is assigned to each word. This vector is sparse, high-dimensional and ternary, which means that its elements can take values in {-1, 0, 1}. The context vector contains a small number of randomly distributed non-zero elements, and the structure of this vector follows the hypothesis behind the concept of Random Projection;

2. context vectors are accumulated incrementally by analyzing documents in which terms occur. In particular, the semantic vector assigned to each document is the sum of the context vectors of the terms which occur in the document. It should be pointed out that context vectors are added by multiplying them by the term frequency.

4 Re-ranking Using Quantum Negation

To build the ideal document d^* in the geometrical space (SDS or VSM), we need to compute a vector which is close to relevant documents and it is unrelated to non-relevant ones. In our space the concept of relevance is expressed in terms of similarity, while the concept of irrelevance is defined by orthogonality (similarity equals to zero). Formally, we want to compute the vector which represents the following logical operation:

$$d^* = d_1^+ \vee d_2^+ \vee \ldots \vee d_n^+ \wedge NOT(d_1^-) \wedge NOT(d_2^-) \wedge \ldots \wedge NOT(d_m^-) \quad (3)$$

where $D^+ = \{d_i^+, i = 1 \ldots n\}$ and $D^- = \{d_j^-, j = 1 \ldots m\}$.

As shown in [22], given two vectors a and b in a vector space V endowed with a scalar product, $a \quad NOT \quad b$ corresponds to the projection of a onto the orthogonal space $\langle b \rangle^\perp \equiv \{v \in V : \forall b \in \langle b \rangle, v \cdot b = 0\}$, where $\langle b \rangle$ is the subspace $\{\lambda b : \lambda \in \mathbb{R}\}$. Equation 3 consists in computing a vector which represents the disjunction of the documents in D^+, and then projecting this vector onto all m orthogonal spaces defined by the documents in D^-. This operation is quite complex to compute, but applying De Morgan rules to the conjunction of negations, it can be transformed in a single negation of disjunctions:

$$d^* = d_1^+ \vee d_2^+ \vee \ldots \vee d_n^+ \wedge NOT(d_1^- \vee d_2^- \vee \ldots \vee d_m^-) \quad (4)$$

Thus, it is possible to build the ideal document vector d^* in two steps:

1. compute the disjunction of relevant documents as the vector sum of relevant documents. Indeed, disjunction in set theory is modeled as set union, which corresponds to the vector sum in linear algebra;
2. compute the projection of the vector sum of relevant documents onto the orthogonal space defined by the vector sum of non-relevant documents, for example using the Gram-Schmidt method. This means that the result vector captures those aspects that are common to relevant documents and are distant from non-relevant ones.

Disjunction and negation using quantum logic are thoroughly described in [22]. Finally, the re-ranking algorithm is implemented as described in Section 2.

5 Evaluation

The goal of the evaluation is to prove that our re-ranking strategy based on quantum negation improves retrieval performance and outperforms the classical

"information subtraction" method. Moreover, we want to evaluate the performance in both spaces: *Semantic Document Space* (SDS) and the classical *Vector Space Model* (VSM).

We set up a baseline system based on the BM25 multi-fields model [15]. The evaluation has been designed using the CLEF 2009 Ad-Hoc WSD Robust Task collection [1]. The Robust task allows us to evaluate IR system performance even when difficult queries are involved. The CLEF 2009 collection consists of 166,717 documents which have two fields: HEADLINE and TEXT. Table 1 shows the BM25 parameters, where b is a constant related to the field length, k_1 is a free parameter, and *boost* is the boosting factor applied to that field.

Table 1. BM25 parameters used in the experiments

Field	k_1	b	*boost*
HEADLINE	3.25	0.70	2.00
TEXT			1.00

To evaluate the performance we executed several runs using the topics provided by CLEF organizers. In detail, the CLEF 2009 collection has 150 topics. Topics are structured in three fields: TITLE, DESCRIPTION and NARRATIVE. We used only TITLE and DESCRIPTION, because NARRATIVE field is the topic description used by assessors. Moreover, we used different boosting factors for each topic field (TITLE=4 and DESCRIPTION=1) to highlight terms in the TITLE.

We performed 150 runs by considering all possible combinations of the three parameters involved in our method. In particular, we took into account: n (the cardinality of D^+), m (the cardinality of D^-) and the parameter α used for the linear combination of the scores (see Equation 1). We selected different ranges for each parameter: n ranges in $[1, 5, 10, 20, 40]$, m ranges in $[0, 1, 5, 10, 20, 40]$, while α ranges in $[0.3, 0.4, 0.5, 0.6, 0.7]$. The dimension of context vectors in the SDS has been set to 1,000. In addition we set the cardinality of D_{init} to 1,000. All the metrics have been computed on the first 1,000 returned documents, as prescribed by the CLEF evaluation campaign.

We proposed two strategies to select the set (D^-) of non-relevant documents:

1. *BOTTOM*, which selects the non-relevant documents from the bottom of the rank;
2. *RELJUD*, which relies on relevance judgments provided by CLEF organizers. This technique selects the top m ranked documents which are non-relevant exploiting the relevance judgments. We use this strategy to "simulate" the user's explicit feedback; in other words we assume that the user selects the first m non-relevant documents.

Both strategies are not grounded on a theory, but rather they are based on plausible heuristics. Our hypothesis is that, in order to develop a theoretically sound framework, an analysis of scores distribution in the rank could help. With

that goal in mind, we plan to perform that analysis in a future work. In this paper, our final goal is to exploit non-relevant documents in re-ranking.

We evaluate each run in terms of Mean Average Precision (MAP) and Geometric Mean Average Precision (GMAP) over all the queries.

Table 2 reports the results for the "information subtraction" strategy, while Tables 3 and 4 show the results for the quantum negation re-ranking in the VSM and SDS spaces, respectively. Each table reports the *baseline* and, under the baseline, the best performance obtained when only relevant documents are involved. Moreover, each table shows the best five runs for $BOTTOM$ and $RELJUD$ strategies with respect to MAP values. Improvements in percentage ($\Delta\%$) with respect to the baseline are reported for MAP and GMAP values.

Table 2. Results using "information subtraction" strategy

Method	*Run*	n	m	α	MAP	$\Delta\%$	$GMAP$	$\Delta\%$
-	baseline	-	-	-	0.4139	-	0.1846	-
-	1.no.neg.	1	0	0.6	0.4208	+1.67	0.1754	-4.98
BOTTOM	1.B1	1	1	0.6	0.4175	0.87	0.1750	-5.20
	1.B2	1	10	0.5	0.4174	0.85	0.1762	-4.55
	1.B3	1	20	0.5	0.4172	0.80	0.1762	-4.55
	1.B4	1	5	0.6	0.4171	0.77	0.1757	-4.82
	1.B5	1	10	0.6	0.4166	0.65	0.1749	-5.25
RELJUD	1.R1	40	40	0.7	0.5932	+43.32	0.2948	+59.70
	1.R2	40	40	0.6	0.5778	+39.60	0.2849	+54.33
	1.R3	40	40	0.5	0.5517	+33.29	0.2677	+45.02
	1.R4	20	20	0.7	0.5512	+33.17	0.2535	+37.32
	1.R5	20	20	0.6	0.5426	+31.09	0.2500	+35.43

Table 3. Results using *Vector Space Model*

Method	*Run*	n	m	α	MAP	$\Delta\%$	$GMAP$	$\Delta\%$
-	baseline	-	-	-	0.4139	-	0.1846	-
-	2.no.neg.	1	0	0.5	0.4372	+5.63	0.1923	+4.17
BOTTOM	2.B1	1	5	0.6	0.4384	+5.92	0.1923	+4.17
	2.B2	1	10	0.6	0.4379	+5.80	0.1921	+4.06
	2.B3	1	1	0.5	0.4377	+5.75	0.1928	+4.44
	2.B4	1	5	0.5	0.4376	+5.73	0.1926	+4.33
	2.B5	1	20	0.6	0.4372	+5.73	0.1917	+3.85
RELJUD	2.R1	40	40	0.7	0.6649	+60.64	0.3240	+75.51
	2.R2	40	40	0.6	0.6470	+56.32	0.3156	+70.96
	2.R3	40	40	0.5	0.6223	+50.35	0.3124	+69.23
	2.R4	20	40	0.7	0.6176	+49.21	0.2859	+54.88
	2.R5	20	20	0.7	0.6107	+47.55	0.2836	+53.63

Table 4. Results using *Semantic Document Space*

Method	*Run*	n	m	α	MAP	$\Delta\%$	$GMAP$	$\Delta\%$
-	baseline	-	-	-	0.4139	-	0.1846	-
-	3.no.neg.	1	0	0.5	0.4362	+5.39	0.1931	+4.60
BOTTOM	3.B1	1	5	0.6	0.4367	+5.51	0.1928	+4.44
	3.B2	1	5	0.5	0.4365	+5.46	0.1934	+4.77
	3.B3	1	1	0.5	0.4363	+5.41	0.1931	+4.60
	3.B4	1	10	0.5	0.4358	+5.29	0.1934	+4.77
	3.B5	1	20	0.5	0.4352	+5.15	0.1926	+4.33
RELJUD	3.R1	40	40	0.7	0.6646	+60.57	0.3415	+84.99
	3.R2	40	40	0.6	0.6508	+57.24	0.3314	+79.52
	3.R3	40	40	0.5	0.6260	+51.24	0.3162	+71.29
	3.R4	20	40	0.7	0.6157	+48.76	0.3014	+63.27
	3.R5	20	20	0.7	0.6077	+46.82	0.2947	+59.64

The experimental results are very encouraging. Both methods ($BOTTOM$ and $RELJUD$) show improvements with respect to the baseline in all the approaches.

The main outcome is that quantum negation outperforms "information subtraction" strategy. Generally, $BOTTOM$ strategy results in not significant improvements. Moreover, in the case of "information subtraction", the introduction of non-relevant documents results in lower performance. This suggests that the $BOTTOM$ strategy is not able to identify non-relevant documents. The blind selection of non-relevant documents produces a side effect in "information subtraction" strategy due to the information loss, while the quantum negation has the effect of removing from relevant documents only those "negative" aspects that belong to the non-relevant ones. Considering the document spaces for the quantum negation strategy, SDS behaves better than classical VSM in terms of GMAP, while in terms of MAP the differences are not significant.

As expected, the method $RELJUD$ obtains very high results. In this case quantum negation obtains very high improvements with respect to the "information subtraction" strategy. This proves that quantum negation is able to take advantage of information about non-relevant documents. The best results in GMAP are obtained using SDS, while the MAP is similar to the one achieved by VSM. Generally, the results show that the differences between SDS and VSM are not relevant, but computing our re-ranking algorithm in SDS is more efficient.

The best results in $RELJUD$ are obtained when a lot of non-relevant documents are involved, but in a real scenario this is highly improbable. In a more realistic setting, the user selects just one non-relevant document. We performed several runs considering only one non-relevant document and varying the numbers of those relevant. The highest MAP value for SDS is 0.4606 (GMAP=0.2056), while for VSM is 0.4588 (GMAP=0.2028). Both values are obtained with five relevant documents (these results are not reported in Table 4 for the sake of simplicity).

Figure 1 plots the MAP values for each run and method: COS stands for "information subtraction" strategy, while SV and VSM for quantum negation in SDS and VSM spaces, respectively. This graph highlights as the system performance vary according to parameters changes. It is possible to note that SV and VSM tend to have a similar trend underlined by the frequent overlap of their lines in the graph. The method based on "information subtraction" generally achieves lower values of MAP, with an absolute minimum of 0.3418. To a large extent, the methods based on quantum negation have a more stable trend, the lowest MAP value is 0.3748. This value occurs when only one non-relevant document is involved. These values support our thesis: quantum negation works better when several non-relevant documents are considered.

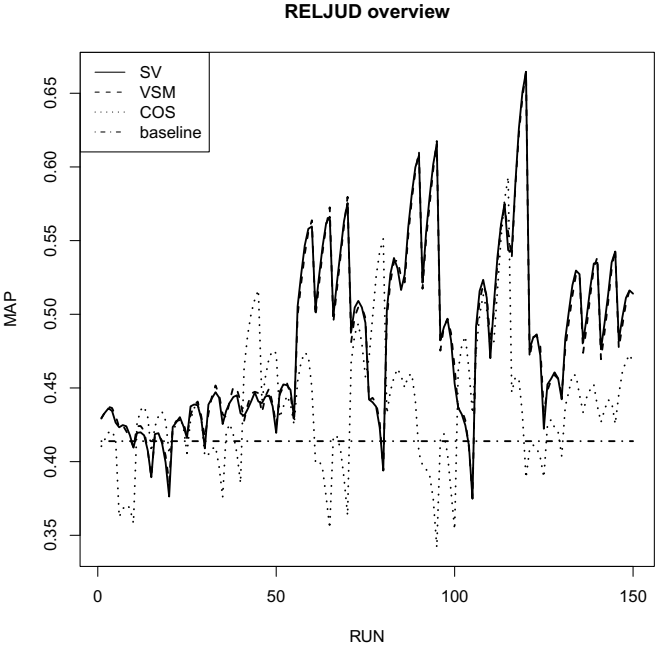


Fig. 1. Plot of MAP values using RELJUD strategy

6 Related Work

Document re-ranking techniques have been thoroughly investigated in Information Retrieval. In the language modeling framework, the usage of this paradigm has been twofold: the traditional cluster-based retrieval has been juxtaposed with document language model smoothing in which document representation incorporates cluster-related information [10,11,13]. These types of re-ranking algorithms in the language modeling framework have shown promising results, especially when cluster information is exploited for document smoothing. Based on the notion that different clusters can cover different query aspects, either

query-independent or query-specific cluster techniques have been exploited to re-rank the result list giving more importance to documents which cover as many aspects as possible [10,11]. The cluster hypothesis has also inspired some re-ranking techniques based on the inter-document similarities [4,9].

The idea to build a document which represents the "ideal" response to the user's information need is of course not new. In [4] documents in the result list are re-weighed according to a relevance function which reflects the distance between documents and the "ideal document". As documents in response to a query are distant from the "ideal document", their weight in the final list should drop down. Authors suppose that similar documents should get a similar weight in the final rank; moreover, the distance between each document and the ideal one represents a degree of dissimilarity to the query. However, authors have merely proposed a theoretical method that was not supported by any empirical evidence. The assumption that documents with related content should obtain similar scores in response to a query has also inspired the work by Diaz [6], where the concepts of inter-document relatedness and score regularization take the place of inter-document similarities and document re-ranking. In a similar vein, other researchers [3,9] use inter-document similarities to combine several retrieved lists. In this case, the idea of "similarity" is used to give support to documents with similar content highly ranked across multiple result lists. Improving retrieval effectiveness by exploiting top-ranked documents has also fed another kind of IR technique: the pseudo-relevance feedback [16]. Pseudo-relevance feedback relies on the assumption that top-ranked documents are also the relevant ones. These documents are exploited to add new terms or to re-weigh the original query. A comparison of classification (label propagation and K-nearest neighbor) versus pseudo-relevance feedback methods was carried out in [20]. All experiments were performed on three Chinese collections and led to the conclusion that pseudo-relevance feedback helps, whereas the effectiveness of the other methods has still to be proven. Conversely, negative feedback has not been deeply explored. A recent work [21] explored the use of non-relevant documents in two IR frameworks: vector space model and language modeling, concluding that negative relevance feedback can increase the system effectiveness, especially with poorly performing queries.

7 Conclusions

This paper proposes a novel approach based on "negative" evidence for document re-ranking by inter-document similarities. The novelty lies on the use of quantum logic to capture the negative aspects of non-relevant documents. This method has shown its effectiveness with respect to a baseline system based on BM25 and a re-ranking method based on the classic "information subtraction" method. Moreover, the evaluation has proved the robustness of the proposed strategy and its capability to remove from relevant documents only those "negative" aspects that belong to the non-relevant ones. The main outcome of this work is that negation expressed by quantum logic operator is able to model effectively the

concept of non-relevance. This opens new perspectives in all those tasks where the idea of non-relevance is a key issue.

Acknowledgements. This research was partially funded by MIUR (Ministero dell'Università e della Ricerca) under the contract Fondo per le Agevolazioni alla Ricerca, DM19410 "Laboratorio" di Bioinformatica per la Biodiversità Molecolare (2007-2011).

References

1. Agirre, E., Di Nunzio, G.M., Mandl, T., Otegi, A.: CLEF 2009 Ad Hoc Track Overview: Robust-WSD Task. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mostefa, D., Penas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 36–49. Springer, Heidelberg (2010)
2. Birkhoff, G., von Neumann, J.: The logic of quantum mechanics. Annals of Mathematics 37(4), 823–843 (1936)
3. Caputo, A., Basile, P., Semeraro, G.: From fusion to re-ranking: a semantic approach. In: Crestani, F., Marchand-Maillet, S., Chen, H.H., Efthimiadis, E.N., Savoy, J. (eds.) SIGIR, pp. 815–816. ACM, New York (2010)
4. Danilowicz, C., Balinski, J.: Document ranking based upon Markov chains. Information Processing & Management 37(4), 623–637 (2001)
5. Dasgupta, S., Gupta, A.: An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures & Algorithms 22(1), 60–65 (2003)
6. Diaz, F.: Regularizing ad hoc retrieval scores. In: CIKM 2005: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 672–679. ACM, New York (2005)
7. Harris, Z.: Mathematical Structures of Language. Interscience, New York (1968)
8. Kanerva, P.: Sparse Distributed Memory. MIT Press, Cambridge (1988)
9. Kozorovitzky, A., Kurland, O.: From "identical" to "similar": Fusing retrieved lists based on inter-document similarities. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 212–223. Springer, Heidelberg (2009)
10. Kurland, O.: Re-ranking search results using language models of query-specific clusters. Information Retrieval 12(4), 437–460 (2009)
11. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information retrieval. In: SIGIR 2004: Proceedings of the 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 194–201. ACM, New York (2004)
12. Landauer, T.K., Dumais, S.T.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. Psychological Review 104(2), 211–240 (1997)
13. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: SIGIR 2004: Proceedings of the 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 186–193. ACM, New York (2004)
14. van Rijsbergen, C.J.: Information Retrieval. Butterworth, London (1979)
15. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In: CIKM 2004: Proceedings of the Thirteenth ACM Int. Conf. on Information and Knowledge Management, pp. 42–49. ACM, New York (2004)

16. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. Knowledge Engineering Review 18(2), 95–145 (2003)
17. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. Ph.D. thesis, Stockholm University, Department of Linguistics (2006)
18. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4), 288–297 (1990)
19. Singhal, A., Mitra, M., Buckley, C.: Learning routing queries in a query zone. In: SIGIR 1997: Proceedings of the 20th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 25–32. ACM, New York (1997)
20. Tseng, Y., Tsai, C., Chuang, Z.: On the robustness of document re-ranking techniques: a comparison of label propagation, knn, and relevance feedback. In: Proceedings of NTCIR-6 Workshop (2007)
21. Wang, X., Fang, H., Zhai, C.: A study of methods for negative relevance feedback. In: SIGIR 2008: Proceedings of the 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 219–226. ACM, New York (2008)
22. Widdows, D., Peters, S.: Word vectors and quantum logic: Experiments with negation and disjunction. Mathematics of language (8), 141–154 (2003)
23. Widdows, D.: Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In: ACL 2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pp. 136–143. Association for Computational Linguistics, Morristown (2003)

A Descriptive Approach to Classification

Miguel Martinez-Alvarez and Thomas Roelleke

Queen Mary, University of London
{miguel,thor}@eecs.qmul.ac.uk

Abstract. Nowadays information systems are required to be more adaptable and flexible than before to deal with the rapidly increasing quantity of available data and changing information needs. Text Classification (TC) is a useful task that can help to solve different problems in different fields. This paper investigates the application of descriptive approaches for modelling classification. The main objectives are increasing abstraction and flexibility so that expert users are able to customise specific strategies for their needs.

The contribution of this paper is two-fold. Firstly, it illustrates that the modelling of classifiers in a descriptive approach is possible and it leads to a close definition w.r.t. mathematical formulations. Moreover, the automatic translation from PDatalog to mathematical formulation is discussed. Secondly, quality and efficiency results prove the approach feasibility for real-scale collections.

1 Introduction and Motivation

Nowadays, information systems have to deal with multiple sources of available data in different formats and rapidly changing requirements reflecting diverse information needs. As a result, the importance of adaptability and productivity in Information Retrieval (IR) systems is increasing. This fact is especially important in business environments when the information required at different moments can be extremely different and its utility may be contingent on timely implementation. How quickly a new problem is solved is often as important as how well you solve it.

Current systems are usually developed for specific cases, implying that too much time is spent having to rewrite and check high portions of the original implementation for other purposes. We believe that if the gap between the conceptual model and the implementation is minimised, expert users would be able to directly define specific strategies for solving their information needs. Therefore, increasing the productivity and adaptability of current approaches. Descriptive approaches are a well suited solution for this objective due to the high-level definition of models and their "Plug & Play" capabilities that allow quick changes with minimum engineering effort. This paper focuses on providing an abstraction for the classification task using a descriptive approach. Classification has been applied in several situations for different purposes and fields, making it a suitable candidate for being part of a flexible framework.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 297–308, 2011.

We aim to show, using different examples, how our approach provides an understandable and elegant modelling of classifiers, close (or even translatable) to its mathematical equation. This abstraction provides the flexibility and adaptability required for dynamic environments, allowing expert users to customise specific strategies. The long-term goal is to achieve a descriptive and composable IR technology that can be customised into a task-specific solution.

The remainder of this paper is structured as follows: Section 2 briefly reviews Naive-Bayes and k-NN classifiers and descriptive approaches, Section 3 presents their modelling using a descriptive approach. Furthermore, the automatic translation of these models to mathematical equations is discussed, Section 4 shows the results achieved using two standard Text Classification collections. Finally, Section 5 concludes the paper and discusses future work.

2 Background and Related work

2.1 Standard Classifiers

Text Classification is a useful task that assigns elements to one or more classes from a preselected set. It has been used in different fields for different purposes such as news categorisation or spam detection [18]. It has been traditionally based on term-based representations, viewing documents as bags of words.

