Studies in Computational Intelligence 370

Alexander Mehler
Kai-Uwe Kühnberger
Henning Lobin
Harald Lüngen
Angelika Storrer
Andreas Witt (Eds.)

Modeling, Learning, and Processing of Text-Technological Data Structures

Springer

Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen, Angelika Storrer, and Andreas Witt (Eds.)

Modeling, Learning, and Processing of Text-Technological Data Structures

Studies in Computational Intelligence, Volume 370

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 348. Beniamino Murgante, Giuseppe Borruso, and Alessandra Lapucci (Eds.)
Geocomputation, Sustainability and Environmental Planning, 2011
ISBN 978-3-642-19732-1

Vol. 349. Vitor R. Carvalho
Modeling Intention in Email, 2011
ISBN 978-3-642-19955-4

Vol. 350. Thanasis Daradoumis, Santi Caballé, Angel A. Juan, and Fatos Xhafa (Eds.)
Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding, 2011
ISBN 978-3-642-19813-7

Vol. 351. Ngoc Thanh Nguyen, Bogdan Trawiński, and Jason J. Jung (Eds.)
New Challenges for Intelligent Information and Database Systems, 2011
ISBN 978-3-642-19952-3

Vol. 352. Nik Bessis and Fatos Xhafa (Eds.)
Next Generation Data Technologies for Collective Computational Intelligence, 2011
ISBN 978-3-642-20343-5

Vol. 353. Igor Aizenberg
Complex-Valued Neural Networks with Multi-Valued Neurons, 2011
ISBN 978-3-642-20352-7

Vol. 354. Ljupco Kocarev and Shiguo Lian (Eds.)
Chaos-Based Cryptography, 2011
ISBN 978-3-642-20541-5

Vol. 355. Yan Meng and Yaochu Jin (Eds.)
Bio-Inspired Self-Organizing Robotic Systems, 2011
ISBN 978-3-642-20759-4

Vol. 356. Slawomir Koziel and Xin-She Yang (Eds.)
Computational Optimization, Methods and Algorithms, 2011
ISBN 978-3-642-20858-4

Vol. 357. Nadia Nedjah, Leandro Santos Coelho, Viviana Cocco Mariani, and Luiza de Macedo Mourelle (Eds.)
Innovative Computing Methods and their Applications to Engineering Problems, 2011
ISBN 978-3-642-20957-4

Vol. 358. Norbert Jankowski, Włodzisław Duch, and Krzysztof Grąbczewski (Eds.)
Meta-Learning in Computational Intelligence, 2011
ISBN 978-3-642-20979-6

Vol. 359. Xin-She Yang, and Slawomir Koziel (Eds.)
Computational Optimization and Applications in Engineering and Industry, 2011
ISBN 978-3-642-20985-7

Vol. 360. Mikhail Moshkov and Beata Zielosko
Combinatorial Machine Learning, 2011
ISBN 978-3-642-20994-9

Vol. 361. Vincenzo Pallotta, Alessandro Soro, and Eloisa Vargiu (Eds.)
Advances in Distributed Agent-Based Retrieval Tools, 2011
ISBN 978-3-642-21383-0

Vol. 362. Pascal Bouvry, Horacio González-Vélez, and Joanna Kolodziej (Eds.)
Intelligent Decision Systems in Large-Scale Distributed Environments, 2011
ISBN 978-3-642-21270-3

Vol. 363. Kishan G. Mehrotra, Chilukuri Mohan, Jae C. Oh, Pramod K. Varshney, and Moonis Ali (Eds.)
Developing Concepts in Applied Intelligence, 2011
ISBN 978-3-642-21331-1

Vol. 364. Roger Lee (Ed.)
Computer and Information Science, 2011
ISBN 978-3-642-21377-9

Vol. 365. Roger Lee (Ed.)
Computers, Networks, Systems, and Industrial Engineering 2011, 2011
ISBN 978-3-642-21374-8

Vol. 366. Mario Köppen, Gerald Schaefer, and Ajith Abraham (Eds.)
Intelligent Computational Optimization in Engineering, 2011
ISBN 978-3-642-21704-3

Vol. 367. Gabriel Luque and Enrique Alba
Parallel Genetic Algorithms, 2011
ISBN 978-3-642-22083-8