Naive-Bayes Classifiers. Naive-Bayes classifiers use the Bayes Theorem for inferring knowledge assuming the independence between features, given the context of a class [10]. This is a common assumption that makes the computation feasible. As an indirect result, it can be applied to larger collections. Given this assumption, the probability of a document being labelled in a class is defined in equation 1.

$$P(\mathrm{c}_k|d_i) = \frac{P(\mathrm{c}_k) \cdot P(\mathrm{d}_i|\mathrm{c}_k)}{P(\mathrm{d}_i)} = \frac{P(\mathrm{c}_k) \cdot \prod_{\mathrm{t} \in \mathrm{d}_i} P(\mathrm{t}|\mathrm{c}_k)}{P(\mathrm{d}_i)} \tag{1}$$

$$\mathrm{score}_{\mathrm{NB}}(\mathrm{c}_k, \mathrm{d}_i) := \frac{P(\mathrm{c}_k) \cdot \prod_{\mathrm{t} \in \mathrm{d}_i} P(\mathrm{t}|\mathrm{c}_k)}{\sum_k P(\mathrm{c}_k) \cdot \prod_{\mathrm{t} \in \mathrm{d}_i} P(\mathrm{t}|\mathrm{c}_k)} \tag{2}$$

All classifiers that make this assumption are usually referred to as Naive-Bayes, even if there are differences between them in the probabilities computation [10].

k-Nearest Neighbours (k-NN) Classifiers. K-NN is a lazy learning instance-based method that has been used for classification and pattern recognition tasks for the last 40 years [18]. It categorises documents into classes taking into account what train examples are the "nearest" based on a similarity measure. There are several strategies for, given a document, computing the score for each class. One of the most common is presented in Equation 4 where the score of a class is the sum of similarity scores for those documents labelled in the class, observing only the k most similar documents. Although this is the most

common technique, there are others such as counting the number of neighbours from each class, without taken into account the score of their similarity.

$$\text{sim}(d_i, d_j) := \frac{d_i \cdot d_j}{||d_i|| \cdot ||d_j||} \tag{3}$$

$$\text{score}_{\text{k-NN}}(c, d_i) := \sum_{d_j \in c} \text{sim}_k(d_i, d_j) \tag{4}$$

Two parameters are needed, the number of neighbours (k) and the similarity algorithm. The former varies depending on the collection while, for the latter, the cosine similarity is the most common option.

2.2 Descriptive Approaches

Descriptive approaches allow to define high-level functionality making the implementation clearer and the knowledge transfer easier. As a result, productivity will be increased [8]. Models and tasks can be defined as modules and then "concatenated", processing the information as a pipeline where some outputs of one module are the input of the following one. This combination does not involve any coding process due to the paradigm's "Plug & Play" capabilities offered by its functional syntax. This solution provides the flexibility needed for specifying and quickly combining different IR tasks and/or models. Furthermore, it is possible to represent complex objects and structured data.

Research has been done related to abstraction layers using descriptive approaches for different tasks. For instance, a declarative specification language (Dyna) has been used for modelling Natural Language Processing (NLP) algorithms [2], concluding that it is extremely helpful, even if it is slower than "handcrafted" code. Other example is the description of a framework that synthesises and extends deductive and semiring parsing, adapting them for translation [9]. This work shows that logic make an attractive shorthand for description, analysis and construction of decoding algorithms for translation. It also explains that descriptive approaches could be very beneficial when implementing large-scale translation systems which the authors identify as a major engineering challenge, requiring a huge amount of resources. In addition, the logical description has helped them to understand and compare the common elements for different models/algorithms and their differences. Among descriptives approaches, Probabilistic Logics has been applied for modelling and reasoning in different environments several times [7]. The language that has been used in this paper, Probabilistic Datalog(explained in section 2.3), is one of its representatives. Similar languages such Problog [14] and P-Log [6] have also been used for modelling and reasoning. In addition, although there are is research specifically related to the task of modelling classifiers using a descriptive approach [13,1], they are focused in learning rules and/or use domain ontologies.

2.3 Probabilistic Datalog

The specific language that is used for the modelling of classifiers is Probabilistic Datalog (PDatalog). It is based on Probabilistic Logics and it combines Datalog (query language used in deductive databases) and probability theory [5,15]. It was extended to improve its expressiveness and scalability for modelling ranking models [16].

```
1  #P(grade|degree): Learned from knowledge base.
2  p_grade_degree SUM(Grade, Degree) :-
3          grade(Student, Grade, Degree)|(Degree);

5  #P(grade|person): Inferred using P(grade|degree)
6  p_grade_person (Grade, Person) :-
7          p_grade_degree(Grade, Degree)
8          & register(Person, Degree);

10 # Given... grade(John, B, Art);
11 #          grade(Mary, B, Maths); grade(Anna, A, Art);
12 #          register (Matt, Art);  register (Mike, Maths);

14 # Results...  0.5 p_grade_person(A, Matt);
15 #            0.5 p_grade_person(B, Matt);
16 #            1.0 p_grade_degree(B, Mike);
```

Fig. 1. PDatalog example code

It is a flexible platform that has been used as an intermediate processing layer for semantic/terminological logics in different IR tasks such as *ad-hoc* retrieval [11,12], annotated document retrieval [4] and summarisation [3]. Furthermore, Datalog has been applied for Information Extraction [19], highlighting the advantages of its declarative paradigm. Figure 1 shows and example that computes the probability of a student obtaining a specific grade ($P(grade|student)$) based on probabilities of grades given subjects from the previous year.

3 Modelling Classification in Probabilistic Datalog

Probabilistic Logics allows to model more compact and shorter definitions than other approaches, minimising the gap w.r.t. the mathematical formulation. This fact implies that the processes of knowledge transfer and maintainability will be easier. Furthermore, this abstraction leads to the possibility of experts users modelling specific and complex information needs. The main challenges of this approach are the efficiency/scalability and the expressiveness. The reason for the latter is that the increase in abstraction also implies that certain operations cannot be modelled. Therefore, a balance between abstraction and expressiveness is needed. Moreover, the specific case of modelling classifiers presents the

additional difficulty of modelling a huge number of methods with various theoretical foundations where the different nature of the methods implies that some of the techniques are easier to model than others.

Before illustrating the modelling of classifiers, Tables 2 and 3 show a sample of data representation in tabular and probabilistic logical format for the relations tf_sum (normalised term-doc occurrences by the number of terms) and $part\_of$ which models the labelling of documents and classes.

tf_sum		
Value	Term	Document
0.23	economy	d40
0.52	expectation	d23
0.12	provider	d23
0.16	reuters	d1

part_of		
Value	Document	Class
1	d1	cocoa
1	d5	grain
1	d5	wheat
1	d5	oil

Fig. 2. Tabular Data Representation

```
1  0.23 tf_sum(economy, d40);
2  0.52 tf_sum(expectation, d23);
3  0.12 tf_sum(provider, d23);
4  0.16 tf_sum(reuters, d1);
```

```
1  part_of(d40, cocoa);
2  part_of(d23, grain);
3  part_of(d23, wheat);
4  part−of(d1, oil);
```

Fig. 3. Probabilistic Logical Data Representation

Figures 4 and 5 illustrate the general modelling of Naive-Bayes and k-NN classifiers (the modelling of cosine similarity is also shown as the similarity function). The modelling of Naive-Bayes requires the relation $term$[1] which models each word occurrence in a document and $part\_of$. Rules for representing the term and class space are specified (is_term, is_class), as well as the occurrences of terms in classes ($term\_class$) and all the combinations between terms and classes ($term\_class\_full$). There are two different estimations for $P(t|c)$ based on Laplace and minimum probability smoothing. In the first case the relation $term\_class$ is augmented[2] adding one extra occurrence for each term and class. The final $P(t|c)$ estimation is computed by using the bayes expression, dividing each tuple in $term\_class$ by the sum of all tuples with the same class. The minimum probability smoothing adds a fixed probability to estimation based on the term-class space. $P(d|c)$ is computed by aggregating all the tuples in p_t_c for a given test document using the PROD expression. The last steps are multiplying this value by the probability of the class and normalising of the results.

On the other hand, for the modelling of k-NN, relations $final\_test\_weight$ and $final\_weight$ model the importance of terms for testing and training documents. The specific weighting schema to be used is specified by the user (i.e. ltc). The first two rules of the modelling compute the euclidean normalisation

[1] Automatically created based on a set of documents.

[2] The same head in different rules implies that the tuples from both rules are united.

```
1  # preliminary and term−class relations
2  is_term FIRST(T) :− term(T, D);
3  is_class FIRST(T) :− part_of(D, C);
4  term_class(T, C) :− term(T, D) & part_of(D, C);
5  term_class_full (T, C) :− is_term(T) & is_class (C);

7  #Laplace estimation for P(t|c)
8  term_class_laplace (T,C) :− term_class(T,C);
9  term_class_laplace (T,C) :− term_class_full (T,C);
10 p_t_c_laplace SUM(T,C) :− term_class_laplace(T,C) | (C);

12 # Minimum probability estimation for P(t|c)
13 p_t_c_aux_min (T,C) :− term_class(T,C) | (C);
14 p_t_c_aux_min (T,C) :− minProb() & term_class_full(T, C);
15 p_t_c_min SUM(T,C) :− p_t_c_aux_min(T,C);

17 # Generic computation given P(t|c)
18 p_d_c PROD(D, C) :− test_term(T, D) & p_t_c(T,C);
19 p_c_d(C, D) :− p_c(C) & p_d_c(D, C);
20 score_nb(C, D) :− p_d_c SUM(D,C)|(D);
```

Fig. 4. Modelling of Naive-Bayes Family in PDatalog

for term-document weight w.r.t. the same document (tuples sharing the same D). The the last line for the cosine modelling computes the product of train and test documents sharing the same term. The final aggregation (*score_knn*) needs *top_similarity* to be customised, specifying the number of neighbours k and a similarity measure. For instance, cosine similarity with 45 neighbours (top_similarity(D1, D2) :- cosine(D1, D2):45).

The "Plug & Play" capabilities of our approach allow to use and customise any strategies or algorithm that has been modelled before with minimum engineering effort. For each of the algorithms modelled in the framework, a list of defined predicates is presented as a predicate dictionary (example given in Appendix B). In addition to the enumeration of available predicates, it specifies the requirements for making their computation possible. This includes the

```
1  # Cosine definition
2  vec1_norm(T, D) :− final_test_weight(T, D)|EUCLIDEAN(D);
3  vec2_norm(T, D) :− final_weight(T, D)|EUCLIDEAN(D);
4  cosine SUM(D1, D2) :− vec1_norm(T, D1) & vec2_norm(T, D2);

6  # k−NN score
7  score_knn SUM(C,D1) :− top_similarity(D1,D2) & part_of(D2,C);
```

Fig. 5. Modelling of k-NN PDatalog

mapping rules for customisation (i.e. specifying number of neighbours) and the relations that have to be defined (i.e. term relation).

Figure 6 illustrates this idea where the predicates are obtained from specific models for classification and other IR-related tasks. As a result, an expert user would be able to create specific (multi task) models using these predicates without any engineering effort using a high-level language increasing the productivity and adaptability. As en example, Figure 7 illustrates the high-level specification of a k-NN algorithm with *ltc* weights, cosine similarity and 45 neighbours.

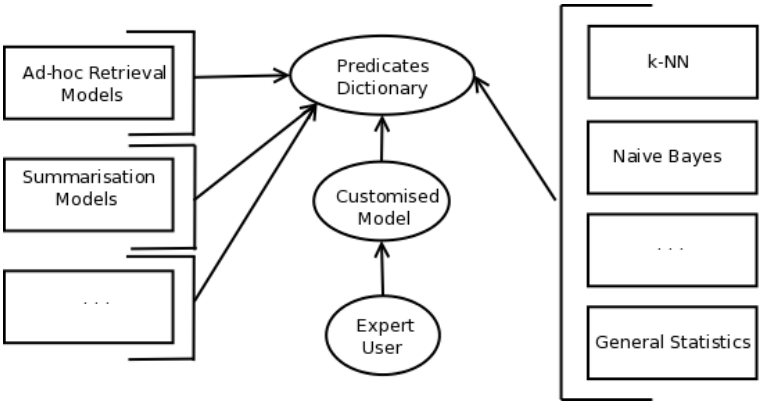


Fig. 6. Dictionary-Based Architecture

```
1  # Representation algorithms.
2  final_weight (T,D) :- ltc (T,D);
3  final_test_weight (T,D) :- test_ltc(T,D);

5  # Similarity based on cosine
6  _sort(cosine);
7  top_similarity (D1, D2) :- cosine(D1, D2):45;

9  # Final score
10 score(C, D) :- score_knn(C, D);
```

Fig. 7. Example of Strategy Customisation

3.1 Proving the Correctness of PD Programs

One of the benefits of high-level modelling is that not only the definitions are close to the mathematical formula but it is possible to analytically corroborate if they represent the same concept. Mathematical proof of the correctness of Naive-Bayes (with minimum probability estimation) and k-NN with cosine similarity are discussed in Propositions 1 and 2 respectively. Translations from PDatalog expressions to mathematical formulas is provided in Appendix B.

Proposition 1. *The modelling of Naive-Bayes using PDatalog, illustrated in Figure 4, is correct w.r.t. Equation 2 assuming a minimum probability for the cases where* $P(t|c) = 0$.

Proof. It is assumed that $P(t|c)$ is computed by using the maximum likelihood in the term class occurrences space $P(t|c) = \frac{n_L(t,c)}{\sum_i n_L(t_i,c)}$, assigning a minimum probability for the cases when its zero. This is modelled using the relational Bayes expression "|" over the *term_class_min* relation that has been computed by adding the *term_class* elements and (different rules with the same head are translated as a sum aggregation) the minimum value for all possible term-class tuples. The next step, after having $P(t|c)$ represented, is applying the product (using $PROD$) over the relation, computing $P(d|c)$. After this, only a product with $P(c)$ is needed (expression & in this PDatalog case).

Proposition 2. *The modelling of k-NN in PDatalog presented in Figure 5 is correct w.r.t. Equation 4.*

Proof. Cosine similarity is modelled as a product of the euclidean normalisation of test and train documents. In both cases, the normalisation is computed as $\frac{weight(t,d)}{\sqrt{\sum_i weight(t_i,d)^2}}$. This is modelled with the expression "$EUCLIDEAN$" which represents that the normalisation is done for those tuples sharing the same variable D. The relation *top_similarity* matches perfectly the function sim_k. The SUM expression aggregates all tuples with the same C and D1 variables. Therefore, it could be translated as a sum over tuples with different values for the variable D2 (d_j in the mathematical version). Finally, the fact that only values with a value of $part\_of$ are considered means that similarity is only computed if D2 belongs to class C which is directly translated into $\sum_{d_j \in C}$.

4 Feasibility Study

Experiments have been carried out using different real-scale collections and a variety of models, all of which have been modelled using a descriptive approach. The main goal is to empirically prove that our approach achieve comparable quality than other paradigms while maintaining reasonable efficiency levels.

4.1 Experiment Set-Up

Two traditional text classification collections have been used for the experiments: 20newsgroups and Reuters-21578. 20 Newsgroups[3] is a collection of approximately 20,000 newsgroup documents and 20 classes, some of them extremely similar (i.e. "comp.windows.x" and "comp.os.ms-windows.misc"), with almost uniform distribution of documents over classes. The split for the collection is based on time as it is suggested.

[3] Obtained from http://people.csail.mit.edu/jrennie/20Newsgroups/

Reuters-21578[4] contains structured information about newswire articles that can be assigned to several classes. The "ModApte" split is used and only documents that belong to classes with at least one train and one test documents are considered. As a result, there are 7770 documents for training and 3019 for testing, observing 90 classes with a highly skewed distribution over classes.

In both cases several feature selection measures, weighting schemas and variations of Naive-Bayes and k-NN algorithms are tested. The name of each row represents (in this order) feature selection measure and number of features (i.e. chi_2000) and model (i.e. knn_45_ltc_cosine). For k-NN, the model name includes the weighting schema and the similarity function. The quality achieved by each module is presented in micro and macro Precision/Recall break-even point. A (shared) server with four dual-core 3GHz Opteron and 32GB of RAM and the engine HySpirit [17] have been used for the executions. The average time per document in testing has been obtained, as it is usual, by averaging the time for classifying the testing documents one by one. Only one representative per model is represented in the table because changing the configurations almost does not have any impact in the efficiency.

Table 1. Quality of Classifiers

	20-newsgroups		**Reuters-21578**	
	mBEP	MBEP	mBEP	MBEP
chi_1000_bayes_log_laplace	62.63	63.88	70.28	50.84
chi_1000_bayes_log_min	63.25	64.76	72.2	53.15
chi_1000_knn_30_ltc_norm_cosine	68.18	68.13	79.33	60.93
chi_1000_knn_30_tfc_norm_cosine	66.69	66.08	80.77	62.77
chi_1000_knn_45_ltc_norm_cosine	68.1	67.78	78.99	59.95
chi_1000_knn_45_tfc_norm_cosine	65.89	65.85	80.43	64.51
chi_2000_bayes_log_laplace	63.44	65.97	71.48	48.27
chi_2000_bayes_log_min	63.53	65.55	73.19	49.57
chi_2000_knn_30_ltc_norm_cosine	**70.03**	**69.87**	81.36	62.8
chi_2000_knn_30_tfc_norm_cosine	67.9	67.76	**82.3**	64.85
chi_2000_knn_45_ltc_norm_cosine	69.57	69.32	80.59	59.95
chi_2000_knn_45_tfc_norm_cosine	67.64	67.78	82.16	**64.95**

4.2 Results

Our models achieve comparable quality results with values reported in the literature. It provides empirical confirmation for the model correctness. As expected, k-NN outperforms Naive-Bayes and the difference between micro and macro BEP for 20-newsgroups is minimal while it is significant for Reuters-21578 which have large differences between the number of documents per class.

Efficiency results show that the application of both algorithms is possible with reasonable training and testing times in both collections.

[4] Available at http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 2. Efficiency/Scalability of Classifiers in PDatalog

	20-newsgroups		**Reuters-21578**	
	train(min)	test(s/doc)	train(min)	test(s/doc)
NB	37	0.543	11	0.480
k-NN	27	0.552	9	0.873

5 Discussion and Future Work

This paper has shown the benefits of modelling classifiers using a descriptive approach. The compact high-level definitions and the use of a predicate dictionary leads to a flexible framework where expert users can model specific strategies with minimum engineering effort. In addition, it allows to prove the correctness of the models due to the fact that the abstraction makes possible to translate from the modelling in PDatalog to a mathematical formulation. Proofs have been presented for illustrating how the modelling of k-NN and Naive-Bayes result in the same equations as the mathematical concept. Experimental results empirically shows that the quality results achieved by our approach are the same as in the literature and that it is feasible to be used in real-scale environments.

Future work includes the modelling of more competitive text classification algorithms (i.e. SVM) that are not possible to be modelled at the moment, mainly because of the inexistence of optimisation expressions in PDatalog. With respect to correctness checking, the next step will be the investigation of an automatic derivation of mathematical expressions from a PDatalog program.

References

1. Cumbo, C., Iiritano, S., Rullo, P.: Reasoning-Based Knowledge Extraction for Text Classification. In: Suzuki, E., Arikawa, S. (eds.) DS 2004. LNCS (LNAI), vol. 3245, pp. 380–387. Springer, Heidelberg (2004)
2. Eisner, J., Goldlust, E., Smith, N.A.: Compiling Comp Ling: practical weighted dynamic programming and the Dyna language. In: Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pp. 281–290 (2005)
3. Forst, J.F., Tombros, A., Roelleke, T.: POLIS: A Probabilistic Logic for Document Summarisation. In: Proceedings of the 1st International Conference on the Theory of Information Retrieval (ICTIR 2007), pp. 201–212 (2007)
4. Frommholz, I., Fuhr, N.: Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In: Proceedings of Joint Conference on Digital Libraries (JCDL 2006), pp. 55–64 (2006)
5. Fuhr, N.: Probabilistic Datalog - a logic for powerful retrieval methods. In: Proceedings of the 18th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995), pp. 282–290 (1995)
6. Gelfond, M., Rushton, N., Zhu, W.: Combining Logical and Probabilistic Reasoning. In: Proceedings of AAAI 2006 Spring Symposium, pp. 50–55 (2006)
7. Hunter, A., Liu, W.: A survey of formalisms for representing and reasoning with scientific knowledge. The Knowledge Engineering Review 25, 199–222 (2010)

8. Lloyd, J.W.: Practical Advantages of Declarative Programming. In: Proceedings of Joint Conference on Declarative Programming, GULP-PRODE 1994 (1994)
9. Lopez, A.: Translation as Weighted Deduction. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), pp. 532–540 (2009)
10. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: Workshop on Learning for Text Categorization in AAAT/ICML 1998, p. 41 (1998)
11. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 298–307 (1993)
12. Nottelmann, H.: PIRE: An Extensible IR Engine Based on Probabilistic Datalog. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 260–274. Springer, Heidelberg (2005)
13. Nottelmann, H., Fuhr, N.: Learning Probabilistic Datalog Rules for Information Classification and Transformation. In: Proceedings of International Conference on Information and Knowledge Management (CIKM 2001), pp. 387–394 (2001)
14. Raedt, L.D., Kimmig, A., Toivonen, H.: ProbLog: a probabilistic Prolog and its application in link discovery. In: Proceeding of the International Joint Conference on Artificial Intelligence (JCAI 2007), pp. 2468–2473 (2007)
15. Roelleke, T., Fuhr, N.: Information retrieval with probabilistic Datalog. In: Crestani, F., Lalmas, M., Rijsbergen, C.J. (eds.) Uncertainty and Logics - Advanced Models for the Representation and Retrieval of Information. Kluwer Academic Publishers, Dordrecht (1998)
16. Roelleke, T., Wu, H., Wang, J., Azzam, H.: Modelling retrieval models in a probabilistic relational algebra with a new operator: The relational Bayes. VLDB Journal 17(1), 5–37 (2008)
17. Rolleke, T., Lubeck, R., Kazai, G.: The HySpirit retrieval platform. In: Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 454. ACM, New York (2001)
18. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34, 1–47 (2002)
19. Shen, W., Doan, A., Naughton, J.F., Ramakrishnan, R.: Declarative information extraction using datalog with embedded extraction predicates. In: VLDB 2007: International Conference on Very Large Data Bases, pp. 1033–1044 (2007)

A Mathematical Translation of PDatalog Expressions

Table 3 shows the mathematical translations of the PDatalog expressions that have been used in this paper. Let r_a, r_b be relations; let A and B be attributes sets and *m(A)* be the set of values v assigned to each attribute in A.

Table 3. Mathematical Translations of PDatalog Expressions

PDatalog expression	Mathematical Formulation
r_a FIRST(A) :- $r_b(B)$;	$r_b(B)_1$
r_a SUM(A) :- $r_b(B)$;	$\sum_{m(B) \subseteq m(A)} r_b(B)$
r_a PROD(A) :- $r_b(B)$;	$\prod_{m(B) \subseteq m(A)} r_b(B)$
r_a (A) :- $r_b(B)$|DISJOINT(K);	$\frac{r_b(B)}{\sum_{m(K) \subseteq m(B)} r_b(B)}$
r_a (A) :- $r_b(B)$|EUCLIDEAN(K);	$\frac{r_b(B)}{\sqrt{\sum_{m(B) \subseteq m(K)} r_b(B')^2}}$

B Predicate Dictionary Specification

```
term(T, D): Occurrences of terms in documents

part_of(D, C): Document-class labels

p_t_c_min(T, C): P(t|c) using minimum probability smoothing
minProbability() has to be specified

p_c_d_bayes(C, D):  Score for class-document using Naive-Bayes classifier
p_t_c(T, D) has to be specified for the estimation of P(t|c)

cosine(D1, D2): Similarity score based on cosine distance
final_test_weight(T, D) is needed for measuring the importance of terms in test documents
final_weight(T, D) is needed for computing the importance of terms in train documents

score_knn(C, D): Score for class-document using k-NN classifier
top_similarity(D1, D2) is needed modelling the k most similar documents.
```

Do Subtopic Judgments Reflect Diversity?

John A. Akinyemi and Charles L.A. Clarke

University of Waterloo, Canada

Abstract. Current measures of novelty and diversity in information retrieval evaluation require explicit subtopic judgments, adding complexity to the manual assessment process. In some sense, these subtopic judgments may be viewed as providing a crude indication of document similarity, since we might expect documents relevant to common subtopics to be more similar on average than documents sharing no common subtopic, even when these documents are relevant to the same overall topic. In this paper, we test this hypothesis using documents and judgments drawn from the TREC 2009 Web Track. Our experiments demonstrate that higher subtopic overlap correlates with higher cosine similarity, providing validation for the use of subtopic judgments and pointing to new possibilities for measuring of novelty and diversity.

1 Introduction

Several ongoing information retrieval evaluation efforts, including the TREC Web Track[1] and the NTCIR INTENT Task[2] focus on the evaluation of novelty and diversity. For the TREC Web Track, each evaluation topic is structured around a typical Web query. A number of subtopics are defined for the query, with each subtopic reflecting a distinct aspect or interpretation of that query. For example, subtopics associated with the query "tornadoes" (topic 75) address their causes, occurrences, forecasting, and fatalities, as well as requesting images and videos. Prior to submitting their experimental runs, Web Track participants are given a collection of Web documents and a set of queries, but not the subtopics associated with the queries. For each query, participants attempt to infer the diversity underlying the query and return a ranked list of documents that balances novelty against relevance [4]. After submission, assessors judge each document independently with respect to each subtopic. Results are reported using measures designed to evaluate novelty and diversity, such as α-nDCG [5], ERR-IA [2], and "intent aware" versions of traditional measures [1], all of which depend upon the availability of subtopic judgments.

In this paper, we investigate the relationship between measured document similarity and the subtopic judgments rendered by the assessors. If these judgments genuinely reflect diversity, the average similarity between documents relevant to same subtopic should be higher than the average similarity between documents

[1] plg.uwaterloo.ca/~trecweb

[2] www.thuir.org/intent/ntcir9

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 309–312, 2011.

that are relevant to different subtopics. By testing this hypothesis, we seek to provide validation for the use of subtopics to measure novelty and diversity in information retrieval evaluation. In addition, we hope to lay the groundwork to augment or replace explicit subtopic judgments with measured inter-document similarity values, providing a basis for using measured document similarity as an alternative to subtopic-by-subtopic judgments.

We see an obvious connection between our hypothesis and the venerable cluster hypothesis [6,7,8,9], which states that "closely associated documents tend to be relevant to the same requests" [9]. We extend the idea to the subtopic level, but with a focus on the relationship between the documents relevant to a given query. We expect that documents relevant to the same subtopic will tend to be more closely associated than documents relevant to different subtopics.

In the next two sections, we experimentally investigate our hypothesis. For each pair of documents relevant to a given query, we consider the degree of *subtopic overlap* between them, i.e. the number of subtopics for which both documents are relevant. We then compare this overlap against the traditional cosine similarity function. In effect we treat subtopic overlap between relevant documents as a crude similarity value. As the basis for our experiments, we use the topics and judgments from the TREC 2009 Web Track [3].

2 Method

We start with the `qrels` file provided by TREC 2009 Web Track's diversity task, which encodes the judgments for the task. This file contains a list of tuples, each composed of four fields: document id, topic number, subtopic number, and relevance judgment. Each tuple indicates that the given document is either relevant or not relevant to the given subtopic of the given topic. While the `qrels` file includes both relevant and non-relevant judgments, in this paper we exclude documents that were not judged relevant to at least one subtopic, since we aim to compare similarity between pairs of relevant documents.

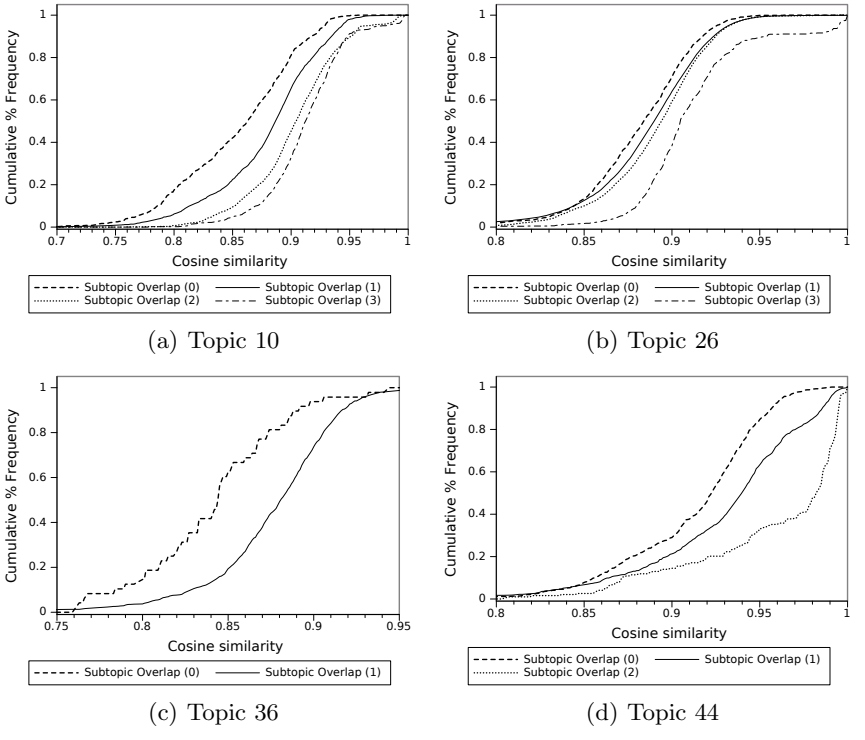
For each topic, we consider all pairs of documents relevant to at least one of the subtopics of that topic. For each pair, we compute two values: 1) standard cosine similarity, and 2) subtopic overlap, which indicates the number of relevant subtopics shared by the two documents. For the TREC 2009 Web Track documents, the subtopic overlap values range from 0 to 4, with most document pairs having subtopic overlap values of 0, 1 or 2.

3 Results

For our analysis, we focus on document pairs having subtopic overlap values of 0, 1 and 2, because only a very small number of document pairs have subtopic overlap values of 3 or 4. Our hypothesis suggests that larger cosine similarity values should correlate with larger subtopic overlap values. To compare these values, we compute the distribution of the cosine values for each topic with respect to the cumulative percentage frequency of their subtopic overlap values.

The plots in Figure 1 show the distribution of cosine similarity values for document pairs with different levels of subtopic overlap for four example topics. Each curve provides a cumulative distribution for a given level of subtopic overlap, where a specific point on the curve indicates the percentage of pairs with cosine values less than or equal to that value. All four examples support our hypothesis, with document pairs having higher subtopic overlap values consistently having higher cosine similarity values. For example, in Figure 1(d) more than 80% of pairs with overlap 0 have cosine similarity values falling below 0.95, while more than 65% of pairs with overlap 2 have values above 0.95.

Plots for most other topics follow the same trend, supporting our hypothesis and providing validation for the use of subtopics as an indicator of novelty. For each TREC 2009 topic we calculated the mean cosine similarity for document pairs with different overlap values. For 86% of TREC 2009 topics, documents pairs with overlap > 0 exhibited higher mean similarity than documents with overlap 0. For example, for topic 10, pairs with overlap 0 have a mean cosine similarity of 0.855, while pairs with overlap 1 have a mean cosine value of 0.880 and pairs with overlap 2 have a mean cosine value of 0.903. As a statistical test, we computed a paired t-test across the topics, comparing different levels of overlap. All p-values are $\ll 0.01$.



(a) Topic 10 (b) Topic 26

(c) Topic 36 (d) Topic 44

Fig. 1. Distribution of cosine similarity values for four example topics

4 Conclusions and Future Directions

In this paper, we demonstrate that document pairs having overlapping subtopics also tend to have higher similarity values when measured by standard cosine similarity. This result provides validation and support for the use of subtopic judgments to measure novelty and diversity in information retrieval evaluation. In the future, we hope to extend our experiments to other similarity measures and test collections.

As we noted earlier, subtopic overlap provides a crude measure of document similarity. Since current evaluation measures for novelty and diversity essentially measure similarity in this crude fashion, it may be possible to develop new measures of novelty and diversity that incorporate more traditional measures of similarity. Such measures might operate by combining manual assessments of broad topic relevance with automatic assessments of specific inter-document similarity, avoiding the need for explicit subtopics. Our work provides a first step in that direction.

To make this speculation a little more concrete, consider a ranked list of documents $\langle d_1, d_2, ... \rangle$. Let Z_k be the set of relevant documents above rank k. Let $sim(d_k, Z_k)$ be an appropriate measure of the similarity between a relevant document at rank k and the set of relevant documents above it. We might then replace the cascade gain value in a typical novelty measure [4] with

$$1 - f(sim(d_k, Z_k)),$$

where the function f serves to convert the similarity value into an appropriate loss value. We are actively exploring this idea in current research.

References

1. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: 2nd ACM WSDM, pp. 5–14 (2009)
2. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: 18th ACM CIKM, pp. 621–630 (2009)
3. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 web track. In: 18th TREC (2009)
4. Clarke, C.L.A., Craswell, N., Soboroff, I., Ashkan, A.: A comparative analysis of cascade measures for novelty and diversity. In: 4th ACM WSDM, pp. 75–84 (2010)
5. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: 31st ACM SIGIR (2008)
6. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In: 19th SIGIR. pp. 76–84 (1996)
7. Voorhees, E.M.: The cluster hypothesis revisited. In: 8th SIGIR, pp. 188–196 (1985)
8. Smucker, M.D., Allan, J.: A new measure of the cluster hypothesis. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 281–288. Springer, Heidelberg (2009)
9. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)

On Upper Bounds for Dynamic Pruning

Craig Macdonald[1], Nicola Tonellotto[2], and Iadh Ounis[1]

[1] School of Computing Science, University of Glasgow, Glasgow, UK
{craig.macdonald,iadh.ounis}@glasgow.ac.uk
[2] Information Science and Technologies Institute, CNR, Pisa, Italy
nicola.tonellotto@isti.cnr.it

Abstract. Dynamic pruning strategies enhance the efficiency of search engines, by making use of term upper bounds to decide when a document will not make the final set of k retrieved documents. After discussing different approaches for obtaining term upper bounds, we propose the use of multiple least upper bounds. Experiments are conducted on the TREC ClueWeb09 corpus, to measure the accuracy of different upper bounds.

1 Introduction

Instead of the exhaustive scoring of every document containing one or more query term, efficiency savings can be attained in an information retrieval (IR) system by deploying dynamic pruning strategies – such as MaxScore [2] and WAND [3] – which shortcut or omit entirely the scoring of documents that will not make the top k retrieved documents [1]. To facilitate pruning decisions, some strategies make use of upper bounds on the score that a term can achieve [3,4]. However, the accuracy of these term upper bounds naturally impacts on the attainable efficiency improvements – for instance, an upper bound that is too high will result in more documents being scored when they will never reach the set of top k retrieved documents.

Term upper bounds are normally pre-computed. In particular, for a given weighting model, each term's posting list [1,3] is fully scored at indexing time, and a single *least term upper bound* – the largest observed score for any document in the term's posting list – is recorded.

However, it is expensive at indexing time to score an entire index using the pre-determined weighting model to obtain the least upper bound for every indexed term. Instead, in [4], Macdonald et al. show that for various weighting models a *statistical approximate upper bound* can be obtained based only on statistics of the term, including the highest term frequency observed in the posting list. While a statistical approximate upper bound is obviously not as *tight* as the least upper bound, it has been shown to still provide sufficient information to dynamic pruning strategies to attain efficient retrieval [4].

However, we argue that maintaining a single upper bound score can produce *weak bounds*, limiting the potential efficiency of any pruning strategy using it. Consider the right hand side of example score distribution Figure 1(a). At a

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 313–317, 2011.

certain point during scoring, it is possible that the minimum score of the current top k documents is now sufficiently high that after this point no more postings for this term will make it into the top k. However, with a single upper bound, it cannot be asserted that none of the remaining documents can have a score as high as the current top k minimum score. Hence, a pruning technique must score at least one posting for all of these documents. Instead, if the dynamic pruning technique was informed that the remaining postings for the term have only a smaller upper bound, then they could be pruned more aggressively, or even skipped entirely.

Given this, it is clear that recording only a single least upper bound to represent the maxima across the score distributions of an entire term is not an accurate reflection of the actual likelihood of observing another higher scoring document in a posting list. Instead, in this work, we propose using *multiple least upper bounds* when modelling maxima in term score distributions[1]. For instance, in Figure 1(a), the same score distribution is also modelled by computing the least upper bound for three equal sized blocks of postings. By doing so, we can provide more information to the dynamic pruning strategy about the highest score that can be obtained in a block of postings. If the term upper bound is sufficiently low, then none of the remaining documents can make it into the top retrieved k set, and these postings may be skipped completely.

Note that fixed size blocks of max score are not the only option. Equally possible is the use of variable sized blocks. However, for reasons of brevity, we leave this as future work. In the remainder of this paper, we compare the accuracy of single and multiple least upper bounds.

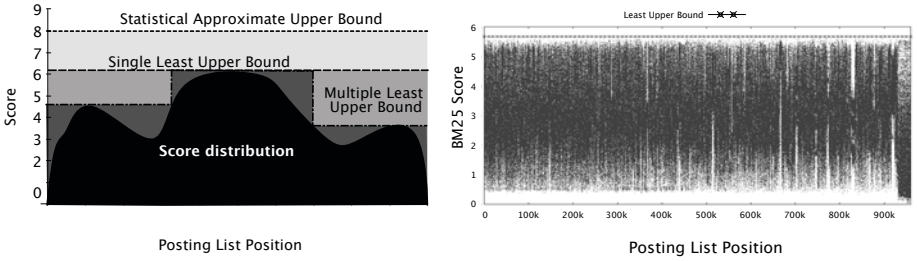
2 Experimental Setup

In the following section, we experimentally examine multiple term upper bounds, by addressing our main research question:
RQ 1. How accurate are multiple actual upper bounds compared to a single actual upper bound?

We follow previous works in considering a standard retrieval setting within our experiments. However, anchor text has become a key feature of modern Web search [6]. Hence, we also include our final research question:
RQ 2. Does the presence of anchor text impact on the attained accuracy?

To evaluate the accuracy of an upper bound, we use measures based on score distribution area, to indicate how 'tightly' an upper bound mirrors the real score distribution for a term's posting list. More accurate upper bounds provide more information about the score distribution of a query term to the dynamic pruning technique, thus positively benefiting overall efficiency. As the focus of this paper is on an analytical treatment of upper bounds to gain insights on this component alone, we assert that upper bound accuracy is correlated with query response

[1] The idea of using multiple upper bounds also briefly appears in [5]. However these upper bounds are encoded within the inverted index, and added independently from any characteristic of the posting lists and scores.



(a) Example (b) Term 'obama'

Fig. 1. Example and real score distributions

time [4], and consider timing experiments to be outwith the scope of this paper, leaving them as future work.

An (unachievable) term upper bound that perfectly predicts the maximum score attainable at any position in the posting list would, in essence, predict the actual score given a document (e.g. the black area in Figure 1(a)). We quantify this perfect case as the sum S_t of the scores for all postings for the term t. However, as the single least upper bound (σ_t) would cover a larger area, say E_t (e.g. the mid-grey region of Figure 1(a)). The ratio between areas E_t and S_t provide a measure accuracy of the term upper bound along the posting list I_t. We refer to this quantity as *overscore* O_t, which is calculated as follows: $O_t = \frac{E_t}{S_t}$, where $S_t = \sum_{d \in I_t} s_t(d)$ and $E_t = \sigma_t \times |I_t| - S_t$. As a percentage, overscore will be smaller for more accurate (tighter) upper bounds. Moreover, overscore can also be computed when using more than one term upper bound, by adjusting the computation of E_t to deal with portions of the posting list.

We use the TREC ClueWeb09 collection (cat. B) in our experiments to measure the overscore of the upper bounds. We index the content of the documents and the anchor text of the incoming hyperlinks using the Terrier IR platform[2], without removing stopwords nor applying any stemming. The occurrences of query terms are weighted using BM25 [7], with the parameters at their default settings. The behaviour of the upper bounds is measured with respect to a stream of 1,000 queries from the MSN 2006 query log [8].

3 Experimental Results

Table 1 reports the mean overscore for the query terms of the 1000 queries, calculated for different numbers of least term upper bounds. Results are shown for indices both with and without anchor text.

Firstly, it is clear from the results in Table 1 that by increasing the number of least upper bounds recorded for each query, the score distribution can be better modelled. Indeed, for RQ1, as the number of upper bounds are doubled, there is typically a 1% reduction in mean overscore.

[2] http://terrier.org

Table 1. Mean overscore for BM25 with different numbers of least upper bounds

Anchor Text	Single Least	2	4	8	16	32	64	128	256	512	1024
		Multiple Least									
✗	50.8%	50.6%	50.4%	50.2%	50.0%	49.6%	49.1%	48.5%	47.7%	46.7%	**45.6%**
✔	48.8%	48.7%	48.5%	48.3%	48.0%	47.6%	47.2%	46.6%	45.8%	44.8%	**43.8%**

For RQ2, comparing the results between indices without and with anchor text, we note that overscore is always smaller when anchor text is included in each indexed document. This is explained by the different 'spiky' nature of the score distribution for terms when anchor text is included. In particular, documents which have lots of anchor text for a given term will gain a score much higher than other documents, resulting in a spike in the score distribution. If this spike is larger than the maximum scores in other parts of the term's posting list, then overscore can be reduced by considering additional least upper bounds.

Overall, while it is not surprising that the score distribution is better modelled as the number least upper bounds increases, the margin of improvements are rather low. This suggests that in practice the score distributions of real terms are not as straightforward as our example in Figure 1(a). To illustrate this, Figure 1(b) shows the score distribution for the term 'obama'. From this figure, we note that there is a large variance of scores in the range [0.5,5.6]. However, with many documents achieving scores very close to the upper bound of 5.6, the difficulty in improving estimation by recording more least upper bounds recorded is clear.

4 Conclusions

This is an initial analytic study of the distributions of scores in posting lists, such that more efficient IR systems can be obtained by enhanced dynamic pruning strategies such as MaxScore. In particular, we proposed the recording of multiple least term upper bounds for each term, and compared these with a single least term upper bound with respect to their accuracy at modelling score distribution characteristics. We found that each doubling of the number of least upper bounds recorded for a posting list further reduced mean overscore by 1%. Indeed, obtaining term upper bounds are made difficult by the high variance of term scores in typical posting lists. In the future, we will study the resulting efficiency advantages of dynamic pruning strategies that take more accurate upper bounds into account, and whether it is possible to predict the appropriate number of upper bounds for each term.

References

1. Croft, W.B., Metzler, D., Strohman, T.: Search Engines – Information Retrieval in Practice. Addison-Wesley, Reading (2009)
2. Moffat, A., Zobel, J.: Self-indexing inverted files for fast text retrieval. Trans. on Information Systems 14(4), 349–379 (1996)

3. Broder, A.Z., Carmel, D., Herscovici, M., Soffer, A., Zien, J.: Efficient query evaluation using a two-level retrieval process. In: Proc. of CIKM 2003 (2003)
4. Macdonald, C., Ounis, I., Tonellotto, N.: Upper bound approximations for dynamic pruning. Trans. on Information Systems (in press, 2011)
5. Strohman, T.: Efficient Processing of Complex Features for Information Retrieval. PhD thesis (2007)
6. Hawking, D., Upstill, T., Craswell, N.: Towards better weighting of anchors. In: Proc. SIGIR 2005 (2005)
7. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In: Proc. of TREC-1 (1992)
8. Craswell, N., Jones, R., Dupret, G., Viegas, E. (eds.): Proceedings of the Web Search Click Data Workshop at WSDM 2009 (2009)

A Comparative Study of Pseudo Relevance Feedback for Ad-hoc Retrieval

Kai Hui[1], Ben He[1], Tiejian Luo[1], and Bin Wang[2]

[1] Graduate University of Chinese Academy of Sciences, Beijing 100190, China
huikai10@mails.gucas.ac.cn, {benhe,tjluo}@gucas.ac.cn
[2] Institute of Computational Technology, Beijing 100190, China
wangbin@ict.ac.cn

Abstract. This paper presents an initial investigation in the relative effectiveness of different popular pseudo relevance feedback (PRF) methods. The retrieval performance of relevance model, and two KL-divergence-based divergence from randomness (DFR) feedback methods generalized from Rocchio's algorithm, are compared by extensive experiments on standard TREC test collections. Results show that a KL-divergence based DFR method (denoted as *KL1*), combined with the classical Rocchio's algorithm, has the best retrieval effectiveness out of the three methods studied in this paper.

Keywords: Pseudo relevance feedback, Rocchio's algorithm, Divergence from randomness.

1 Introduction

Many PRF algorithms and methods have been proposed in the literature of information retrieval (IR). For example, *RM3* [4] derived from relevance model [3] improves the KL-divergence language model with Dirichlet smoothing (DirKL) [8], and the KL-based DFR feedback (*KL2*) [1,2] improves over the PL2 model [1]. Also based on the KL-divergence, an improved version of Rocchio's algorithm (*KL1*[1]) [7] is applied to enhance retrieval performance over BM25 [5].