Vol. 368. Roger Lee (Ed.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2011, 2011
ISBN 978-3-642-22287-0

Vol. 369. Dominik Ryżko, Piotr Gawrysiak, Henryk Rybinski, and Marzena Kryszkiewicz (Eds.)
Emerging Intelligent Technologies in Industry, 2011
ISBN 978-3-642-22731-8

Vol. 370. Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen, Angelika Storrer, and Andreas Witt (Eds.)
Modeling, Learning, and Processing of Text-Technological Data Structures, 2011
ISBN 978-3-642-22612-0

Alexander Mehler, Kai-Uwe Kühnberger,
Henning Lobin, Harald Lüngen, Angelika Storrer,
and Andreas Witt (Eds.)

Modeling, Learning, and Processing of Text-Technological Data Structures

Springer

Editors

Dr. Alexander Mehler
Bielefeld University
Faculty of Linguistics and Literature
Universitätsstraße 25, 33615 Bielefeld
Germany
E-mail: Alexander.Mehler@uni-bielefeld.de

Kai-Uwe Kühnberger
University of Osnabrück
Institute of Cognitive Science
Albrechtstr. 28, 49076 Osnabrück
Germany
E-mail: kkuehnbe@uos.de

Prof. Dr. Henning Lobin
Justus-Liebig-Universität Gießen
Angewandte Sprachwissenschaft und
Computerlinguistik, Otto-Behaghel-Straße
10D, 35394 Gießen, Germany
E-mail:
Henning.Lobin@germanistik.uni-giessen.de

Harald Lüngen
Justus-Liebig-Universität Gießen
Angewandte Sprachwissenschaft und
Computerlinguistik, Otto-Behaghel-Straße
10D, 35394 Gießen, Germany
E-mail:
Harald.Luengen@germanistik.uni-giessen.de

Angelika Storrer
Technical University Dortmund
Institut für deutsche Sprache und Literatur
Emil-Figge-Straße 50, 44227 Dortmund
Germany
E-mail: Angelika.Storrer@uni-dortmund.de

Andreas Witt
Eberhard Karls Universität Tübingen
SFB 441 Linguistic Data Structures
Nauklerstraße 35, 72074 Tübingen
Germany
E-mail: Andreas.Witt@uni-tuebingen.de

ISBN 978-3-642-22612-0 e-ISBN 978-3-642-22613-7

DOI 10.1007/978-3-642-22613-7

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2011934155

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Contents

1 **Introduction: Modeling, Learning and Processing of Text-Technological Data Structures** 1
Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen, Angelika Storrer, Andreas Witt
1.1 Textual Units as Data Structures 1
1.2 Overview of the Book 3
1.2.1 Text Parsing: Data Structures, Architecture and Evaluation 3
1.2.2 Measuring Semantic Distance: Methods, Resources, and Applications 4
1.2.3 From Textual Data to Ontologies, from Ontologies to Textual Data 5
1.2.4 Multidimensional Representations: Solutions for Complex Markup 6
1.2.5 Document Structure Learning 8
1.2.6 Interfacing Textual Data, Ontological Resources and Document Parsing 9
References 11

Part I: Text Parsing: Data Structures, Architecture and Evaluation

2 **The MOTS Workbench** 15
Manfred Stede, Heike Bieler
2.1 Introduction and Overview 15
2.2 Background: Natural Language Processing Frameworks 17
2.3 The PAULA XML Format 19
2.3.1 PAULA: Logical Structure 20
2.3.2 PAULA: Physical Structure 20

2.4 Processing Pipeline . . . 22
2.4.1 Normalized Input Format: LDOC . . . 23
2.4.2 Tokenization . . . 25
2.4.3 Conversion to PAULA . . . 25
2.4.4 Integrating Analysis Components . . . 25
2.5 Analysis Components . . . 26
2.6 User Interface . . . 28
2.6.1 XML Inline Representation . . . 29
2.6.2 Visualization . . . 30
2.7 Current Developments and Conclusion . . . 31
References . . . 33

3 Processing Text-Technological Resources in Discourse Parsing . . . 35
Henning Lobin, Harald Lüngen, Mirco Hilbert, Maja Bärenfänger
3.1 Introduction . . . 35
3.2 Corpus . . . 37
3.3 Architecture . . . 39
3.4 Parser . . . 40
3.4.1 Initialisation . . . 40
3.4.2 Cascade Step . . . 41
3.4.3 Generic Annotation Parser (GAP) . . . 43
3.4.4 Traversing the Chart . . . 52
3.5 Evaluation . . . 52
3.6 Conclusion . . . 54
References . . . 56