Despite the effectiveness of PRF in improving the ad-hoc retrieval effectiveness, there exists a need for further understanding in the relative strength and weakness of different PRF methods [4]. Among the rare previous work, Lv & Zhai compare the effectiveness of relevance model to the model-based feedback [4]. This paper conducts a comparative study on the effectiveness of various popular PRF methods. While the work in [4] focuses on the PRF methods derived based on language model, this work compares the retrieval performance of RM3, KL1 and KL2, which have been previously applied on top of the DirKL, BM25, and PL2 weighting models, respectively. Note that the model-based feedback is not studied in this paper since its performance is comparable to RM3 [4].

[1] The PRF method applied in [7] is denoted as KL1 since it can be seen as a Type I model of the DFR feedback in [1].

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 318–322, 2011.

2 Related PRF Methods

In this section, we introduce the three PRF methods involved in this study. The algorithms of these PRF methods follow similar steps as described below, where the difference among them is explained:

1. There are two parameters in the PRF methods, namely $|ED|$, the feedback document set size, and $|ET|$, the number of expansion terms. The top-ranked documents in the first-pass retrieval form a feedback document set ED.
2. Each candidate term in ED is assigned an expansion weight. Different PRF algorithms apply their own weighting methods as follows.

RM3 estimates a feedback model $P(t|ED)$ for a candidate term t as follows:

$$P(t|ED) \propto \sum_{d \in ED} P(t|d)P(d) \prod_{q \in Q} P(q|d) \tag{1}$$

where $\prod_{q \in Q} P(q|d)$ is proportional to the relevance weight $w(Q, d)$, which indicates the relative importance of d in ED. $P(t|d)$ is the probability of generating t from the smoothed language model of document d. Moreover, for each d in ED, its relevance weight is aggregated by the $w(Q, d)$ of the top-ranked document to normalize the gap in the relevance weights among different feedback documents[2].

KL1 weighs a candidate term t by the KL-divergence of the term's distribution in each feedback document from its distribution in the whole collection:

$$w(t, ED) = \sum_{d \in ED} \frac{P(t|d) \log_2 \frac{P(t|d)}{P(t|C)}}{|ED|} \cdot w(Q, d) \tag{2}$$

where $P(t|d) \log_2 \frac{P(t|d)}{P(t|C)}$ is 0 if t is unseen in d. The relevance weight $w(Q, d)$ works as a quality-biased factor to balance between feedback documents with different importance [7].

KL2, similar to KL1, also uses the KL-divergence measure, but at a larger granularity by considering a candidate term's distribution in the entire feedback document set:

$$w(t, ED) = P(t|ED) \log_2 \frac{P(t|ED)}{P(t|C)} \tag{3}$$

3. Finally, the ET most weighted candidate terms, called expansion terms, are added to the original query.

RM3 uses an interpolation of the feedback model with the original query model with a free parameter α:

$$\theta_{Q'} = (1 - \alpha) * \theta_Q + \alpha * \theta_F \tag{4}$$

[2] The interpretation of RM3 follows the implementation in the Lemur toolkit.

Table 1. Information about the test collections used

Coll.	TREC Task	Topics	# Docs
disk1&2	1, 2, 3 ad-hoc	51-200	741,856
disk4&5	Robust 2004	301-450, 601-700	528,155
GOV2	2004-2006 Terabyte Ad-hoc	701-850	25,178,548

where θ_Q, θ_F and $\theta_{Q'}$ are the query model, feedback document model, and the modified query model, respectively.

Using both **KL1** and **KL2**, the vector of query terms weight is modified by taking a linear combination of the initial query term weights with the expansion weight $w(t, ED)$ as follows:

$$Q_1 = \alpha_1 * Q_0 + \beta_1 * \sum_{r \in ED} \frac{r}{R} \tag{5}$$

where Q_0 and Q_1 represent the original and first iteration query vectors, r is the expansion term weight vector, and R is the maximum $w(t, ED)$ of all candidate terms. α_1 and β_1 are tuning constants controlling how much we rely on the original query and the feedback information. In this paper, we fix α_1 to 1 to reduce the number of parameters that require tuning.

3 Experiments

We experiment on title-only ad-hoc topics over 3 standard TREC test collections as shown in Table 1. Porter's English stemmer, and standard English stopword removal are applied. On each collection, each of the baseline models and the corresponding PRF method are evaluated by a two-fold cross-validation, where the test topics associated to each collection are split into two equal-sized subsets by parity. Each pair of baseline model and PRF method has several free parameters that require tuning on the training topics. In our experiments, we first tune the length normalization/smoothing parameter using Simulated Annealing, and then, scan a wide range of values for the parameters $|ED|$, the feedback set size, and $|ET|$, the number of expansion terms, namely $2 < |ED| < 50$ and $10 < |ET| < 100$. Finally, the linear combination parameter that merges the expansion terms with the original query is tuned by Simulated Annealing.

The experimental results are summarized in Table 2. To examine the effectiveness of PRF with different evaluation purposes, the results are reported in three evaluation measures respectively: mean average precision (MAP), precision at 10 (Pre@10), and normalized discounted cumulative gain (nDCG). The best results obtained by the baseline models and the PRF methods are in bold. The improvement over the corresponding baseline model in percentage is also given in the table. Moreover, a $*$ or † indicates a statistically significant difference over DirKL+RM3 or PL2+KL according to the Wilcoxon matched-pairs signed-ranks test at 0.05 level.

Table 2. Experimental Results

Coll.	DirKL	PL2	BM25	DirKL+RM3	PL2+KL2	BM25+KL1
Results in MAP						
disk1&2	0.2351	0.2336	**0.2404**	0.2744, 16.72%	0.2814, 20.46%	**0.3036∗†, 26.29%**
disk4&5	0.2565	**0.2570**	0.2535	0.2832, 10.41%	0.2886, 12.30%	**0.2950∗, 16.37%**
GOV2	0.3028	**0.3042**	0.2997	0.3352, 10.70%	0.3227, 6.08%	**0.3434†, 14.58%**
Results in Pre@10						
disk1&2	0.4967	0.4986	**0.5106**	0.5266, 6.02%	0.5373, 7.78%	**0.5626∗†, 10.18%**
disk4&5	0.4400	**0.4420**	0.4405	0.4404, ≈ 0	0.4477, 1.29%	**0.4557, 3.45%**
GOV2	0.5617	0.5657	**0.5810**	0.5980, 6.46%	0.5758, 1.78%	**0.6053, 4.18%**
Results in nDCG						
disk1&2	0.4990	0.4978	**0.5018**	0.5390, 8.02%	0.5434, 9.16%	**0.5688, 13.35%**
disk4&5	0.5297	**0.5320**	0.5303	0.5592, 5.57%	0.5668, 6.54%	**0.5776, 8.92%**
GOV2	0.5924	**0.5960**	0.5876	**0.6110, 3.14%**	0.6036, 1.28%	0.6076, 3.40%

According to the results, the baseline models have in general comparable retrieval performance on all three test collections. As for the effectiveness of PRF, apart from on GOV2 in nDCG, BM25+KL1 provides the best retrieval performance on all three test collections used. The three PRF methods have shown comparable retrieval performance, although BM25+KL1 can lead to statistically significant better effectiveness on disk1&2 and disk4&5. Overall, out of the three PRF methods used, KL1, a DFR feedback method derived from Rocchio's algorithm, provides the best effectiveness on the datasets used. A possible explanation is that KL1 evaluates the importance of the candidate expansion terms in individual feedback documents separately. In this case, the effectiveness of KL1 has a less chance of being affected by poor feedback documents than KL2, while the latter could risk contaminating the feedback documents by considering the high-quality and poor feedback documents as a single sample from the collection.

4 Conclusions and Future Work

This paper has conducted a large-scale comparative study on the effectiveness of three popular PRF methods on standard TREC test collections. As shown by the experiments, KL1, a variant of the DFR feedback derived from the classical Rocchio's algorithm, has the best retrieval effectiveness on the datasets used. In the future, we plan to extend this study by including more recently proposed PRF methods, and by experimenting on larger test collections such as the ClueWeb dataset.

Acknowledgements. This work is supported in part by the President Fund of GUCAS.

References

1. Amati, G.: Probabilistic models for information retrieval based on divergence from randomness. PhD thesis, DCS, Univ. of Glasgow (2003)
2. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. 19(1), 1–27 (2001)
3. Lavrenko, V., Croft, W.B.: Relevance-Based Language Models. In: SIGIR, pp. 120–127 (2001)
4. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: CIKM, pp. 1895–1898 (2009)
5. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gatford, M., Payne, A.: Okapi at TREC-4. In: TREC (1995)
6. Rocchio, J.: Relevance feedback in information retrieval. In: The SMART Retrieval System. Prentice-Hall, Englewood Cliffs (1971)
7. Ye, Z., He, B., Huang, X., Lin, H.: Revisiting Rocchio's Relevance Feedback Algorithm for Probabilistic Models. In: Cheng, P.-J., Kan, M.-Y., Lam, W., Nakov, P. (eds.) AIRS 2010. LNCS, vol. 6458, pp. 151–161. Springer, Heidelberg (2010)
8. Zhai, C., Lafferty, J.D.: Model-based feedback in the language modeling approach to information retrieval. In: CIKM, pp. 403–410 (2001)

A Generic Data Model for Schema-Driven Design in Information Retrieval Applications

Hany Azzam and Thomas Roelleke

Queen Mary, University of London, UK. E1 4NS
{hany,thor}@eecs.qmul.ac.uk

Abstract. Database technology offers design methodologies to rapidly develop and deploy applications that are easy to understand, document and teach. It can be argued that information retrieval (IR) lacks equivalent methodologies. This poster discusses a generic data model, the Probabilistic Object-Oriented Content Model, that facilitates solving complex IR tasks. The model guides how data and queries are represented and how retrieval strategies are built and customised. Application/task-specific schemas can also be derived from the generic model. This eases the process of tailoring search to a specific task by offering a layered architecture and well-defined schema mappings. Different types of knowledge (facts and content) from varying data sources can also be consolidated into the proposed modelling framework. Ultimately, the data model paves the way for discussing IR-tailored design methodologies.

1 Introduction and Motivation

Nowadays, large-scale knowledge bases can be automatically generated from high-quality knowledge sources such as Wikipedia and other semantically explicit data repositories such as ontologies and taxonomies that explain entities (e.g. mark-up of persons, movies, locations and organisations) and record relationships (e.g. bornIn, actedIn and isCEOof). Such knowledge bases are leveraged by information retrieval (IR) application developers to develop more semantically-aware retrieval systems as opposed to systems that utilise text only. However, the developed systems are usually tailored to a particular data format and/or application. This is problematic since developing new applications or incorporating new data formats usually requires "reimplementing APIs, introducing new APIs, introducing new query languages, and even introducing new indexing and storage structures" [3].

The question, thus, becomes how diverse applications and data formats can be supported by a single unifying framework. Additionally, how techniques developed for a particular data format such as text can be easily transferred/extended to other data formats. This poster attempts to answer these two questions. We propose a generic data model that facilitates the development process of IR applications. The data model represents facts (e.g. objects and their relationships) and content knowledge (e.g. text in documents) in one congruent data model. The model can also be used to transfer text retrieval models such as TF-IDF, language modelling (LM) and BM25 to more knowledge-oriented retrieval models. Finally, the model can facilitate the expression of more complex and semantically expressive representations of information needs.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 323–326, 2011.

2 The Data Model

The proposed data model, the Probabilistic Object-Oriented Content Model (POOCM), combines 1) probability theory, 2) object-oriented modelling and 3) content-oriented modelling into one framework. The POOCM consists of term, classification, relationship and attribute propositions. Additionally, in order to perform content-oriented reasoning (traditional IR), each predicate has a context (context refers to documents, sections, databases and any other object with content).

A distinctive characteristic of this data model is that unlike standard artificial intelligence and database approaches content is modelled via a concept separate from the existing object-oriented concepts (classifications, relationships and attributes). This keeps the design tidy and captures the distinctive characteristics of each of the concepts, i.e. it enables the construction of evidence spaces based on each of the modelling concepts. The following representation of the movie "Apocalypse Now" illustrates the nature of the POOCM. The example shows two possible syntactic formulations: one based on predicate logic (e.g. Datalog), and the other similar to terminological logics [4].

```
        # Term 'vietnam' in movie_329171
0.5 vietnam(movie_329171);                  # movie_329171[0.5 vietnam]
        # Classification 'marlon_brando is an actor' in imdb
0.7 actor(marlon_brando, imdb);             # imdb[0.7 actor(marlon_brando)]
        # Classification 'walter_kurtz is a colonel' in movie_320971
colonel( walter_kurtz , movie_3209171);     # movie_329171[colonel( walter_kurtz )]
        # Relationship 'marlon_brando playsRoleOf walter_kurtz' in movie_329171
playsRoleOf(marlon_brando, walter_kurtz , movie_329171);
        # Attribute 'movie_329171 has release date 1979' in imdb
hasReleaseDate(movie_329171, 1979, imdb);   # movie_329171.hasReleaseDate(1979)
```

From an entity-relationship modelling point of view, the POOCM generally represents relationships between objects, relationships between classes and relationships between objects and classes. However, unlike the entity-relationship model, the POOCM incorporates content-oriented modelling techniques and concepts of probability theory which lead to a data model that is tailored to solving IR applications/tasks. The probabilities can be based on frequencies such as those commonly used in IR models.

The data model allows the handling of the physical data structures to remain transparent for (decoupled from) the rest of the system design, thus achieving what the DB field calls 'data independence' [2,3]. Furthermore, it enables the development of retrieval models that leverage the underlying data while remaining independent of the physical data representation. This is a desirable feature for designing complex retrieval systems as it ensures the independence of the developed retrieval models and query languages from the actual document representation [1].

Another benefit is that the object-oriented and content-oriented concepts of the POOCM provide the ability to instantiate retrieval models comprised of term, classification, relationship and attribute propositions. This leads to knowledge-oriented retrieval models that exploit a particular type of evidence explicated by the propositions. On the information need side, the data model can enrich query representation which facilitates the expression of more complex and expressive representations of information needs.

3 Modelling Layers

Application-independent and application-specific schemas can be instantiated from the generic POOCM. A simplified structure of the model distinguishes between three modelling layers: basic, structural and semantic layer. *Layer 0* (the basic layer) is *application-independent*, and the upper layers are more application-specific.

Generally speaking, overly specific schemas (e.g. fully flagged and normalised relational schemas as proposed by traditional DB design) and overly general schemas (triplet storages) are two extremes for IR. The POOCM does not argue that one approach is better than another, but demonstrates how application-specific schema layers can be derived from more general/basic ones.

Layer 1 is the *element-based* layer. It contains rules that can derive structural predicates from the L0, and the structural object Ids are made explicit. These rules can "lift" the basic classifications and attributes into structural classifications and attributes.

Layer 2 is the *entity-based* layer. It contains rules that derive semantic classification and relationships. For example, the rules extract objects by combining structural information about element types and their attributes. Such modelling of entities is prevalent in Entity-Relationship-graphs, such as RDF, where URIs are used to denote objects.

Fig. 1 highlights the main schema layers. These layers form an abstraction hierarchy that helps to achieve data independence. Any data format (e.g. XML, RDF, text) can be represented using the application-independent and application-specific schemas.

Another advantage of this layered approach is that explicitly stating how the basic and semantic layers are related can impact the modelling of probability estimations and aggregations. The predicates in the basic layer can, for example, be used to construct an evidence space for term-based retrieval models (e.g. LM) and for basic semantic models (e.g. attribute-based LM). In the structural and semantic layers, however, more complex and tailored (application-specific) models can be constructed while maintaining the advocated reusability and ability to be customised.

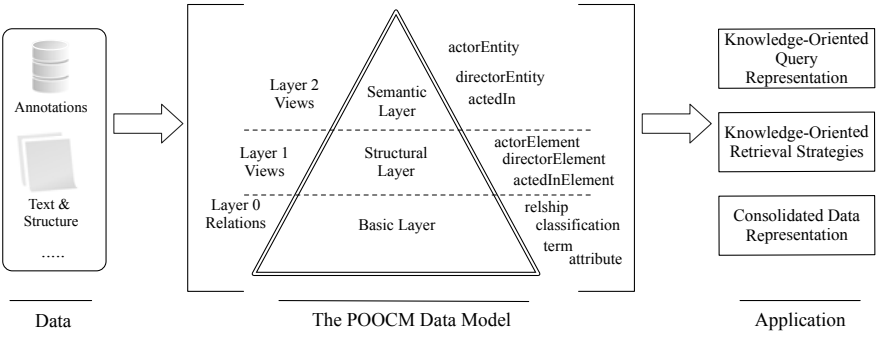


Fig. 1. POOCM Layers: Basic, Structural and Semantic

4 Probability Estimation

Probabilities used to develop IR models are an inherent component of the schema. The POOCM can guide which and how probabilities are estimated. For example, L1 consists of relations tailored to modelling the structure of data. This includes relations for context-based segmentations, e.g. term_doc(Term,Doc) to index documents, and term_sec(Term,Sec) to index sections. Probabilistic relations can be derived for each of the L1 relations. L2 comprises of relations reflecting the semantics of the data (semantic lifting of L1 leads to L2 relations). For instance, "actedIn(Actor,Movie,Context)" is L2. Note that L2 relation names bear a meaning, L1 relation names indicate the type of the context, and L0 relation names reflect classifications and relationships.

For each relation in each layer there are probabilistic relations for the sets of attributes. The probabilities can be value- or tuple-based. As such, the concepts of IR naturally apply to the semantic and generic schema. Concepts such as the tuple frequency of a class or the IDF of a class name make immediate sense. The following illustrates some of these probabilistic relations.

- $P_{\text{VF}}(t|i)$: Value-Frequency-based probability of term t derived from an index i such as "term(Term, Doc)" where the occurrence in documents (values) is the evidence.
- $P_{\text{TF}}(t|i)$: Tuple-Frequency-based probability of term t derived from an index i where the occurrence in locations (tuples) is the evidence.
- $P_{\text{IVF}}(t|i)$: IVF-based (IVF: inverse value frequency) probability of term t, e.g. $-\log P_{\text{VF}}(t|i) / \max(\{-\log P_{\text{VF}}(t'|i)\})$. For document retrieval IDF=IVF, and for actor retrieval InvActorFreq=IVF.

5 Conclusion

The generic data model (POOCM) advocated in this poster supports the design process when solving different IR tasks. The role of the model can be compared to what terminological logic [4] is for modelling knowledge: a conceptual quasi-standard that offers guidance while eschewing syntactical constraints. This poster aims at initiating a discussion about the role of the "design process" in IR - a process that so far has not been guided by an IR-tailored methodology. The hypothesis is that IR urgently needs such a methodology to respond to the growing need for the management of complex engineering processes and diverse content representations.

References

1. Cornacchia, R., de Vries, A.: A parameterised search system. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 4–15. Springer, Heidelberg (2007)
2. Fuhr, N.: Towards data abstraction in networked information retrieval systems. IP&M 35(2), 101–119 (1999)
3. Hiemstra, D., Mihajlovic, V.: A database approach to information retrieval: The remarkable relationship between language models and region models. CTIT Technical Report (2010)
4. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: SIGIR (1993)

A Query-Basis Approach to Parametrizing Novelty-Biased Cumulative Gain

Teerapong Leelanupab, Guido Zuccon, and Joemon M. Jose

School of Computing Science, University of Glasgow,
Glasgow, G12 8RZ, United Kingdom
{kimm,guido,jj}@dcs.gla.ac.uk

Abstract. Novelty-biased cumulative gain (α-NDCG) has become the *de facto* measure within the information retrieval (IR) community for evaluating retrieval systems in the context of sub-topic retrieval. Setting the incorrect value of parameter α in α-NDCG prevents the measure from behaving as desired in particular circumstances. In fact, when α is set according to common practice[1] (i.e. $\alpha = 0.5$), the measure favours systems that promote redundant relevant sub-topics rather than provide novel relevant ones. Recognising this characteristic of the measure is important because it affects the comparison and the ranking of retrieval systems. We propose an approach to overcome this problem by defining a safe threshold for the value of α on a query basis. Moreover, we study its impact on system rankings through a comprehensive simulation.

Keywords: diversity, sub-topic retrieval, effectiveness measure, web search.

1 Introduction

The purpose of an IR system is to respond to a given query with relevant documents so as to satisfy a information need. Nevertheless, queries posed by users are often inherently ambiguous and/or under-specified. Presenting redundant information may also be undesirable as users have to endure examining duplicate information repeatedly. Therefore, the IR system should present documents covering a complete combination of possible query-intents, in order to maximise the probability of retrieving relevant information (i.e. "provide complete coverage for a query"[2]). Such intents address several sub-topics of the information need and so they should be all retrieved; consequently, there is a need to avoid redundantly repeating them in the document ranking (i.e. "avoid excessive redundancy"[2]).

2 Analysis of α-NDCG

Clarke et. al. [1] proposed a modified version of normalised discounted cumulative gain, called α-NDCG, for evaluating novelty and diversity in search results.

[1] See http://plg.uwaterloo.ca/~trecweb/2010.html guidelines.
[2] Quote extracted from the TREC 2009 and 2010 Web Diversity Tracks guidelines.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 327–331, 2011.

Table 1. Five documents relevant to the sub-topics of query 26, "lower heart rate", from the TREC 2009 Web Diversity Track (*Left*), and corresponding evaluations of three imaginary system rankings, when α=**0.5** (*Right*)

Document ID	Sub-topic 1	2	3	4	Total
a. "en0001-55-27315"	1	-	1	1	3
b. "en0004-47-03622"	-	1	-	-	1
c. "en0001-69-19695"	1	-	1	1	3
d. "en0003-94-18489"	-	-	1	1	2
e. "en0000-31-13205"	-	-	-	-	0

system	r	doc	g(r)	ng(r)	dng(r)	dcng(r)	α-ndcg(r)	s-r(r)
A	1	a	3	3.0	3.0	3.0	1.0	0.75
	2	c	3	1.5	0.9	3.9	**1.0**	**0.75**
	3	e	0	0.0	0.0	3.9	0.9	0.75
B	1	a	3	3.0	3.0	3.0	1.0	0.75
	2	d	2	1.0	0.6	3.6	**0.9**	**0.75**
	3	e	0	0.0	0.0	3.6	0.8	0.75
C	1	a	3	3.0	3.0	3.0	1.0	0.75
	2	b	1	1.0	0.6	3.6	**0.9**	**1.00**
	3	e	0	0.0	0.0	3.6	0.8	1.00

Information needs are represented with respect to a query as sets of nuggets, or sub-topics. Consider a query Q with a total of $|S| > 1$ sub-topics. Let $J(d_r, s)$ be a graded relevant judgement indicating whether a document d_r at rank r is relevant to a sub-topic s or not. A duplication measure[3] $D_{s,r-1}$ is defined to monitor the degree of redundancy of documents ranked above r, given a sub-topic s. The measure has the role of quantifying the benefit of a document in a ranking, or what we call *novelty-biased gain*, $NG(Q, r)$:

$$NG(Q, r) = \sum_{s=1}^{|S|} J(d_r, s)(1 - \alpha)^{D_{s,r-1}} \tag{1}$$

where the parameter $0 < \alpha \leq 1$ represents the probability that a user is less likely to be interested in the sub-topic that is redundantly repeated by the document. In practice, this parameter is used to manipulate the reward of a document carrying novel information. To account for the late arrival of documents containing relevant sub-topics, the gain is discounted by a function of the rank position and then progressively cumulated[4]. The discounted cumulative gain at rank r is then normalised by that of the optimal ranking.

Table 1 (Left) shows five documents relevant to (some of) four sub-topics of query 26 belonging to the TREC 2009 Web Diversity Track. For the purpose of showing how an incorrect setting of α affects α-NDCG, we illustrate three imaginary system rankings (A, B, C), where the top three documents are ranked differently. In Table 1 (Right), the first column shows the rank position, (r), followed by document id, (doc), and the gain, g(r), wrt. sub-topic relevance. The next columns are the novelty-biased gain, ng(r), discounted novelty-biased gain, dng(r), discounted cumulative novelty-biased gain, dcng(r), its normalised gain, α-ndcg(r) when α=0.5, and finally sub-topic recall [3], s-r(r). Note that, while *a*-*b*-*c*-*d*-*e* is an ideal ordering of the documents, setting α to 0.5 produces a

[3] $D_{s,r-1} = \begin{cases} \sum_{i=1}^{r-1} J(d_i, s) & \text{if } r > 1 \\ 0 & \text{if } r = 1 \end{cases}$

[4] $DCNG(r) = \sum_{i=1}^{r} NG(Q, i)/log_2(1 + i)$

maximal gain, resulting in the *false* ideal document ranking *a-c-b-d-e*, which in turn is used when normalising, as shown in the table. If systems are evaluated according to α-NDCG with α=0.5, the following system rankings are obtained: $\{A, B, C\}$ or $\{A, C, B\}$. Note that system C obtains a lower α-NDCG than system A at positions 2 and 3 although at rank 2 it covers the only missing sub-topic (26.2), thus achieving complete sub-topic coverage (i.e. s-r(2)=1.0) earlier than A. In these circumstances α-NDCG with α=0.5 rewards documents containing *novel* relevant sub-topics *less* than *redundant* ones.

3 Deriving a Threshold for α

We consider the case where the gain obtained by a system retrieving novel relevant sub-topics, say system X, is expected to be higher than the gain of a system retrieving only redundant sub-topics, say system Y.

Let s^* be a novel relevant sub-topic[5], and s a redundant relevant sub-topic. At rank position r, in the worst case scenario (i.e. when system X retrieves only a single *novel* relevant sub-topic whereas system Y retrieves the remainder $|S|-1$ relevant but *redundant* sub-topics) system X should have higher α-NDCG than system Y. Thus, since we expect $NG_X(r) > NG_Y(r)$, we can rewrite this as:

$$J(d_r, s^*) \cdot (1-\alpha)^{D_{s^*,r-1}} > \sum_{s}^{|S|-1} J(d_r, s) \cdot (1-\alpha)^{D_{s,r-1}}$$

This inequality can be used to define boundaries on α so that the inequality is true, i.e. a system retrieving novel relevant sub-topics is awarded with an higher α-NDCG than a system retrieving redundant sub-topics. At this stage we make a simplifying assumption, following the relevance judgements that have been collected in the TREC Web Diversity track: we assume a binary decision schema regarding the relevance of documents to each sub-topic. Therefore:

$$(1-\alpha)^{D_{s^*,r-1}} > \sum_{s}^{|S|-1} (1-\alpha)^{D_{s,r-1}}$$

and with a further assumption that the $D_{s,r-1}$ of all redundant relevant sub-topics are identical, we obtain

$$(1-\alpha)^{D_{s^*,r-1}} > (|S|-1) \cdot (1-\alpha)^{D_{s,r-1}}$$

Let $\beta = D_{s,r-1} - D_{s^*,r-1}$ be the difference in redundancy[6]. Note that β is always an integer when relevance judgements are binary. Thus, we can resolve wrt. α, ignoring the case $\alpha < 1 + \left(\frac{1}{|S|-1}\right)^{1/\beta}$, as $\alpha < 1$ by definition:

[5] Or the sub-topic with smaller degree of redundancy.

[6] Measuring a relative amount of novel information in documents, where redundant sub-topics have higher degree of redundancy than novel sub-topics, i.e. $D_{s,r-1} > D_{s^*,r-1}$, and thus $\beta > 0$.

$$(st): \qquad \alpha > 1 - \left(\frac{1}{|S|-1}\right)^{1/\beta} \tag{2}$$

Eq (2) is the necessary and sufficient condition that has to be satisfied if we expect α-NDCG to reward systems retrieving novel relevant sub-topics more than systems retrieving redundant sub-topics. Figure 1 shows the safe threshold (st) on α according to Eq (2) for varying circumstances, suggesting that considering values of α *below* or *equal to* the threshold (inside highlighted areas) can lead to an unexpected behaviour of the measure. That is, if α=0.5 for all the information needs, α-NDCG may misjudge documents conveying novel information. This is crucial, in particular, when analysing *high quality*[7] ranking results @2, @3, etc., or when the redundancy difference of the rankings (β) at lower positions is small. For queries containing 2 or less sub-topics this problem does not occur, as α=0.5 is greater than the safe threshold.

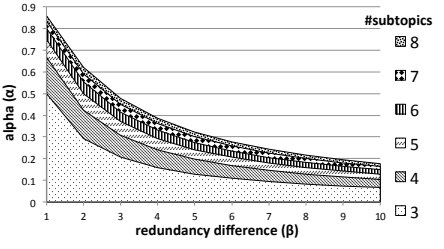


Fig. 1. Values of the safe threshold for α

4 Remarks and Conclusion

To verify the impact of setting α according to the proposed threshold, we conducted a comprehensive study by simulating system rankings using relevance judgements from the TREC 2009-2010 Web Diversity Tracks. We used the Fisher-Yates shuffle algorithm (with 100 re-starts) to generate 6! (factorial) random samples of all possible permutations of relevant document rankings. Rankings were then evaluated according to sub-topic recall and α-NDCG @10 with α=0.5, and α=st+0.01 where β=1 to avoid possible undesired scenarios. Figure 2 presents the Kendall's τ correlation of system rankings between sub-topic recall and the two different settings of α for 25 example queries (out of 98 total queries). Setting α>st produces rankings of systems based on α-NDCG that are significantly [8] more correlated to the ones obtained using sub-topic recall[9] than those obtained with α=0.5. These results are consistent over all the

[7] i.e. when a high number of relevant documents are ranked within the early ranking positions.

[8] Measured by a 1 tail t-test (p<0.01) and indicated by * in Figure 2.

[9] Correlation pairs are relatively low. This is because once complete sub-topic coverage is achieved at position r, the value of sub-topic recall for ranks >r is always 1. Therefore, sub-topic recall is unable to measure the utility of ranking in such circumstances.

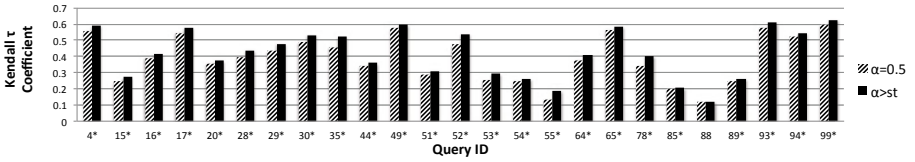


Fig. 2. Kendall's τ coefficient for 25 queries wrt. sub-topic recall and α-NDCG for α=0.5 and α >st

query set. Although the use of correlation with sub-topic recall as a mean to assess whether a measurement is better than another might be criticised, we believe that this can provide an indication of the measure behaviour, in particular because the intent of the Diversity Tracks is to provide complete coverage of all sub-topics.

In summary, by setting α on a query basis according to the safe threshold of Eq (2), the diversity of document rankings can be correctly measured without recurring to further modify α-NDCG, as suggested in [2].

References

1. Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR 2008, pp. 659–666 (2008)
2. Sakai, T., Craswell, N., Song, R., Robertson, S., Dou, Z., Lin, C.: Simple Evaluation Metrics for Diversified Search Results. In: EVIA 2010, pp. 42–50 (2010)
3. Zhai, C.X., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR 2003, pp. 10–17 (2003)

Investigating Query-Drift Problem from a Novel Perspective of Photon Polarization

Peng Zhang[1], Dawei Song[1], Xiaozhao Zhao[2], and Yuexian Hou[2]

[1] School of Computing, The Robert Gordon University, United Kingdom
[2] School of Computer Sci & Tec, Tianjin University, China
{p.zhang1,d.song}@rgu.ac.uk, {0.25eye,krete1941}@gmail.com

Abstract. Query expansion, while generally effective in improving retrieval performance, may lead to the query-drift problem. Following the recent development of applying Quantum Mechanics (QM) to IR, we investigate the problem from a novel theoretical perspective inspired by photon polarization (a key QM experiment).

1 Introduction

Query expansion usually improves overall retrieval performance [1]. However, some expanded query may shift from the underlying intent of the original query, leading to the query-drift problem [6]. As a result, for some individual queries, the performance of the expanded query can be inferior to that of the original one. Motivated by the emerging research in applying Quantum Mechanics (QM) as a new IR formalism [3], we investigate the query-drift problem from a novel perspective of photon polarization [4], which has recently inspired a new model [5] to re-rank the top n (e.g. 50) documents obtained from the first-round retrieval. In this paper, our focus is on the query-drift problem with the expanded query.

The photon polarization experiment [4] involves the probability measurement of photons that can pass through a polarization filter. We can view documents as photons, and the retrieval process as measuring the probability of each document that can pass through the query's retrieval filter (as polarization filter). Then, the measured probability can be regarded as the estimated probability of relevance of each document. This QM experiment usually inserts an additional filter between the original filter and the photon receiver (e.g. a screen). Similarly, in query expansion, the expanded query is constructed for the second-round retrieval.

In QM, the probability that a photon can pass through an additional filter is the combined effect of probability measurement on both filters (i.e., the original and the additional ones). This inspires us, in IR, to fuse (i.e. combine) the retrieved results from the original query and the expanded one. Indeed, such fusion-based method has been shown to be an effective approach to tackling the query-drift problem [6]. Photon polarization provides a new perspective and a novel mathematical framework to look at the problem by considering the representation of the additional filter under the same basis as the original filter. This means that the expanded query can be implicitly observed with respect to the

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 332–336, 2011.

original one. In this paper, we formulate the query expansion under the QM and derive a novel fusion approach to alleviating the query-drift problem.

2 Quantum-Inspired Approach

2.1 Photon Polarization

We first briefly introduce the idea of photon polarization [4]. A photon's state can be modeled by a unit vector $\varphi = a\,|{\rightarrow}\rangle + b\,|{\uparrow}\rangle$, which is a linear combination of two orthogonal basis vectors $|{\rightarrow}\rangle$ (horizontal polarization) and $|{\uparrow}\rangle$ (vertical polarization). The amplitudes a and b are complex numbers such that $|a|^2+|b|^2 = 1$. Suppose the original filter is a horizontal polarization filter. Each photon will be measured by the basis $|{\rightarrow}\rangle$ and the probability is $|a|^2$, i.e., the squared norm of corresponding amplitude a in the horizontal direction. After the measurement, the photon's state will collapse to the original basis vector $|{\rightarrow}\rangle$. If we now insert an additional filter (e.g. with direction $\nearrow$ of 45-degree angle), then the new basis vectors become $|{\nearrow}\rangle$ and its orthogonal counterpart $|{\nwarrow}\rangle$.

2.2 QM-Inspired Fusion Approach

In the first-round retrieval, under the QM formulation, a document d's state can be formulated as:

$$|\varphi_d\rangle = a_d\,|q\rangle + b_d\,|\neg q\rangle \tag{1}$$

where q is the original query, $|q\rangle$ denotes the basis vector for relevance, $|\neg q\rangle$ denotes the basis for irrelevance which is orthogonal to $|q\rangle$, and $|a_d|^2 + |b_d|^2 = 1$. $|a_d|^2$ can denote the estimated relevance probability of the document d with respect to q. If we do not consider the state collapse after the first-round retrieval, d's state with respect to the expanded query q^e can be represented as

$$|\varphi_d^e\rangle = a_d^e\,|q^e\rangle + b_d^e\,|\neg q^e\rangle \tag{2}$$

where $|a_d^e|^2 + |b_d^e|^2 = 1$ and $|a_d^e|^2$ denotes the estimated relevance probability of document d with respect to q^e.

To prevent query-drift, the existing fusion models in [6] directly combine two probabilities $|a_d|^2$ and $|a_d^e|^2$. This direct combination ignores the theoretical fact that the two probabilities are under different basis, i.e. $|q\rangle$ and $|q^e\rangle$, respectively.

In this paper, we propose to fuse $|a_d|^2$ and $|a_d^e|^2$ on the same basis. First, to connect different basis $|q\rangle$ and $|q^e\rangle$, let $|q^e\rangle = a_{q^e}\,|q\rangle + b_{q^e}\,|\neg q\rangle$, where $|a_{q^e}|^2 + |b_{q^e}|^2 = 1$. Assuming that the amplitudes in Eq. 1 and Eq. 2 have been estimated, a_{q^e} can be estimated by solving the equation $|\varphi_d\rangle = |\varphi_d^e\rangle$ (see Eq. 1 and 2). If we consider the collapse of $|\varphi_d\rangle$ to $|q\rangle$ after the first-round retrieval, another equation $|q\rangle = a_d^f\,|q^e\rangle + b_d^f\,|\neg q^e\rangle$ needs to be solved too, using the estimate of a_{q^e}. The a_d^f here denotes the fused amplitude on the basis $|q^e\rangle$. The process of solving the above equations is omitted due to the space limit. The solution is that $a_d^f = a_d a_d^e + b_d b_d^e$. The amplitudes b_d and b_d^e correspond to the irrelevance basis and often lead to unstable performance in our experiments.

For the purpose of this paper, we drop the term $b_d b_d^e$ in a_d^f. Nevertheless, we will investigate its effect in more detail in the future. Then, we have

$$a_d^f = a_d a_d^e \tag{3}$$

Let $|a_d^f|^2 = |a_d|^2 \cdot |a_d^e|^2$ denote the fused relevance probability, which considers both $|a_d|^2$ (see Eq. 1) and $|a_d^e|^2$ (see Eq. 2), on the same basis $|q^e\rangle$. For each document d, $|a_d|^2$ and $|a_d^e|^2$ can be estimated as the normalized scores by a retrieval model for the original query q and the expanded query q^e, respectively.

It is also necessary [6] to define two functions $\delta_q(d)$ and $\delta_{q^e}(d)$, the value of which is 1 if d is in the result list of the corresponding query, and 0 otherwise. Then, based on Eq. 3, we propose two QM-inspired Fusion Models (namely QFM1 and QMF2), as formulated in Tab. 1. Two existing fusion models in [6], namely combMNZ and interpolation, are re-formulated in Tab. 1 for comparison. The combMNZ and interpolation are additive (i.e. adding up two scores $|a_d|^2$ and $|a_d^e|^2$), while the QM-based models are multiplicative. In QMF2, the smaller η can make scores of different documents retrieved for q^e more separated from each other, leading to more distinctive scores. In interpolation model, the smaller λ, the more the fused score is biased to the second-round score (i.e. $|a_d^e|^2$).

Table 1. Summary of Fusion Models

Model	**Fused Score for each** d
combMNZ	$(\delta_q(d) + \delta_{q^e}(d)) \cdot (\delta_q(d)\|a_d\|^2 + \delta_{q^e}(d)\|a_d^e\|^2)$
interpolation	$\lambda\delta_q(d)\|a_d\|^2 + (1-\lambda)\delta_{q^e}(d)\|a_d^e\|^2 \quad (0 \le \lambda \le 1)$
QFM1	$(\delta_q(d)\|a_d\|^2) \cdot (\delta_{q^e}(d)\|a_d^e\|^2)$
QFM2	$(\delta_q(d)\|a_d\|^2) \cdot (\delta_{q^e}(d)\|a_d^e\|^2)^{1/\eta} \quad (\eta > 0)$

3 Empirical Evaluation

Experimental Setup. Our experiments are constructed on four TREC collections (see Tab. 2). The title field of TREC topics is used as the original query q. Lemur 4.7 is used for indexing and retrieval [2]. The Dirichlet prior for smoothing language model is set as default 1000. Top 50 documents from the first-round retrieval are used for constructing the expanded query q^e (with 100 terms) by the Relevance Model (RM) [1]. For both q and q^e, the negative KL-divergence model [2] is adopted as the retrieval model and 1000 documents are retrieved. In both cases, the normalized score is computed by $\exp\{-D\}/Z$, where $\exp\{-D\}$ is to transform the negative KL-Divergence $(-D)$ into the interval $(0, 1)$, and Z as a normalization factor is the sum over all the transformed scores.

The Mean Average Precision (MAP) is used as the effectiveness measure, and the Wilcoxon significance test is used to compute the statistical significance. A robustness measure, i.e. $<$*Init* as used in [6], is adopted to test the percentage of queries for which the (M)AP drops after the query expansion.

Experimental Results. From Tab. 2, we can observe that, on ROBUST2004 and WT10G collections, for almost 50% queries (see $<$*Init.*), the performance

Table 2. Experimental Results. The smaller $<Init$ generally means more robust performance. Statistical MAP improvements (at significance level 0.05) over Init.Rank. and RM are marked with α and β, respectively.

Collections	**WSJ8792**		**AP8889**		**ROBUST2004**		**WT10G**	
Topics	Topics 151-200		Topics 151-200		Topics 601-700		Topics 501-550	
Metrics	MAP(%)	$<Init$(%)	MAP(%)	$<Init$(%)	MAP(%)	$<Init$(%)	MAP(%)	$<Init$(%)
Init. Rank.	31.27	–	30.58	–	28.80	–	20.22	–
RM	37.75^{α}	22	$\mathbf{39.74}^{\alpha}$	28	32.82^{α}	44	21.72	46
combMNZ	35.76^{α}	10	35.42^{α}	**12**	32.60^{α}	**19**	$23.31^{\alpha\beta}$	26
QFM1	36.87^{α}	**8**	36.12^{α}	14	32.81^{α}	21	$23.69^{\alpha\beta}$	**24**
interpolation	$38.84^{\alpha\beta}$	14	39.53^{α}	16	$34.47^{\alpha\beta}$	29	$24.38^{\alpha\beta}$	30
QFM2	$\mathbf{39.01}^{\alpha\beta}$	16	39.20^{α}	18	$\mathbf{34.90}^{\alpha\beta}$	30	$\mathbf{24.58}^{\alpha\beta}$	30

of query expansion by RM is inferior to that of initial retrieval. All the fusion-based models can improve the robustness of query expansion. Two parameter-free models, i.e., combMNZ and QFM1, performs better than other two models in terms of robustness. QFM1 outperforms combMNZ in terms of MAP. On the other hand, QFM2 achieves a competitive performance in comparison with the interpolation model in terms of both MAP and robustness. For the interpolation model, we selected the best performing λ in the interval [0.1, 0.9] with step 0.1 for each collection, since we find that the model is sensitive to λ. For QFM2, we fix the η value as 0.1 and the performance is stable on all collections.

4 Conclusion

In this paper, we propose to investigate query expansion from a novel theoretical perspective inspired by the photon polarization in QM, and accordingly we have developed a novel fusion approach to alleviating the query-drift problem. The proposed models have been shown to largely improve the effectiveness and the robustness of a standard query expansion model. The performance is also comparable to two state-of-the-art fusion-based methods [6], shedding light on a promising new angle and mathematical formalism for further investigation.

Acknowledgments. This research is funded in part by the UK's EPSRC (EP/F014708/2), the China's NSFC (61070044) and the EU's Marie Curie Actions-IRSES (247590).

References

1. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: SIGIR 2001, pp. 120–127 (2001)
2. Ogilvie, P., Callan, J.: Experiments using the lemur toolkit. In: TREC 2001, pp. 103–108 (2001)
3. Piwowarski, B., Frommholz, I., Lalmas, M., van Rijsbergen, C.J.: What can quantum theory bring to information retrieval. In: CIKM, pp. 59–68 (2010)

4. Rieffel, E.G., Polak, W.: An introduction to quantum computing for non-physicists. ACM Comput. Surveys 32, 300–335 (2000)
5. Zhao, X., Zhang, P., Song, D., Hou, Y.: A novel re-ranking approach inspired by quantum measurement. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 721–724. Springer, Heidelberg (2011)
6. Zighelnic, L., Kurland, O.: Query-drift prevention for robust query expansion. In: SIGIR, pp. 825–826 (2008)

Using Emotion to Diversify Document Rankings

Yashar Moshfeghi, Guido Zuccon, and Joemon M. Jose

School of Computing Science
University of Glasgow, Scotland, UK
{yashar,guido,jj}@dcs.gla.ac.uk

Abstract. The aim of this paper is to investigate the role of emotion features in diversifying document rankings to improve the effectiveness of Information Retrieval (IR) systems. For this purpose, two approaches are proposed to consider emotion features for diversification, and they are empirically tested on the TREC 678 Interactive Track collection. The results show that emotion features are capable of enhancing retrieval effectiveness.

1 Introduction

Emotion is considered to be an important factor influencing overall human behaviour, including rational tasks such as reasoning, decision making, communication and interaction. Although emotion is subjective, it is presented in some objectively deducible ways in written documents [1]. News and user-generated content such as blogs, reviews, and tweets contain emotionally rich data and several studies have attempted to automatically extract these features from such data [1]. The use of emotion features has been shown to improve retrieval system effectiveness in collaborative search [2]. However, the effectiveness of emotion features when diversifying document rankings has yet to be studied.

Given a query, IR systems generate rankings according to the relevance of documents. Diversity in the ranking results has been shown to be useful in improving the effectiveness of IR systems. This is because diversity avoids redundancy, resolves ambiguity and effectively addresses users' information needs [3].