Part II: Measuring Semantic Distance: Methods, Resources, and Applications

4 Semantic Distance Measures with Distributional Profiles of Coarse-Grained Concepts . . . 61
Graeme Hirst, Saif Mohammad
4.1 Semantic Distance . . . 61
4.2 Measures of Semantic Distance . . . 63
4.3 A Hybrid Method for Semantic Distance Measures . . . 65
4.4 Evaluation in Monolingual Applications . . . 66
4.5 Extension to Cross-Lingual Applications . . . 69
4.5.1 Method . . . 69
4.5.2 Evaluation . . . 69
4.6 Antonymy and Word Opposition . . . 72
4.6.1 Contrasting Categories . . . 74
4.6.2 Degree of Antonymy . . . 75
4.6.3 Evaluation . . . 75
4.7 Conclusion . . . 76
References . . . 77

5 Collecting Semantic Similarity Ratings to Connect Concepts in Assistive Communication Tools 81
Sonya Nikolova, Jordan Boyd-Graber, Christiane Fellbaum
5.1 Background and Motivation 81
5.1.1 Aphasia 82
5.1.2 ViVA: Visual Vocabulary for Aphasia 82
5.2 The Design of ViVA 83
5.3 The Organization of Words 84
5.3.1 WordNet and Evocation 85
5.4 Building the Visual Vocabulary for Aphasia 86
5.4.1 Collecting Inexpensive Ratings from Untrained Annotators 87
5.4.2 Method 87
5.5 Results 88
5.6 Discussion 90
References 91

Part III: From Textual Data to Ontologies, from Ontologies to Textual Data

6 An Introduction to Hybrid Semantics: The Role of Cognition in Semantic Resources 97
Alessandro Oltramari
6.1 Introduction 97
6.2 Statics and Dynamics of Cognition 98
6.3 From Conceptualization to Specification: The Ontological Level 100
6.4 Building 'Hybrid' Semantic Technologies 102
6.5 The Project of a Collaborative Hybrid Semantic Resource 103
6.6 Conclusion 107
References 108

7 Modal Logic Foundations of Markup Structures in Annotation Systems 111
Marcus Kracht
7.1 Introduction 111
7.2 Some Elements of Modal Logic 112
7.3 Classes of Models 115
7.4 Modal Logic and DOMs 117
7.5 XPath 120
7.6 Paths in Dynamic Logic 123
7.7 Conclusion 126
References 126

8 Adaptation of Ontological Knowledge from Structured Textual Data . . . 129
Tonio Wandmacher, Ekaterina Ovchinnikova, Uwe Mönnich, Jens Michaelis, Kai-Uwe Kühnberger
8.1 Introduction . . . 129
8.1.1 Project Context . . . 131
8.2 Theoretical Background . . . 131
8.2.1 Annotation Graphs and Their Logical Representation . . . 131
8.2.2 Ontologies and Description Logics . . . 134
8.3 Automatic Extraction of Ontological Knowledge from Texts . . . 135
8.3.1 Existing Approaches in Ontology Learning . . . 135
8.4 Our Proposal . . . 138
8.5 Axiom Rewriting Procedure . . . 141
8.5.1 Transforming Textual Input into OWL Axioms . . . 141
8.5.2 Discussion . . . 144
8.6 Adaptivity . . . 145
8.6.1 Terminological Inconsistency . . . 146
8.6.2 Adaptation Procedure . . . 147
8.7 Conclusions and Future Work . . . 150
References . . . 151

Part IV: Multidimensional Representations: Solutions for Complex Markup

9 Ten Problems in the Interpretation of XML Documents . . . 157
C.M. Sperberg-McQueen, Claus Huitfeldt
9.1 Background . . . 157
9.1.1 Derivation of Inferences . . . 158
9.1.2 An Example . . . 160
9.2 The Ten Problems . . . 161
9.2.1 The Arity of Statements . . . 161
9.2.2 The Form of Inference Rules . . . 162
9.2.3 Deixis . . . 163
9.2.4 Inheritance . . . 164
9.2.5 Overriding . . . 165
9.2.6 Conflict and Union . . . 167
9.2.7 Milestones . . . 168
9.2.8 The Universe of Discourse . . . 168
9.2.9 Definite Descriptions and Multiple References to the Same Individual . . . 170
9.2.10 Certainty and Responsibility . . . 172
References . . . 173

10 Markup Infrastructure for the Anaphoric Bank: Supporting Web Collaboration 175
Massimo Poesio, Nils Diewald, Maik Stührenberg, Jon Chamberlain, Daniel Jettka, Daniela Goecke, Udo Kruschwitz
10.1 Introduction 175
10.2 How Data Is Added to the Anaphoric Bank 177
10.2.1 Filtering Criteria 177
10.2.2 Data That Has Already Been Annotated 178
10.2.3 Using the Expert Annotation Tool 178
10.2.4 Using a Non-expert Annotation Game 181
10.3 Architecture for the Anaphoric Bank 184
10.3.1 SGF – The XML Architecture Part 184
10.3.2 Database Format 190
10.4 Conclusion 192
References 193

11 Integrated Linguistic Annotation Models and Their Application in the Domain of Antecedent Detection 197
Andreas Witt, Maik Stührenberg, Daniela Goecke, Dieter Metzing
11.1 Introduction 197
11.2 Anaphora Resolution 200
11.3 Logical Document Structure 201
11.3.1 What Is Logical Document Structure? 201
11.3.2 Application of Logical Document Structure for Linguistic Tasks 202
11.3.3 XML-Annotation of Logical Document Structure 204
11.4 Integration of Resources 204
11.4.1 Representation Formats 205
11.4.2 Sekimo Generic Format and XStandoff 207
11.4.3 Antecedent Candidate List 209
11.5 Results of a Corpus Study 210
11.6 Conclusion 214
References 215

Part V: Document Structure Learning

12 Machine Learning for Document Structure Recognition 221
Gerhard Paaß, Iuliu Konya
12.1 Introduction 221
12.2 Document Analysis for Large-Scale Processing 222
12.2.1 Geometric Layout Analysis 224
12.2.2 Logical Layout Analysis 227
12.2.3 Minimum Spanning Tree-Based Logical Layout Analysis 229
12.2.4 Evaluation 231

12.3 Estimating Document Structure by Conditional Random Fields . . . 236
12.3.1 Basic Model . . . 236
12.3.2 Application of Linear-Chain CRFs to Structure Information Extraction . . . 239
12.3.3 Discriminative Parsing Models . . . 240
12.3.4 Graph-Structured Model . . . 241
12.4 Conclusion . . . 244
References . . . 245

13 Corpus-Based Structure Mapping of XML Document Corpora: A Reinforcement Learning Based Model . . . 249
Francis Maes, Ludovic Denoyer, Patrick Gallinari
13.1 Introduction and Motivation . . . 249
13.2 Related Work . . . 252
13.3 Structure Mapping Task . . . 253
13.3.1 Task Description . . . 253
13.3.2 Notations . . . 254
13.3.3 Complexity of Inferring and Learning Mappings . . . 254
13.4 Reinforcement Learning Based Model . . . 255
13.4.1 Informal Description . . . 255
13.4.2 Deterministic Markov Decision Process . . . 256
13.4.3 Modeling the Policy . . . 258
13.4.4 Evaluating and Learning the Policy . . . 259
13.5 Experiments . . . 260
13.5.1 Datasets . . . 260
13.5.2 Evaluation Measures . . . 262
13.5.3 Comparison to Baseline Models . . . 263
13.5.4 Results . . . 263
13.6 Conclusion . . . 265
References . . . 265

14 Learning Methods for Graph Models of Document Structure . . . 267
Peter Geibel, Alexander Mehler, Kai-Uwe Kühnberger
14.1 Introduction . . . 268
14.2 Directed Generalized Trees . . . 269
14.3 Quantitative Structure Analysis by Example of Websites . . . 272
14.4 Kernel Methods . . . 273
14.4.1 The Soft Tree Kernel for GTs . . . 274
14.4.2 The Soft GT Kernel . . . 278
14.4.3 The INDIGO Context and Label Sequence Kernels . . . 280
14.5 Experiments . . . 286
14.5.1 The Corpus . . . 286
14.5.2 Quantitative Structure Analysis . . . 287
14.5.3 Kernel Methods . . . 291