Diversity has been addressed through mathematical models [4] and through the use of external evidence [5]. We propose to use emotional features to enhance the diversity of the retrieved results. We believe that emotion features serve as beneficial information for diversifying document rankings. This is motivated by the fact that IR systems strive to gather conceptual information about a document through an indexing process, e.g., by representing documents as a bag of words. However, such a process ignores the fact that documents are not only vehicles for transmitting information, but also convey meanings and emotion. Here we focus on emotion and propose that diversifying document rankings based on emotion features allows us to better overcome this issue. We posit that relevant documents belonging to different subtopics may differ with respect to their conveyed emotion. For example, documents relevant to subtopic "diseases entering UK" of topic 352i ("British Chunnel impacts") imply different emotion

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 337–341, 2011.

than documents relevant to "increased tourism anywhere on British island": we thus expect that diversifying document rankings based on emotion will yield improvements in performance.

2 Approach

In the following, we outline the diversification approaches used in this work and discuss how emotion features are blended together with estimations of document relevance. Then the emotion extraction technique is explained.

2.1 Diversifying Document Rankings

In order to diversify document rankings, we adopt Maximal Marginal Relevance (MMR) [4] as it is an effective and popular approach. Let $sim(d, q)$ denote a measure of similarity between document d and query q; this can be regarded as a measure of relevance of d to q. Also let $esim(e(d), e(d'))$ represent the similarity between the emotion vector representations (see Section 2.2) of documents d and d'. We consider the situation where $|R|$ documents have been ranked, and the ranking function considers which document has to be ranked next. Following MMR, the next document to be ranked (i.e., d^*) is selected such that:

$$d^* = \arg\max \left[\lambda sim(d, q) - (1 - \lambda) \max_{d' \in R} esim(e(d), e(d'))\right]$$

where λ is a parameter that controls the impact of emotion similarity on the selection of document d^*: if $\lambda = 1$, emotion similarity has no impact on the selection of documents; while if $\lambda = 0$, emotion similarity is the only criterion used for ranking documents.

We further generalise the MMR approach such that the similarity between the candidate document and the query is interpolated with the *average* emotion similarity between the candidate document and those that have been ranked at previous positions. Thus, under the average interpolation approach (AVG-INT), d^* is ranked at rank position $|R| + 1$ if

$$d^* = \arg\max \left[\lambda sim(d, q) - (1 - \lambda) \sum_{d' \in R} \frac{1}{|R|} esim(e(d), e(d'))\right]$$

In contrast to MMR, the AVG-INT approach considers the average similarity between a candidate document and documents ranked in the previous $|R|$ ranks.

Several similarity functions can be used for computing $esim(e(d), e(d'))$. We test our ranking strategies using the cosine similarity and Pearson's correlation as similarity function to measure document relationships with respect to emotion. Other measures can be used (e.g., KL divergence, L1 norm, etc.): we plan to investigate the impact of different functions on empirical results in future works.

Table 1. α-nDCG values of Language Model (LM), MMR with text features (MMR(t)), MMR with emotion features (MMR(e)) and AVG-INT with emotion features (AVG-INT(e)) are reported and percentages of improvement over LM are presented in brackets. The best performing approach at each rank is highlighted in bold. Due to space constraints, for MMR(t) we only report results when re-ranking the top 20 documents: other settings obtain results that exhibit similar trends. Performance of AVG-INT(t) and MMR(t) are similar, and we therefore report the latter.

		LM	*MMR(t)*	*MMR(e)*			*AVG-INT(e)*		
			Top 20	Top 20	Top 50	Top 100	Top 20	Top 50	Top 100
α-nDCG	@5	0.520	0.554 (+7%)	**0.568 (+9%)**	0.555 (+7%)	0.545 (+5%)	0.561 (+8%)	0.559 (+8%)	0.539 (+4%)
	@10	0.532	0.559 (+5%)	0.560 (+5%)	**0.567 (+6%)**	0.551 (+4%)	0.554 (+4%)	0.555 (+4%)	0.547 (+3%)
	@20	0.545	0.556 (+2%)	0.556 (+2%)	0.564 (+4%)	0.546 (+0%)	0.555 (+2%)	**0.565 (+4%)**	0.559 (+3%)

2.2 Construction of Emotion Vectors

There are multiple views of what emotion is and how it should be represented. Ortony, Clore and Collins regard emotion as consequences of events, actions of agents, and aspects of objects. They introduced the OCC model which specifies 22 emotion types and two cognitive states[1] [6], in contrast to sentiment analysis which categorises text into binary classes (i.e. positive/negative), in turn providing potentially more information for diversification. Here we follow the OCC model because it has been considered as a superior view by the cognitive psychology community. Based on this model, Shaikh et al. [1] developed a state-of-the-art text-based emotion extraction system. In this work, we use our own implementation of Shaikh et al. approach which is shown to be more accurate than other state-of-the-art emotion extraction systems.

Our emotion extraction method is sentence-based and makes a binary decision about the presence of each emotion for a given sentence. Since the emotion extractor is rule-based there is no need for training the model. In order to extract emotions from a retrieved document, we consider the following procedure. Let S denote a set of sentences associated to a document d. For each sentence s in S, we construct a 24 dimension vector where each component can take value 1 if the emotion is present in the sentence and 0 otherwise. Then, in order to represent the emotion contained in d, we give equal importance to each sentence by averaging the emotion vectors of the sentences in d.

[1] The emotion categories are: joy , distress, happy-for, sorry-for, resentment, gloating, hope, fear, satisfaction, fears-confirmed, relief, disappointment, shock, surprise, pride, shame, admiration, reproach, gratification, remorse, gratitude, anger and the two cognitive states are love and hate [6].

3 Experiment and Results

Implementation. Documents were indexed using the Lemur toolkit (http://www.lemurproject.org/). Standard stop-word removal and stemming techniques were applied at indexing time to both documents and query topics. The top n documents (with $n = 20, 50, 100, 200$) were retrieved in answer to each query using a unigram language model with Dirichlet smoothing, where the smoothing parameter was set according to standard values (i.e., $\mu = 2000$). The ranking of the top n retrieved documents formed the baseline (identified as LM in Table 1) against which we compared their re-ranked version according to the approaches presented in Section 2.1, where $sim(d, q)$ was estimated according to the scores returned by LM and $esim(e(d), e(d'))$ was computed by the cosine similarity or Pearson's correlation between the emotion vectors representing the documents. We also tested MMR and AVG-INT considering only text features (i.e., *MMR(t)* and *AVG-INT(t)*): these are based upon the diversification approaches presented in Section 2.1, but use term vector representations of documents instead.

Experiment Settings. We tested our approaches on the TREC 678 Interactive Track collection containing 20 topics which also have been used for diversity task evaluation [7]. Ranking approaches were evaluated according to α-nDCG [3] at different rank positions. Results were similar both when using the cosine similarity and the Pearson's correlation: we only report the former due to space limits. For all the diversification approaches, we varied λ in the range $[0, 1]$ with granularity of 0.05. We report the results obtained selecting parameter values that maximise α-NDCG@10 for each query.

Results. The results[2] reported in Table 1 show that considering emotion features improves retrieval effectiveness. Emotion-based approaches display better performances than LM. We found that emotion-based diversification obtained substantial gains (about 20%) for more than 30% of queries over LM. For example, for topic 446i, "tourists, violence", diversifying rankings based on emotion, provides substantial increments at all levels of diversification (i.e. for all λ values). Emotion-based approaches also provide better performance than the *MMR(t)* approach, which employs text features. Whilst the average effectiveness gains are marginal in this preliminary study, there is a case for using emotion features to diversifying document rankings.

4 Conclusions

In this paper we investigated the effectiveness of using emotion features when diversifying document rankings. We adapted existing models (i.e. MMR and AVG-INT) to exploit emotional features. The results are encouraging and show

[2] Since the topic set is small (i.e., 20 queries), performing significance tests would not be appropriate [8, pages 178–180]. Moreover we do not report results obtained for $n = 200$ for space limits.

improvement when including emotion features for re-ranking retrieved results. This work is a foundation towards future research that employs emotion features to improve IR systems. Future work will consider combining both text and emotion features building more elaborate diversity models.

References

1. Shaikh, M.A.M., Prendinger, H., Ishizuka, M.: A Linguistic Interpretation of the OCC Emotion Model for Affect Sensing from Text. Affective Information Processing (2009)
2. Moshfeghi, Y., Jose, J.M.: Role of emotional features in collaborative recommendation. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 738–742. Springer, Heidelberg (2011)
3. Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR 2008 (2008)
4. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998 (1998)
5. Yin, X., Huang, J.X., Zhou, X., Li, Z.: A survival modeling approach to biomedical search result diversification using wikipedia. In: SIGIR 2010 (2010)
6. Ortony, A., Clore, G., Collins, A.: The cognitive structure of emotions. Cambridge University Press, Cambridge (1990)
7. Zhai, C.X., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR 2003 (2003)
8. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworth, London (1979)

Improved Stable Retrieval in Noisy Collections

Gianni Amati, Alessandro Celi, Cesidio Di Nicola,
Michele Flammini, and Daniela Pavone

Fondazione Ugo Bordoni, Rome, Italy
Department, of Computer Science, University of L'Aquila, L'Aquila, Italy

Abstract. We consider the problem of retrieval on noisy text collections. This is a paramount problem for retrieval with new social media collections, like Twitter, where typical messages are short, whilst dictionary is very large in size, plenty of variations of emoticons, term shortcuts and any other type of users jargon. In particular, we propose a new methodology which combines different effective techniques, some of them proposed in the OCR information retrieval literature, such as n-grams tokenization, approximate string matching algorithms, that need to be plugged in suitable IR models of retrieval and query reformulation. To evaluate the methodology we use the OCR degraded collections of the Confusion TREC. Unlike the solutions proposed by the TREC participants, tuned for specific collections and thus exhibiting a high variable performance among the different degradation levels, our model is highly stable. In fact, with the same tuned parameters, it reaches the best or nearly best performance simultaneously on all the three Confusion collections (Original, Degrade 5% and Degrade 20%), with a 33% improvement on the average MAP measure. Thus, it is a good candidate as a universal high precision strategy to be used when there isn't any a priori knowledge of the specific domain. Moreover, our parameters can be specifically tuned in order to obtain the best up to date retrieval performance at all levels of collection degradation, and even on the clean collection, that is the original collection without the OCR errors.

Keywords: Noisy text retrieval, OCR'd documents, approximate string matching, DFR probabilistic models, cumulative term frequencies, n-grams.

1 Introduction

OCR errors have little effect on retrieval with good quality or authoritative text, while effectiveness can be seriously affected in short texts with frequent errors [10,11]. According to the OCR text retrieval literature, errors may be recovered along with two different directions [6]: by either correcting errors in the collections, or coping with possible errors during the matching phase with the use of approximate string matching techniques.

The n-grams tokenization is usually performed during the indexing phase. This technique makes the system error-tolerant and outperforms other effective retrieval or spell correction techniques for OCR documents [8,2,3]. A n-gram token is any subsequence of n successive characters of the token.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 342–345, 2011.

We found the Divergence from Randomness (DFR) models [1] particularly flexible to handle simultaneously the statistics of whole tokens and their extracted n-grams without recurring to parameters and in a pure additive way. Some of the DFR models are indeed parameter-free and, more importantly, have an independent component dedicated to the term frequency normalization. Moreover, they have experimentally shown better performances with respect to the other fundamental models with similar characteristics.

To evaluate results we use the TREC-5 Confusion suite [5], consisting of three different collections, depending on the their degrading factor [9]: Original, Degrade 5% and Degrade 20%.

To face the stability of retrieval with respect to differ levels of noise in document collections, we propose an integrated framework, that extends a DFR model with syntactical query reformulation, approximate string matching and n-grams tokenization. Results are shown in Table 1.

2 Methodology

We here define the model DFR_{NG} that weights the n-grams in the document. After the indexing of the documents by standard tokens and their n-grams constituents (experimentally best when $n = 4$), we have chosen suitable approximate string matching methods to reformulate the query terms, and we have defined the model DFR_{NG} as matching function.

As approximate string matching methods we use the Edit Distance, the Weighted Edit Distance induced by OCR confusion matrices for the error probabilities, and the Jaccard Coefficient calculated on the n-grams sets $NG_n(t)$ of the tokens t (see [7] for a survery). For each term t a set $S(t)$ of reformulated similar terms having limited Edit and Weighted Edit Distance, or a Jaccard Coefficient above a given threshold is selected (details cannot be displayed here due to space limitation). To limit the insertion of similar terms to syntactical errors only, we filter $S(t)$ excluding the terms belonging to a standard English dictionary such as Wordnet.

The DFR term weighting models $w(tfn(tf), p_t)$ have a term frequency normalization component $tfn(tf)$, which is a function of the number of occurrences tf of the term t in a document, and assume a prior distribution p_t for the terms. We use the DFR model $I(n)L2$ (which has exhibited the best performance in the experimental phase) that has the weighting function $w(tfn(tf), p_t) = -\frac{tfn(tf)}{tfn(tf)+1} \log p_t$ with the term frequency normalization $tfn(tf) = tf \cdot \log(1+\frac{c \cdot \overline{l}}{l})$, where $p_t = \frac{n_t+0.5}{N+1}$, c is a parameter, l the document length, $\overline{l}$ the average document length, N the size of the collection and n_t the document frequency of the term t. We substitute different values for the term frequency in the DFR weighting function $w(tfn(\_), p_t)$, according whether the tokens were terms, similar terms, or their generated n-grams, and sum the resulting weights without parameters to obtain the final weight (surprisingly, weighting differently the various types of terms, in the sum and in the frequencies of tf_J, experimentally has not improved the retrieval performance), obtaining the final similarity formula:

$$DFR_{NG}(D,Q) = \sum_{t \in Q} w\left(tfn\left(tf_J\right), p_t\right) + \sum_{t' \in NG_n(t)} w\left(tfn(tf'), p_{t'}\right)$$

where $tf_J = tf + \sum_{t' \in S(t)} tf' \cdot J(t,t')$, tf' is the raw term frequency of t' in the document D and $J(t,t')$ is the Jaccard function (or any other approximate string matching function). Note that tf_J is a *cumulative* frequency of the actual frequencies of t with its similar terms t'. This simple cumulative technique differs from the conventional query reformulation methods, that weight the reformulated terms independently.

Table 1. Stability of DFR_{NG} with n-grams tokenization and approx. string matching with Weighted Edit Distance on TREC-5 Confusion. Values are given in terms of MAP measure. SW stands for stop words elimination, ST for Porter stemming.

Run	c $I(n)L2$ val.	parsing	Original	Degrade 5%	Degrade 20%	Average
TREC ETHFR94P			0.7353	0.3720	0.4978	0.5350
TREC ETHFR94N			0.7353	0.5737	0.3219	0.5436
I(n)L2	3.6	SW,ST	0.9016	0.7021	0.2872	0.6303
DFR_{NG}	3.6	SW,ST	0.9144	**0.7587**	0.4961	**0.7231**
Best DFR_{NG}	*	*	**0.9249**	**0.7587**	**0.5192**	

* Best configuration for DFR_{NG} at each degradation level: Original $c = 4$; Degrade 5% $c = 3.6$; Degrade 20% $c = 3.6$ without ST, SW.

3 Conclusions

We have presented a new model able to improve the quality of retrieval in noisy data collections, and it was tested on the TREC 5 Confusion track. The model is an extension of a DFR model, that combines query reformulation with approximate string matching and n-grams tokenization.

Results in Table 1 demonstrate improvement over the official results of the TREC 5 Confusion track, both in terms of average and absolute achieved MAP values. As it is possible to check, the best solutions proposed by the TREC participants are tuned for dealing with specific collections (Degrade 5% and Degrade 20%), at the expense of the performance of the original (and the other degraded one). Such a variability induces a low average performance. On the other hand, our model is highly stable for all the three degradation levels. In fact, with the same tuned parameters, it reaches the best or nearly best performance simultaneously on all the three Confusion collections (Original, Degrade 5% and Degrade 20%), with a significant improvement on the average MAP measure. In particular, the increase is 33% with respect to TREC's best run and 14% with respect to standard I(n)L2, whose performance is however excessively poor on Degrade 20%. Thus, our model is a good candidate as a universal high precision strategy to be used when there isn't any a priori knowledge of the specific domain.

In the same table we show that slightly tuning the parameters it is possible to obtain the best up to date retrieval MAP values for each single collection. In particular, while the basic model is already the best for Degrade 5%, the Best

DFR_{NG} Original result is obtained just increasing c to 4, and for Degrade 20% by deactivating stop words elimination and stemming (leaving $c = 3.6$).

Interestingly, Table 1 shows a remarkable improvement not only on degraded collections, but even with respect to standard relevance. This outcome is particular important to investigate, because so far the use of n-grams was shown to be useful only with degradation level greater than 10%. This unexpected improvement can be partially due to the fact that, even when collections are considered "clean" or "almost clean", they actually contain typing or spelling errors. A second argument is that, as it can be checked in Table 1 for Degrade 20%, in case of high noise the n-grams tokenization is able to replace more effectively Porter's algorithm, so that it can be assimilated to a sort a language independent stemming algorithm. Future work will explore whether this evaluation outcome of such a pure syntactical method can be generalized to other TREC collections.

One limit of the n-grams strategies is the big dimension of the generated index. However, the size of the TREC-5 Confusion suite (395 MB) did not cause any problem during the indexing process. Our next step is to optimize indices in order to better compress and access the indices for bigger collections. The extension of the DFR models with n-grams is of independent interest and should be further investigated. Our experimental study has been conducted with the Terrier IR platform [4].

References

1. Amati, G., van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20(4), 357–389 (2002)
2. D'Amore, R.J., Mah, C.P.: One-time complete indexing of text: Theory and practice. In: SIGIR, pp. 155–164 (1985)
3. Harding, S.M., Croft, W.B., Weir, C.: Probabilistic retrieval of ocr degraded text using n-grams. In: Peters, C., Thanos, C. (eds.) ECDL 1997. LNCS, vol. 1324, pp. 345–359. Springer, Heidelberg (1997)
4. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 517–519. Springer, Heidelberg (2005)
5. Kantor, P.B., Voorhees, E.M.: Rep. on Trec-5 confusion track. In: TREC (1996)
6. Mitra, M., Chaudhuri, B.B.: Information retrieval from documents: A survey. Inf. Retr. 2, 141–163 (2000)
7. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. 33(1), 31–88 (2001)
8. Shannon, C.E.: A mathematical theory of communication. Mobile Computing and Communications Review 5(1), 3–55 (2001)
9. Rice, S., Kanai, J., Nartker, T.: An evaluation of information retrieval accuracy. UNLV Information Science Research Institute Annual Report (1993)
10. Taghva, K., Borsack, J., Condit, A.: Results of applying probabilistic ir to ocr text. In: SIGIR, pp. 202–211 (1994)
11. Taghva, K., Borsack, J., Condit, A.: Evaluation of model-based retrieval effectiveness with ocr text. ACM Trans. Inf. Syst. 14(1), 64–93 (1996)

On the use of Complex Numbers in Quantum Models for Information Retrieval*

Guido Zuccon**,***, Benjamin Piwowarski**, and Leif Azzopardi

School of Computing Science,
University of Glasgow,
Scotland, UK
guido@dcs.gla.ac.uk, benjamin@bpiwowar.net, leif@dcs.gla.ac.uk

Abstract. Quantum-inspired models have recently attracted increasing attention in Information Retrieval. An intriguing characteristic of the mathematical framework of quantum theory is the presence of *complex numbers*. However, it is unclear what such numbers could or would actually represent or mean in Information Retrieval. The goal of this paper is to discuss the role of complex numbers within the context of Information Retrieval. First, we introduce how complex numbers are used in quantum probability theory. Then, we examine van Rijsbergen's proposal of evoking complex valued representations of informations objects. We empirically show that such a representation is unlikely to be effective in practice (confuting its usefulness in Information Retrieval). We then explore alternative proposals which may be more successful at realising the power of complex numbers.

1 Introduction

In the recent years, there has been increasing interest around quantum-inspired models for Information Retrieval (IR). An intriguing characteristic of the mathematical framework upon which these models are based is the presence of complex numbers. While traditional models, such as the vector space models, are based on the field of real numbers, quantum models use complex vector spaces (i.e., Hilbert spaces). Complex numbers are one of the key concepts of the mathematical framework of quantum theory. They allow to describe and model phenomena such as interference, outlined in the next section.

How to harness the use of complex numbers in quantum-inspired IR models has been largely ignored, and this is also the case for most quantum-inspired models proposed in disciplines outside Physics, i.e., the so called "Quantum Interaction" research area [2]. There are three main exceptions. In [6], van Rijsbergen only sketched out the use of complex numbers, proposing to store the term

* Supported by (**) EPSRC Grant number EP/F014384/ and (***) Zirak s.r.l. (http://www.zirak.it/). The authors are thankful to Peter Bruza, Kirsty Kitto, Massimo Melucci and Keith van Rijsbergen for initial discussion on the use of complex numbers, and to the reviewers for their comments.

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 346–350, 2011.

frequency and the inverse document frequency respectively in the magnitude r and the phase φ of a complex number $re^{i\varphi}$. However, no further theoretical insight supporting this proposal has been given, and no empirical evaluation has been performed. In the context of semantic space models, De Vine and Bruza [3] proposed a novel approach for the construction of spaces based on circular holographic representations, where the construction of complex valued vectors plays a fundamental role in preserving the order information in n-grams. However, they do not provide an interpretation of how complex numbers are used. The same observation applies to the quantum probability ranking principle [8] (qPRP), which relies on the notion of interference. Moreover, in qPRP, as the vector space is not explicitely defined, complex numbers are only implicitely used.

In this paper, we first define what complex numbers are useful for in the context of the mathematical framework of quantum theory, i.e., of so-called "quantum probabilities". We then demonstrate theoretically and empirically that van Rijsbergen's proposal does not hold, and discuss how complex numbers could be made explicit for the qPRP based model [8] and conclude.

2 Use of Complex Numbers in Quantum Theory

As stated, complex numbers are pervasive throughout the mathematical framework of quantum theory, due to the wave nature of matter. As such, they provide more freedom in terms of (quantum) probability distributions, and it is this degree of freedom that we describe in this section. Given the space constraints, we make bold simplifications for the sake of clarity.

First, we need to define what a *quantum probability* is. In its simplest form, a quantum probability is characterised by a quantum probability distribution and an event, which are respectively defined by the *unit* vectors $\mathbf{d}$ and $\mathbf{e}$. The probability $q(\mathbf{e}|\mathbf{d})$ of event $\mathbf{e}$ given distribution $\mathbf{d}$ is then $|d{\cdot}e|^2$, which corresponds to the squared cosine between the two vectors. This relationship shows that vector based IR can be interpreted within quantum probability theory [6].