14.6 Conclusion 293
References 294

15 Integrating Content and Structure Learning: A Model of Hypertext Zoning and Sounding 299
Alexander Mehler, Ulli Waltinger
15.1 Introduction 299
15.1.1 Webgenre Learning in a Two-Level Perspective 300
15.1.2 Thematic-Generic Tracking, Zoning and Sounding 303
15.2 A Two-Level Model of Logical Web Document Structures 305
15.2.1 Hypertext Stage Classifier 306
15.2.2 Hypertext Stage Grammar and Type Classifier 310
15.2.3 Experiments 310
15.3 Thematic-Generic Sounding in the Web 314
15.4 Bounds of Thematic Sounding in Wikipedia 318
15.4.1 Dominator, Successor and Trigger Sets 320
15.4.2 Statistical Moments of Trigger and Dominator Sets 324
15.5 Conclusion 325
References 326

Part VI: Interfacing Textual Data, Ontological Resources and Document Parsing

16 Learning Semantic Relations from Text 333
Gerhard Heyer
16.1 Introduction 333
16.2 How Is It Possible to Automatically Process *Meaning*? 334
16.3 Some Filters for Semantic Relations 340
16.4 An Architecture for Learning Semantic Relations 342
References 345

17 Modelling and Processing Wordnets in OWL 347
Harald Lüngen, Michael Beißwenger, Bianca Selzam, Angelika Storrer
17.1 Research Context and Motivation 348
17.2 Resources 349
17.2.1 GermaNet 349
17.2.2 TermNet 350
17.2.3 GermaTermNet 351
17.3 Modelling Wordnet-Like Resources in OWL 352
17.3.1 Basic Options 352
17.3.2 Related Work 354
17.3.3 OWL Models for GermaNet, TermNet, and GermaTermNet 355
17.4 Processing WordNet-Like Resources in OWL 363

17.4.1 Processing the OWL Models of TermNet 363
17.4.2 Processing the OWL Full Version of GermaTermNet . . . 369
17.5 Conclusion 372
References 374

18 Exploring Resources for Lexical Chaining: A Comparison of Automated Semantic Relatedness Measures and Human Judgments 377
Irene Cramer, Tonio Wandmacher, Ulli Waltinger
18.1 Motivation 377
18.2 Lexical Chaining 378
18.3 Related Work 379
18.3.1 Budanitsky and Hirst 380
18.3.2 Boyd-Graber et al. 381
18.4 Semantic Relatedness Measures 382
18.4.1 Net-Based Measures 382
18.4.2 Distributional Measures 385
18.4.3 Wikipedia-Based Measures 387
18.5 Evaluation 387
18.5.1 Method 387
18.5.2 Results 389
18.5.3 Meta-level Evaluation 392
18.6 Conclusions and Future Work 393
References 394

Author Index 397

List of Contributors

Bärenfänger, Maja
Applied and Computational Linguistics,
Justus Liebig University
Gießen, Germany
E-mail: Maja.Baerenfaenger@germanistik.uni-giessen.de

Beißwenger, Michael
Institute for German Language and
Literature, TU Dortmund University
Dortmund, Germany
E-mail:
michael.beisswenger@tu-dortmund.de

Bieler, Heike
Applied Computational Linguistics,
University of Potsdam
Potsdam, Germany
E-mail: bieler@uni-potsdam.de

Boyd-Graber, Jordan
School of Information Studies,
University of Maryland
College Park MD, USA
E-mail: jbg@umiacs.umd.edu

Chamberlain, Jon
School of Computer Science and Electronic
Engineering, University of Essex
Colchester, United Kingdom
E-mail: jchamb@essex.ac.uk

Cramer, Irene
Institute for German Language and
Literature, TU Dortmund University
Dortmund, Germany
E-mail: irene.cramer@tu-dortmund.de

Denoyer, Ludovic
Computer Science Laboratory,
University of Paris 6
Paris, France
E-mail: ludovic.denoyer@lip6.fr

Diewald, Nils
Faculty of Technology, Bielefeld University
Bielefeld, Germany
E-mail: nils.diewald@uni-bielefeld.de