Let us analyse further the concept of quantum probability, by considering two vectors on a two dimensional space. Specifically, we represent the event as $\mathbf{e} = \sqrt{1/2}(1,1)^\top$ and the distribution as $\mathbf{d} = \sqrt{1/|1+e^{i\varphi}|}(1,e^{i\varphi})^\top$, where $\mathbf{d}$ depends on a parameter, i.e. the angle or phase $\varphi \in [0, 2\pi[$, $|\cdot|$ denotes the usual norm of a complex number, and $\sqrt{1/2}$ and $\sqrt{1/|1+e^{i\varphi}|}$ are the normalising factors that yield unit vectors. Unless $\varphi \in \{0, \pi\}$, $\mathbf{d}$ is expressed by complex numbers with no null imaginary parts. By varying φ between 0 and π, the probability $q(\mathbf{e}|\mathbf{d})$ varies between 1 and 0. Further, an important fact is that multiplying $\mathbf{e}$ and $\mathbf{d}$ by $e^{i\psi}$ would not change the (quantum) probability value, for all $\psi \in \mathbb{R}$. It is the *phase difference* between the components in the vector that is important. In our example, the phase difference between the two components of the vector in $\mathbf{d}$ is φ.

What does this mean in practice?. A simple IR example can clarify the situation. If we assume that $\mathbf{e}_a = (1,0)^\top$ and $\mathbf{e}_b = (0,1)^\top$ are documents containing word a and b, respectively, then $\mathbf{e} = \sqrt{1/2}(1,1)^\top$ means that the document

Table 1. Values of MAP for two matching models based respectively on a real-valued and a complex-valued vector space model ($\mathbb{R}$-VSM and $\mathbb{C}$-VSM). Statistical significance using a two-tailed paired t-test with $p \ll 0.01$ is indicated by †.

	AP8889	**WSJ8792**	**LA8990**	**WT2g**	**WT10g**
$\mathbb{R}$-VSM	.1870	.1789	.1378	.1276	.1038
$\mathbb{C}$-VSM	.1313†	.0967†	.1146†	.0781†	.0232†

contains both words in equal quantities. By varying φ in $\mathbf{d}$, we can express that a document is relevant if it contains either a or b, but not both (case $\varphi = \pi$), or is relevant if it contains a, b or both. (case $\varphi = 0$). Intermediate values of φ enable smooth transitions from one possibility to the other.

The idea of using the phase difference between words could also be used in the Quantum Information Retrieval framework [5] where, based on quantum probability theory, the term vector space is used to represent both documents and information needs. In this framework, words can *interfere* between each other in the measurement of relevance.

Interestingly, one could interpret the negative numbers (i.e., $\varphi = \pi$) obtained when performing Latent Semantic Analysis [4] through the prism of the quantum formalism: in this case, a basis vector would contain two categories of words that are mutually exclusive, i.e., that generally do not co-occur.

3 Analysis of the Potentials of Complex Numbers for IR

Encoding idf in the Phase. In [6, page 25], van Rijsbergen suggested to use complex numbers as a sort of information storage mechanism, which then has to be transformed at matching time, where instead of associating to each component of the vector space a tf $\times$ idf value, it associates tf $\times\, e^{i \cdot \text{idf}}$. As this is the only example of complex number usage in van Rijsbergen's book, let us go beyond its usage as a simple storage scheme, which is not particularly useful in itself, and interpret it directly as a new complex weighting scheme for documents and queries. Note that we normalised the idf so it ranges between 0 and 2π, since these are the extremal values that a phase can take.

From a theoretical point of view, according to section 2, van Rijsbergen's proposal would mean that if the query contains a word a with a high idf and b with an average idf, then a document would have a high probability of being relevant if it contains either a or b, but not both! This counterintuitive behaviour does not really depend on the mapping between idf and the $[0, 2\pi]$ range.

For completeness, we experimented with the standard vector space model ($\mathbb{R}$-VSM) and the "complex" VSM ($\mathbb{C}$-VSM) on a number of TREC collections. Both documents and queries were indexed with the Lemur toolkit (`http://www.lemurproject.org/`), after applying Porter stemming and stop-word removal. Results are reported in Table 1, and show clearly that the

encoding of idf in the phase does not perform well, even when compared to the low baseline of the tf $\times$ idf weighting scheme.

Complex Numbers in qPRP. The quantum probability ranking principle (qPRP) is a ranking approach alternative to the traditional PRP that implicitly relies on interferences, and hence on complex numbers [8]. The qPRP has been shown to perform better than other alternatives for the diversity task in IR, and hence it is interesting to make explicit the representation of documents and to uncover the meaning of complex numbers in that case.

Intuitively, a phase difference corresponds to the fact that documents are relevant for the same topic, and their relevance probability should not add up. A possible re-interpretation of the example of Section 2 is as follows. Assume that a (resp. b) corresponds to the fact that document a (resp. b) is relevant. We can see that with a phase difference of $\pi/2$, a ranking containing the documents a and b would have the same probability of being relevant to the user than a ranking containing only a or b.

How to explicitly encode the relevance of documents and to define the probability distribution is still not clear at this stage. However, the previous example shows that it might be possible to build up the document representation by ensuring that documents do exhibit the same interference as the one that was empirically shown to work well (e.g., defined as a function of the cosine between two documents in the standard document vector space [7]).

4 Conclusions

In this paper we argued that since complex numbers play a central role in quantum theory, it is of interest to harness its extended representational power in quantum-inspired IR models. We have outlined the role of complex numbers in quantum probability theory. We have shown that the proposal of [6] does not hold theoretically or empirically. We have however observed that the qPRP, which was shown empirically to perform well, implicitly relies on complex numbers. In this respect, we have identified a promising direction to further explore the application of complex numbers within IR.

References

1. Accardi, L., Fedullo, A.: On the statistical meaning of complex numbers in quantum mechanics. Lettere Al Nuovo Cimento (1971 – 1985) 34, 161–172 (1982)
2. Bruza, P., Sofge, D., Lawless, W.F., van Rijsbergen, C.J., Klusch, M. (eds.): QI 2009. LNCS, vol. 5494. Springer, Heidelberg (2009)
3. De Vine, L., Bruza, P.: Semantic oscillations: Encoding context and structure in complex valued holographic vectors. In: QI 2010 (2010)
4. Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W. (eds.): Latent Semantic Analysis. Lawrence Erlbaum Associates, Mahwah (2007)
5. Piwowarski, B., Frommholz, I., Lalmas, M., van Rijsbergen, C.J.: What can Quantum Theory bring to IR? In: CIKM 2010, pp. 59–68 (2010)

6. van Rijsbergen, C.J.: The Geometry of Information Retrieval. Cambridge University Press, Cambridge (2004)
7. Zuccon, G., Azzopardi, L., Hauff, C., van Rijsbergen, C.J.: Estimating interference in the qprp for subtopic retrieval. In: SIGIR 2010, pp. 741–742 (2010)
8. Zuccon, G., Azzopardi, L., van Rijsbergen, C.J.: The quantum probability ranking principle for information retrieval. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 232–240. Springer, Heidelberg (2009)

A Query Performance Analysis for Result Diversification

Jiyin He[1], Marc Bron[2], and Maarten de Rijke[2]

[1] CWI, Science Park 123, 1098 XG Amsterdam
[2] ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam
j.he@cwi.nl, {m.m.bron,derijke}@uva.nl

Abstract. Which queries stand to gain or loose from diversifying their results? Some queries are more difficult than others for diversification. Across a number of conceptually different diversification methods, performance on such queries tends to deteriorate after applying these diversification methods, even though their initial performance in terms of relevance or diversity tends to be good.

1 Introduction

Result diversification is a retrieval strategy for dealing with ambiguous or multi-faceted queries; the system makes an educated guess as to the possible facets of the query and presents documents pertaining to different facets to the user [1, 3, 5, 7]. However, diversification is not a universal solution from which all queries stand to gain. Some queries benefit, while others get hurt, e.g., non-relevant documents may be promoted to the top of a ranked list because of their "diversity." [7] address this issue by balancing relevance and diversity with a trade-off parameter on a per-query basis, which leads to improved diversification effectiveness. However, what properties of a query make it suitable for diversification? More generally, how can diversification methods without a "trade-off parameter" benefit from these insights?

We investigate query diversification performance in a more general setting, aiming to provide a better understanding of when a query is (un)suitable for diversification. Let's call a query "difficult" when diversification is ineffective or deteriorates performance (in terms of relevance or diversity) across multiple types of diversification method. We use result diversification methods that are conceptually different and seek answers to the following research questions: *RQ1. Are some queries more difficult than others for diversification across different diversification methods?* and *RQ2. What properties of a query make it difficult?* There are many avenues to explore here, e.g., the ambiguity of a query, the facets associated with a query covered by the collection, etc.; we focus on the relation between diversification effectiveness and the initial performance of queries in terms of relevance and diversity.

2 Method

Diversification methods. We employ three diversification methods: MMR [3], IA-select [1] and Round Robin (RR) [5] that diversify a ranked list via re-ranking. By doing so we expect to identify query properties that hold across diversification methods

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 351–355, 2011.

with different underlying assumptions. MMR determines the value of a document for diversification through a linear combination of its similarity to the query (relevance) and the smallest similarity to the documents already returned (diversity), where the trade-off between relevance and diversity is controlled by a parameter λ: $score_{d,q} = \lambda Rel_{d,q} + (1-\lambda) Div_{d,q}$. Unlike MMR, IA-select explicitly models the facets associated with a query. Documents are selected based on their initial retrieval scores, weighted by the probability that the selected document covers the underlying facets given that previously selected documents failed to do so. In RR, facets are modeled via clustering and ranked according to their estimated relevance to the query. Documents in each cluster keep the order of their original retrieval scores; then, documents in different clusters are selected in a round robin fashion. While IA-select aims to cover the *most important* facet of a query in the top ranked documents, RR seeks to cover *different* facets.

Analysis. For RQ1, we analyse the correlation among different diversification methods. Let m be a diversification method, $Q = q_1, \ldots, q_n$ a list of queries and $S_m = s_{q_1}, \ldots, s_{q_n}$ the per-query evaluation scores of the diversification results for Q, in terms of an evaluation metric. We calculate Pearson's linear (ρ) and Kendall's rank correlation (τ) between the performance of two methods S_{m_1} and S_{m_2}. A high correlation implies that queries with a relatively high (low) score using m_1 also receive a relatively high (low) score using m_2.

For RQ2, we identify two groups of queries. Let $t_q(m, b)$ be the performance difference between an initial baseline result b and a diversification result using method m for a query q as evaluated by a diversification measure. The first group consists of "easy" queries that are improved by at least one method and not hurt by others: $E = \{q | \sum_m t_q(m, b) > 0 \textit{ and } \forall m, t_q(m, b) \geq 0\}$. The second group consists of "difficult" queries that are hurt by at least one method and not improved by others: $D = \{q | \sum_m t_q(m, b) < 0 \textit{ and } \forall m, t_q(m, b) \leq 0\}$, where all diversification methods use the same baseline b. We investigate whether the two groups show different patterns characterized by properties associated with the initial performance of the queries.

Let G_q^K be the top K documents retrieved in response to query q and evaluated by a diversity measure, F_q be the set of facets of q and R_q^N be the N judged relevant documents of q in collection C. The properties we examine are as follows. (i) The performance of the initial ranked list G_q^K in terms of a diversity measure $eval@K$. (ii) The number of relevant documents and facets covered in the top of a ranked list: $R@K = |G_q^K \cap R_q|$ and $F@K = |\{f | f \in G_q^K\} \cap F_q|$. Here, we decompose $eval@K$ into two factors: relevance and diversity, in order to see whether these two factors have a different impact on the diversification performance. (iii) The percentage of relevant documents (facets) covered in the top of a ranked list compared to the total number of relevant documents (facets) for a query in the collection: $R@K\% = R@K/|R|$ and $F@K\% = F@K/|F|$. This takes into account the collection factor, i.e., diversification will not work in a collection without diverse content for a query.

3 Experiments and Results

We conduct our experiments using the ClueWeb category B dataset and the 100 test queries from TREC'09 and '10 Web track diversity task. For evaluation, we take the α-NDCG (@5, 10 and 20) [4], used as official measure at the TREC'09 and '10 diversity

track, with α set to 0.5. We use the Markov Random Field model (MRF) [6] with default parameter settings to generate the initial baseline results b. We diversify with the top 100 documents in b using the three diversification methods described in Section 2.[1] We only include the results of a method with its optimal parameter settings found in a preliminary experiment. For MMR, λ is found to be 0.9. Following [5], we use LDA [2] to model the underlying facets of a query and of a document for both IA-select and RR, where the optimal number of facets are 50 and 10 respectively.

Table 1 shows the correlation between the performance of different diversification methods. All methods show significant positive correlation in terms of both ρ and τ. In particular, MMR and IA-select show a remarkably strong correlation, while both methods show a weaker correlation with RR, suggesting that RR behaves somewhat differently. The overall significant correlation indicates agreement between methods on the relative performance of queries, i.e., some queries consistently perform worse when subjected to diversification, or are more difficult to achieve good diversification results on, than others, regardless of the method applied.

Table 1. Performance correlation between diversification methods. All correlations are significant (p-value < 0.01).

Eval. measure	α-NDCG@5		α-NDCG@10		α-NDCG@20	
Corr. coef.	ρ	τ	ρ	τ	ρ	τ
MMR vs. IA-sel	0.896	0.830	0.917	0.828	0.925	0.818
MMR vs. RR	0.471	0.336	0.650	0.502	0.689	0.502
IA-sel vs. RR	0.495	0.376	0.669	0.533	0.675	0.515

We identify D and E from the 100 queries based on α-NDCG@5, 10 and 20 and list in Table 2 statistics of the query properties for these groups as discussed above.[2] (i) In terms of α-NDCG, queries in D have significantly higher scores than those in E, suggesting that queries with relatively good initial performance (set D), tend to be "difficult" for diversification. (ii) Queries in D cover significantly more relevant documents (facets), i.e., $R@K$ ($F@K$), compared to those in E except in the case where $K = 20$ for $F@K$. We see that both relevance and diversity of b has an impact on diversification performance. (iii) In terms of $R@K\%$ ($F@K\%$), queries in D have significantly higher scores than those in E, i.e., a larger percentage of all relevant documents (facets) in the collection is covered in the top of b for queries in D than in E.

Table 2. Contrasting properties of initial ranked lists (D vs. E). $^{\blacktriangle}$ ($^{\triangle}$) indicates a significant difference; p-value $<.01$ (.05) using Wilcoxon rank sum test.

	$K = 5$		$K = 10$		$K = 20$	
Query set	E	D	E	D	E	D
# queries	41	18	31	17	27	16
α-NCDG@K	0.08	$0.28^{\blacktriangle}$	0.15	$0.28^{\triangle}$	0.19	$0.34^{\blacktriangle}$
$R@K$	0.65	$2.00^{\blacktriangle}$	2.10	$3.71^{\blacktriangle}$	5.33	$7.69^{\triangle}$
$R@K\%$	0.03	$0.18^{\blacktriangle}$	0.09	$0.24^{\blacktriangle}$	0.24	$0.39^{\triangle}$
$F@K$	0.46	$1.17^{\blacktriangle}$	0.84	$1.29^{\triangle}$	1.18	1.63
$F@K\%$	0.16	$0.51^{\blacktriangle}$	0.46	$0.72^{\triangle}$	0.33	$0.59^{\blacktriangle}$

The phenomena listed under (ii) and (iii) can be explained as follows. Given that all diversification methods do not generate perfect results, during re-ranking, diversification can hurt a result list by replacing a top ranked relevant and "novel" document

[1] We did not remove spam. The performance of the three methods are between the median and the best of systems taking part in the diversity task at the TREC 2009 Web track.

[2] Since we only re-rank the top 100 documents, $|F_q|$ and $|R_q|$ are the relevant documents (facets) of a query covered by the top 100 documents in the initial ranked list.

by a non-relevant document or a relevant but "non-novel" document, where a "novel" document covers the facet of a query that is not (adequately) covered by the documents ranked before it. Intuitively, such replacement would have a higher chance to occur if an initial result list whose top K documents cover a large number of relevant documents or diverse facets, especially if most of the documents ranked below top K are non-relevant or non-novel, e.g., as indicated by a high $R@K\%(F@K\%)$. Also, a high $R@K\%(F@K\%)$ implies that there is little room for improvement. E.g., in the case of $K = 10$, on average 72% of the facets are covered by the initial top 10 documents for queries in D, the potential improvement through diversification lies in finding the other 28% of the facets, while the potential for E is 54%, as only 46% of the facets are covered by the initial top 10 documents.

4 Discussion and Conclusion

We investigated the performance of queries in result diversification with three conceptually different diversification methods. Across methods, some queries are more difficult than others for diversification. Further, queries with relatively good initial performance in terms of relevance or diversity tend to deteriorate through diversification.

The contribution of our analysis is two-fold. (i) We provide empirical evidence which confirms that some queries stand to gain more from diversification than others, independent of the diversification method used. (ii) Our analysis provides insights in the properties that should be focused on when identifying such queries.

We plan to look into predictors for the properties analyzed in this study, i.e, properties confirmed to have a high correlation with diversification performance.

Acknowledgements. This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement no. 250430, the Fish4Knowledge project and the PROMISE Network of Excellence, funded and cofunded by the 7th Framework Programme of the European Commission, grant agreement no. 257024 and no. 258191, the DuOMAn project carried out within the STEVIN programme funded by the Dutch and Flemish Governments under project nr STE-09-12, the Netherlands Organisation for Scientific Research under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, the Center for Creation, Content and Technology, the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, and under COMMIT project Infiniti.

References

[1] Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying Search Results. In: WSDM'09 (2009)
[2] Blei, D., Ng, A., Jordan, M., Lafferty, J.: Latent Dirichlet Allocation. In: JMLR (2003)
[3] Carbonell, J., Goldstein, J.: The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In: SIGIR'98 (1998)

[4] Clarke, C., Kolla, M., Cormack, G., Vechtomova, O., Ashkan, A., Büttcher, S., Mackinnon, I.: Novelty and Diversity in Information Retrieval Evaluation. In: SIGIR'08 (2008)
[5] He, J., Meij, E., de Rijke, M.: Result Diversification Based on Query-specific Cluster Ranking. JASIST 62(3) (2011)
[6] Metzler, D., Croft, W.B.: A Markov Random Field Model for Term Dependencies. In: SIGIR'05 (2005)
[7] Santos, R., Macdonald, C., Ounis, I.: Selectively Diversifying Web Search Results. In: CIKM'10 (2010)

Rare Disease Diagnosis as an Information Retrieval Task

Radu Dragusin[1], Paula Petcu[1], Christina Lioma[2], Birger Larsen[3], Henrik Jørgensen[4], and Ole Winther[5]

[1] Computer Science, University of Copenhagen, Copenhagen, Denmark
[2] Informatics, Stuttgart University, Stuttgart, Germany
[3] Royal School of Library and Information Science, Copenhagen, Denmark
[4] Department of Clinical Biochemistry, Bispebjerg Hospital, Copenhagen, Denmark
[5] Informatics, Technical University of Denmark, Lyngby, Denmark
{dragusin,petcu}@diku.dk, liomaca@ims.uni-stuttgart.de, blar@iva.dk, hlj@dadlnet.dk, owi@imm.dtu.dk

Abstract. Increasingly more clinicians use web Information Retrieval (IR) systems to assist them in diagnosing difficult medical cases, for instance rare diseases that they may not be familiar with. However, web IR systems are not necessarily optimised for this task. For instance, clinicians' queries tend to be long lists of symptoms, often containing phrases, whereas web IR systems typically expect very short keyword-based queries. Motivated by such differences, this work uses a preliminary study of 30 clinical cases to reflect on rare disease retrieval as an IR task. Initial experiments using both Google web search and offline retrieval from a rare disease collection indicate that the retrieval of rare diseases is an open problem with room for improvement.

Keywords: rare diseases, clinical information retrieval, web diagnosis.

1 Introduction

Recently web Information Retrieval (IR) systems have gained popularity among clinicians to assist them in difficult medical cases, for instance rare diseases that they may not be familiar with [1]. However, such systems are not necessarily designed or optimised for diagnosing rare diseases. For example, clinicians' queries tend to be long lists of symptoms, whereas web IR systems typically expect very short queries. Similarly, the hyperlink popularity and recommendation principles typically applied in web IR tend to favour popular webpages; however, information on rare diseases is generally very sparse and less hyperlinked than other medical content. Motivated by such differences, this work considers rare disease diagnosis as an IR task, and asks what design considerations are needed to build an IR system that clinicians can use to diagnose rare diseases?

To address this question, a small preliminary study with 30 real clinical cases is conducted, involving both Google web search and offline retrieval from a specialised rare disease collection (Section 2). The resulting findings offer useful

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 356–359, 2011.

insights on the special characteristics, possibilities and challenges of rare disease diagnosis as an IR task (Section 3). Section 4 concludes this work.

2 Retrieving Rare Diseases: Preliminary Study

The queries used in this work were created from 30 clinical cases of rare diseases, where the query text was extracted directly from the patient symptoms listed in the clinical cases. This was done by one medical doctor and two non-experts. The correct disease diagnosed for these symptoms was not included in the query text. This is an important difference from standard web search queries, where the topic sought is usually explicitly mentioned in the query. The average query length was 22.17 terms. E.g., query for the rare Kleine-Levine syndrome: `Jewish boy age 16, monthly seizures, sleep deficiency, aggressive and irritable when` woken, highly increased sexual appetite and hunger.

The 30 queries were used to retrieve documents using Google web search, and separately using the Indri IR system on a small rare disease collection specifically created for this task. This dataset contains 31,746 documents, crawled from web sites specialising on rare and genetic diseases[1]. Specifically, we collected 10,280 documents on rare diseases and 21,466 documents on genetic diseases (many of which are rare), to be referred to as RARE and GENET henceforth.

Three runs were realised with Google: (1) using standard Google web search; (2) customing Google[2] on the RARE dataset but retrieving documents from the whole web; (3) restricting Google to retrieve from the RARE & GENET websites, plus 5 websites containing only url links to rare disease information (these 5 websites were excluded from our collection because they included url links only). Three more runs were realised with Indri: (4) retrieval from RARE only; (5) retrieval from RARE & GENET; (6) retrieval from RARE & GENET, with a rank boost of RARE documents by a factor of 4.

Runs with Indri used the query likelihood language model with Dirichlet smoothing at default settings ($\mu = 2500$ [2], Krovetz stemming). For run 6, boosting RARE documents was implemented as the prior probability of a document being relevant ($P(D)$). Unless specified otherwise, the baseline query likelihood model assumes that all documents are a priori equally likely to be relevant, and ignores $P(D)$. Motivated by the intuition that RARE documents should have a higher likelihood to include relevant documents when searching for rare diseases, we computed $P(D)$ directly from the collection statistics as follows. Let C denote the complete retrieval collection containing both RARE and GENET. Then, $P(R|C)x + P(G|C)y = 1$, where $x = \phi y$, and where $P(R|C)$ (resp. $P(G|C)$) denotes the probability of all RARE (resp. GENET) documents in the whole collection. ϕ is the boosting factor, set to $\phi = 4$ in this work; this value of ϕ is ad-hoc and untuned, used only for illustration purposes.

[1] The list of urls is available here: http://code.google.com/p/raredisss/wiki/RareGenetResources.

[2] http://www.google.com/cse/

The relevance of the retrieved documents in these 6 runs was assessed by the two non-experts in the top 20 ranks using graded relevance on 3 points (relevant, marginally relevant, non-relevant): (i) relevant documents should address mainly the correct disease in the title or within the first 400 words, and name it using any of its synonyms listed in Orphanet[3]; (ii) in cases of inherited diseases, e.g `autosomal neonatal form of Adrenoleukodystrophy`, documents about the main disease, e.g. `X-linked Adrenoleukodystrophy`, are relevant; (iii) documents about different types of the correct disease, e.g. `Loeys-Dietz syndrome type 1A` instead of `Loeys-Dietz syndrome type II`, are relevant; (iv) documents about other diseases and mentioning the correct disease as an alternative diagnostic or pointing to it are marginally relevant; (v) documents listing many diseases are not relevant if the correct disease is listed after the first 10.

Table 1. Retrieval from the web and our rare disease & genetic disease datasets

Collection	Retrieval approach	P@10	P@20	MRR	NDCG@10	NDCG@20
WEB	Standard Google	.023	.013	.056	.168	.189
WEB	Google Custom on RARE	.030	.017	.173	.275	.283
RARE&GENET	Google Restricted	.003	.002	.033	.033	.033
RARE	LM-Dir	.123	.073	.445	**.516**	**.536**
RARE&GENET	LM-Dir	.157	.105	.467	.423	.493
RARE&GENET	LM-Dir prior on RARE	**.173**	**.115**	**.469**	.433	.492

Table 1 shows the retrieval precision at rank k (P@k), the mean reciprocal rank (MRR) and the normalised discounted cumulative gain at rank k (NDCG@k) of our 6 runs averaged for all 30 queries. NDCG uses graded relevance assessments[4]; all other measures use binary relevance assessments which consider marginally relevant documents as non-relevant. Retrieval from the web refers to the part of the web indexed by Google. Two findings emerge: (i) Google overall underperforms for this task, especially when restricted to the sites of our collection; (ii) the MRR scores show that on average the correct diagnosis appears at ranks 2-3 with Indri (.445 - .469) and at best at rank 5-6 with Google (.173). Even though the Google retrieval algorithm is not known, a possible reason for this performance may be the fact that it is not optimised for this task. E.g., if Google uses popularity-based metrics like Page-http://code.google.com/p/raredisss/wiki/RareGenetResourcesRank, the desired relevant documents are not likely to be helped by this, because they are not necessarily as heavily hyperlinked as other medical documents; if Google considers logged user & query features like clickthrough data, rare disease queries are not likely to benefit from this, because they are probably not sufficiently frequent among users; the fact that Google does not accept queries longer than 32 terms indicates that it is optimised for queries shorter than our 22.17 word-long queries.

[3] http://www.orpha.net/

[4] with the following gain values: relevant = 3, marginally relevant = 1.

3 The Characteristics of Rare Disease Retrieval

The above observations indicate that rare diseases retrieval may be seen as a distinct IR task with the following user-based and system-based characteristics.

On the user side, the clinicians' information needs are ideally fullfilled by a single document about the correct rare disease, similarly to early-precision tasks such as named-page finding. However, the clinicians' queries are expressed in very different ways than named-page or other web search queries: (a) they are very long; (b) they consist of lists of patient symptoms, where term independence assumptions could lead to topic drift (e.g. `sleep deficiency, increased sexual appetite` is topically different to `sexual deficiency, increased sleep`); (c) some symptoms listed in the query may not apply to the correct disease, and conversely, some pertinent symptoms for the correct disease may be missing from the query because they are masked under different conditions. In short, the clinicians' queries on rare diseases are likely to be more feature-rich but also more noisy than in web IR, and should be treated as such.

On the system side, popularity-based metrics derived from hyperlinking, user visit rates, or other forms of recommendation may not benefit the retrieval of rare diseases. Instead, features that may aid this task could be domain-specific enhancements (such as the prior on the RARE dataset), or information about the rarity, geographic distribution and statistics of a disease. Finally, often efficiency concerns lead to brute-force index pruning for web search, e.g. by removing from the index terms of low frequency or that are unusually long. Such practices may be particularly damaging for rare disease retrieval, as the medical terminology involved may be exceptionally rare or formed by heavy term compounding.

4 Conclusion

This work reflected on rare disease diagnosis as an IR task, where clinicians use symptoms as queries in order to retrieve a correct diagnosis. A small preliminary study involving real clinical cases of rare diseases was conducted in collaboration with a medical doctor. Findings revealed that rare disease retrieval has several distinct features that differentiate it from standard web IR, and that applying standard web IR for this task may not be optimal. Future work includes developing IR approaches for the domain of rare diseases.

References

1. Bouwman, M.G., Teunissen, Q.G.A., Wijburg, F.A., Linthorst, G.E.: Doctor Google ending the diagnostic odyssey in lysosomal storage disorders: parents using internet search engines as an efficient diagnostic strategy in rare diseases. Arch. Dis. Child. 95(8), 642–644 (2010)
2. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004)

Distilling Relevant Documents by Means of Dynamic Quantum Clustering

Emanuele Di Buccio and Giorgio Maria Di Nunzio

Department of Information Engineering – University of Padua
Via Gradenigo, 6/a – 35131 Padua – Italy
{emanuele.dibuccio,giorgiomaria.dinunzio}@unipd.it

Abstract. Dynamic Quantum Clustering (DQC) is a recent clustering technique based on physical intuition from quantum mechanics. Clusters are identified as the minima of the potential function of the Schrödinger equation. In this poster, we apply this technique to explore the possibility to select highly relevant documents relative to a query of a user. In particular, we analyze the clusters produced by DQC with a standard test collection.

1 Introduction

Clustering is the problem of automatically organizing large unlabeled data collection into a finite set of clusters on the basis of a similarity measure among them. The procedure of cluster analysis can be summarized with four basic steps [4]: feature selection or extraction; clustering algorithm design or selection; cluster validation; results interpretation.

In this poster we study a possible application of a recently proposed clustering method, named Dynamic Quantum Clustering (DQC), to the field of Information Retrieval (IR). This method works by analog by considering data samples as particles that obey quantum physics laws. Clusters are computed by means of the time dependent Schrödinger equation. We investigate the feasibility of the application of this method to the problem of textual document clustering. In particular, we want to investigate the following problems: how different document feature selections affect the performance; how the analysis of the principal components of the matrices involved in the calculations affects the quality of the clusters. Standard IR collections are used for the experiments.

2 Dynamic Quantum Clustering

In Dynamic Quantum Clustering (DQC) [3], the problem of finding clusters is mapped into a problem of quantum mechanics, then the mathematical tools of quantum mechanics are used to reveal the clusters. The intuition behind this quantum clustering approach relies on the analogy between data points and small particles: each data point is a particle characterized by a radial influence region which is specified by a kernel function. The Gaussian function is usually

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 360–363, 2011.

the kernel function. The influence of N data points on a certain point in space is given by the effect of all the particles, that is the sum of all the kernel functions:

$$\psi(\mathbf{x}) = \sum_{j=1}^{N} e^{-\frac{\mathbf{x}-\mathbf{x}_j}{2\sigma^2}} \ . \tag{1}$$

$\psi(\mathbf{x})$ have relative maxima in the regions where the concentration of points is higher. However, finding the relative maxima of an n-dimensional function is computationally time consuming and it is highly sensitive to slight variations of the value of σ. For this reason, DQC uses Eq. 1 indirectly to construct a potential function whose minima are related to the maxima relative to the clusters. In particular, the aim is to search for the Schrödinger potential $V(\mathbf{x})$ for which $\psi(\mathbf{x})$ is a solution:

$$H\psi(\mathbf{x}) \equiv \left(-\frac{\sigma^2}{2}\nabla^2 + V(\mathbf{x})\right)\psi(\mathbf{x}) = E\psi(\mathbf{x}) \ , \tag{2}$$

where H is the Hamiltonian operator, ∇^2 the Laplace operator, and E the system energy. DQC identifies local minima by letting the particles of the quantum system to "be attracted" by the local minima of the potential function. This is performed by defining the evolution of the system to be $\psi(\mathbf{x}, t) = e^{-iHt}\psi(\mathbf{x})$, being i the imaginary unit. From a computational point of view, the advantage of using DQC relies on the fact that the algorithm translates the problem of solving the Schrödinger equation into a matrix form which captures most of the details of the analytic problem. This matrix has at most dimension $N \times N$. In [3], the authors present a detailed analysis of the issues that may arise when large datasets are analyzed by means of DQC. The computational complexity of the problem is controlled by the number of data-points, since this defines the size of the matrix to be exponentiated. The computational cost associated with keeping more features is only related to computing the matrices associated with multiplying a wave-function by a given coordinate which is a one time cost. The computational cost of computing the values of these operators only grows linearly with the number of features.

3 Experiments and Results

Test Collection and Experimental Methodology. Experiments were carried out by using the TREC 2001 Web Track test collection, specifically focusing on the fifty ad-hoc topics. We adapted the code of the COMPACT software in Matlab [2] for these experiments. The experimental methodology consists in the following steps performed for each topic in the test collection:

1. consider the set J_q of all the documents that have been manually judged for the considered topic q;
2. select $k = 1000$ terms to represent the documents; the selected terms are the h terms constituting the topic title and $k - h$ terms extracted from the documents in J_q — stop words are removed;

Table 1. Table 1a reports the mean number of true positive, false positive, and mean values of recall and precision for each of the distinct feature adopted in the term selection strategy, where the mean is computed over all the fifty topics and for the first two and four principal components after a SVD decomposition (k'={2, 4}). Table 1b reports the number of topics n_T for which DQC was able to achieve the 100% of precision, when using a specific feature f for term selection. Table 1b reports also the mean and the median value of k' at which DQC was able to achieve the 100% of precision.

f	True Pos	False Pos	Recall	Precision
totTF	31.07	561.80	0.505	0.125
DF	37.11	662.90	0.544	0.139
IDF	27.51	466.20	0.431	0.138
RSJ	27.86	574.07	0.510	0.101

(a)

f	n_T	Mean k'	Median k'
totTF	44	30.68	20.00
DF	49	17.67	20.00
IDF	20	24.20	9.00
RSJ	44	29.68	20.00

(b)

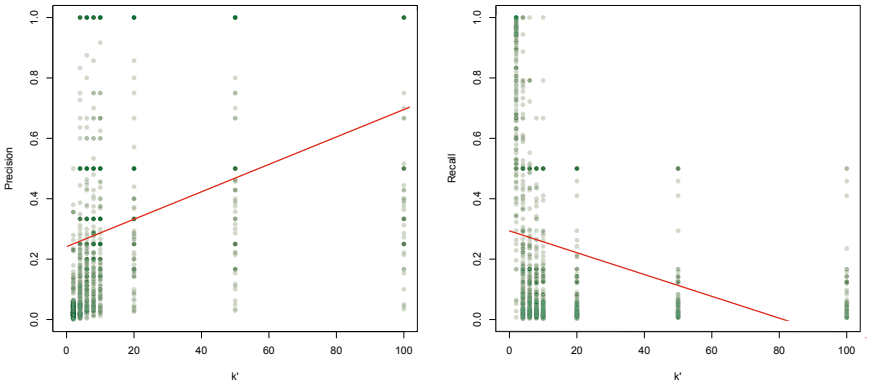
3. prepare a term-by-document matrix $A \in \mathbb{R}^{k \times |J_q|}$ where the element $A_{j,i}$ is the TF·IDF weight of the term j in the document i;
4. apply Singular Value Decomposition to the prepared matrix A, thus decomposing A as $A = U\Sigma V^T$ and consider the first k' columns of V;
5. apply DQC to the matrix V^T.

At step 2 we exploited diverse term selection strategies, specifically ranking terms according to the following features: Document Frequency (DF), Inverse Document Frequency (IDF), total frequency of the terms in the considered documents (totTF) and Robertson and Spark Jones term weighting [1] (RSJ).

Results. Table 1 reports the results obtained for the different term selection strategies adopted. DQC based on document representation obtained by DF is the most effective both in terms of precision and recall — the paired t-test shows that the only significant difference is between recall values obtained when using DF and IDF. Another finding concerns with capability of DQC to achieve a 100% precision using a relative small number k' of principal components and for almost all the considered topics; the only exception is the methodology implementation where IDF is adopted for term selection. Moreover, results depicted in Figure 1 show a positive correlation between the adopted number of components and precision (corr =0.312), whereas a negative correlation with recall (corr =-0.257).

Here, we summarize important findings of these experiments we compare with some of the claims of the seminal work [3]:

– the concept of "very large" dimensions may differ by order of magnitudes from one research field to another, and even though "DQC can handle a large number of features without much difficulty", after thousands of tests we may conclude that it would not be advisable to use it as an online learning method;



(a) (b)

Fig. 1. The figures depict the relationship between k' and precision (Figure 1a) and recall (Figure 1b). Values refer to the results for all the term selection strategies.

- a high recall implies a very low precision, and the number and the purity of clusters varies unpredictably with k'. This goes in the opposite direction of "The quality of the clustering degrades very slowly with loss in accuracy";
- there is indeed one extremely positive (and unexpected) aspect which is the possibility to tune the parameter k' such that in 90% of the cases one relevant document is found in one cluster with precision equal to one. This means that DQC is able to distill one pure document given a list of the top n documents retrieved by a search engine.

Acknowledgement. This work has been supported by the QONTEXT project under grant agreement N. 247590 (FP7/2007-2013) and the project FIRB "Un'inchiesta grammaticale sui dialetti italiani: ricerca sul campo, gestione dei dati, analisi linguistica" (FIRB – Futuro in ricerca 2008, cod. RBFR08KRA_003).

References

1. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Journal of the American Society for Information Science 27(3), 129–146 (1976)
2. Varshavsky, R., Linial, M., Horn, D.: Compact: A comparative package for clustering assessment. In: Chen, G., Pan, Y., Guo, M., Lu, J. (eds.) ISPA-WS 2005. LNCS, vol. 3759, pp. 159–167. Springer, Heidelberg (2005)
3. Weinstein, M., Horn, D.: Dynamic quantum clustering: A method for visual exploration of structures in data. Phys. Rev. E 80(6), 066117 (2009)
4. Xu, R., Ii: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)

Adding Emotions to Pictures

Claudia Hauff[1] and Dolf Trieschnigg[2]

[1] Delft University of Technology, Delft, The Netherlands
c.hauff@tudelft.nl
[2] University of Twente, Enschede, The Netherlands
r.b.trieschnigg@utwente.nl

Abstract. A large number of out-of-copyright children books are available online, but are not very attractive to children due to a lack of illustrations. Automatic text illustration may enhance the reading experience of these books, but inappropriate picture coloring may convey inappropriate emotions. Since already at a very early age, children can map colors to certain emotions, we propose an approach to automatically alter picture colors according to the emotion conveyed in the text.

1 Introduction

Initiatives such as Google Books[1] and Project Gutenberg[2] have made a large number of out-of-copyright books freely available as e-books, including classic works of children's literature. These e-books are currently not very appealing to children, as they are either offered in plain text or as images of scanned pages. Automatically adding pictures to such texts can make them more appealing. Existing work in automatic text illustration [5,6,8] focuses on factoid texts, while children texts often contain emotional passages such as the following extract from *The White Snake*, Grimm's Fairy Tales (1812):

> *The youth sat down in the garden and considered how it might be possible to perform this task, but he could think of nothing, and there he sat sorrowfully awaiting the break of day, when he should be led to death.*

We propose to post-process pictures that were identified by a text illustration algorithm according to the sentiment expressed in the text passage. Research in psychology has shown that children, even at a young age, associate certain colors with certain emotions. Based on this result, we derive a basic procedure that alters a picture's color scheme according to the emotion that shall be conveyed. We envision the process of illustrating children's literature as follows:

1. Determine a passage of text for illustration.
2. Run an automatic text illustration algorithm to determine a suitable picture, e.g., [5,6,8].

[1] http://books.google.com/
[2] http://www.gutenberg.org

G. Amati and F. Crestani (Eds.): ICTIR 2011, LNCS 6931, pp. 364–367, 2011.

3. Perform sentiment analysis to determine the conveyed emotion, e.g., [1].
4. Alter the picture colors according to the found sentiment.

In this poster, we describe our first step in the direction of altering pictures according to emotions. The rest of the poster is organized as follows: in Sec. 2 we outline the findings of children's ability to map colors to emotions. Then, in Sec. 3 we report our approach and results, followed by the conclusions (Sec. 4).

2 Children: Colors and Emotions

Color combinations are known to communicate moods and emotions [4], even at a very young age. Boyatzis et al. [2] studied the emotions children associate with colors. In the experiment, each child was given a color sample (red, blue, pink, etc.) and asked about her thoughts about the color. The sixty children (four to seven year old) in this study in general exhibited positive feelings towards bright colors and negative feelings towards dark colors. A gender gap was found with respect to dark colors: boys were more likely to have a positive feeling toward them than girls. Whether or not a color invokes a positive/negative emotion in an individual child often also depends on the child's personal experience with that color. Zentner et al. [7] investigated the same question with even younger children (three to four year old) and reported very similar results. In a different experimental setup, Burkitt et al. [3] asked four to eleven year old children to color three figures with colors of their choice: a happy, a nasty and a neutral figure. It was found that the children used their preferred colors for the happy figure and their least preferred colors for the nasty character, implicitly assigning emotions to colors.

These studies show that children are indeed able to map colors to emotions. Based on these results, we believe that by altering the color scheme of a picture we will be able to convey different sentiments.

3 Adding Emotions to Simple Pictures

The existing research on assigning images to text (automatic text illustration, text-to-picture), e.g., [5,6,8], focuses on identifying suitable images for *factoid* sentences. We assume for the purposes of this work, that such an algorithm identified a suitable picture from a pool of available pictures. As we focus on children's literature, we aim for simple pictures, such as those in Fig. 2(a), 2(d) and 2(g), which belong to the OpenClipart[3] library, our chosen picture corpus.

We also need a set of color schemes that convey different emotions. To this end, we collected color schemes from Kuler[4], a portal where users can create/upload/download color schemes, that were tagged with one of the following tags: *happy*, *sad* and *angry*. Examples of color schemes that we found for each

[3] http://openclipart.org/
[4] http://kuler.adobe.com/

tag are shown in Fig. 1. Each color scheme consists of four to five colors and in general, *happy* color schemes contain bright colors (as we would expect), while *angry* schemes often contain strong green and red shades. Color schemes tagged with *sad* mostly contain dull or dark colors.

(a) Happy (b) Sad (c) Angry

(d) Happy (e) Sad (f) Angry

Fig. 1. Color schemes taken from Kuler, tagged with one of three emotions

For each of the three emotions, we retrieved fifty different color schemes. Given a picture and an emotion, the most suitable color scheme is found as follows: all color schemes available for that emotion are evaluated for their similarity to the colors of the picture: for a color scheme with c colors, the c most dominant colors in the picture are determined and each color in the color scheme is matched to the most similar color in the picture. We calculate the distance between these color pairs and then select the color scheme with the smallest distance in color to the original picture. The c dominant colors in the picture are then replaced by the colors of the color scheme.

(a) Original (b) Happy (c) Sad

(d) Original (e) Happy (f) Angry

(g) Original (h) Angry (i) Sad

Fig. 2. The first column shows the original pictures, while the remaining pictures are the output of our color-altering algorithm

In Fig. 2, the results of our algorithm are exemplified. Fig. 2(a), 2(d) and 2(g) are the original pictures, as they occur in our corpus, while the remaining pictures were altered by our prototype system according to the emotion in question. The results indeed show the influence of the changing color schemes.

4 Conclusions

In this work, we have presented an idea and a first prototype of how to add emotions to simple pictures, that can be used in the automatic illustration of children's literature. We have devised a basic algorithm that relies on available color schemes (tagged by humans according to the emotion they express) to change the conveyed emotion of a picture. A natural next step for our work is to evaluate this algorithm in a user study with children; we are going to address the following questions: (i) do children recognize the different emotions that we aim to convey in our automatically altered pictures, and, (ii) are children interested in having pictures accompanying text that convey emotions?

A limitation of the current approach is the small size of the color schemes ($\approx$ 5 colors), which limits the usefulness of the approach to pictures with few dominant colors. In the future, we plan to combine different color schemes to have a larger color base as well as to rely on hue and saturation to change the emotional content of pictures.

References

1. Alm, C., Roth, D., Sproat, R.: Emotions from text: machine learning for text-based emotion prediction. In: PHLT 2005/EMNLP, pp. 579–586 (2005)
2. Boyatzis, C., Varghese, R.: Children's emotional associations with colors. The Journal of Genetic Psychology 155(1), 77–85 (1994)
3. Burkitt, E., Barrett, M., Davis, A.: Children's colour choices for completing drawings of affectively characterised topics. Journal of Child Psychology and Psychiatry 44(3), 445–455 (2003)
4. Gao, X., Xin, J.: Investigation of human's emotional responses on colors. Color Research & Application 31(5), 411–417 (2006)
5. Joshi, D., Wang, J., Li, J.: The Story Picturing Engine—a system for automatic text illustration. TOMCCAP 2(1), 68–89 (2006)
6. Mihalcea, R., Leong, C.: Toward communicating simple sentences using pictorial representations. Machine Translation 22(3), 153–173 (2008)
7. Zentner, M.: Preferences for colours and colour–emotion combinations in early childhood. Developmental Science 4(4), 389–398 (2001)
8. Zhu, X., Goldberg, A., Eldawy, M., Dyer, C., Strock, B.: A text-to-picture synthesis system for augmenting communication. In: AAAI 2007, pp. 1590–1595 (2007)

Author Index