Fellbaum, Christiane
Department of Computer Science,
Princeton University
Princeton NJ, USA
E-mail: fellbaum@princeton.edu

Gallinari, Patrick
Computer Science Laboratory,
University of Paris 6
Paris, France
E-mail: patrick.gallinari@lip6.fr

Geibel, Peter
Electrical Engineering and Computer
Sciences, Technical University Berlin
Berlin, Germany
E-mail: info@peter-geibel.de

Goecke, Daniela
Faculty of Linguistics and Literary Studies,
Bielefeld University, Bielefeld, Germany
E-mail: daniela.goecke@uni-bielefeld.de

Heyer, Gerhard
Institut for Computer Science,
Leipzig University
Leipzig, Germany
E-mail: heyer@informatik.uni-leipzig.de

Hilbert, Mirco
Applied and Computational Linguistics,
Justus Liebig University
Gießen, Germany
E-mail: Mirco.Hilbert@germanistik.uni-giessen.de

Hirst, Graeme
Department of Computer Science,
University of Toronto
Toronto, Ontario, Canada
E-mail: gh@cs.toronto.edu

Huitfeldt, Claus
Department of Philosophy,
University of Bergen
Bergen, Norway
E-mail: Claus.Huitfeldt@fof.uib.no

Jettka, Daniel
Faculty of Linguistics and Literary Studies,
Bielefeld University
Bielefeld, Germany
E-mail: daniel.jettka@uni-bielefeld.de

Kracht, Marcus
Faculty of Linguistics and Literary Studies,
Bielefeld University
Bielefeld, Germany
E-mail: marcus.kracht@uni-bielefeld.de

Konya, Iuliu Vasile
Fraunhofer Institute for Intelligent Analysis
and Informations Systems IAIS
Schloss Birlinghoven, Germany
E-mail:
iuliu.vasile.konya@iais.fraunhofer.de

Kruschwitz, Udo
School of Computer Science and Electronic
Engineering, University of Essex
Colchester, United Kingdom
E-mail: udo@essex.ac.uk

Kühnberger, Kai-Uwe
Institute of Cognitive Science,
University of Osnabrück
Osnabrück, Germany
E-mail: kkuehnbe@uos.de

Lobin, Henning
Applied and Computational Linguistics,
Justus Liebig University
Gießen, Germany
E-mail: Henning.Lobin@germanistik.uni-giessen.de

Lüngen, Harald
Institut für Deutsche Sprache,
Programmbereich Korpuslinguistik
Mannheim, Germany
E-mail: luengen@ids-mannheim.de

Maes, Francis
Computer Science Laboratory,
University of Paris 6
Paris, France
E-mail: francis.maes@lip6.fr

Mehler, Alexander
Computer Science and Mathematics,
Goethe-University Frankfurt
Frankfurt am Main, Germany
E-mail: Mehler@informatik.uni-frankfurt.de

Metzing, Dieter
Faculty of Linguistics and Literary Studies,
Bielefeld University
Bielefeld, Germany
E-mail: dieter.metzing@uni-bielefeld.de

Michaelis, Jens
Faculty of Linguistics and Literary Studies,
Bielefeld University
Bielefeld, Germany
E-mail: jens.michaelis@uni-bielefeld.de

Mönnich, Uwe
Department of Linguistics,
University of Tübingen
Tübingen, Germany
E-mail: uwe.moennich@uni-tuebingen.de

Mohammad, Saif
Institute for Information Technology,
National Research Council Canada
Ottawa, Ontario, Canada
E-mail: saif.mohammad@nrc-cnrc.gc.ca

Nikolova, Sonya
Department of Computer Science,

Princeton University
Princeton NJ, USA
E-mail: nikolova@princeton.edu

Oltramari, Alessandro
Psychology Department,
Carnegie Mellon University
Pittsburgh, USA
E-mail: aoltrama@andrew.cmu.edu

Ovchinnikova, Ekaterina
Institute of Cognitive Science,
University of Osnabrück
Osnabrück, Germany
E-mail: e.ovchinnikova@gmail.com

Paaß, Gerhard
Fraunhofer Institute for Intelligent Analysis
and Informations Systems IAIS
Schloss Birlinghoven, Germany
E-mail: Gerhard.Paass@iais.fraunhofer.de

Poesio, Massimo
School of Computer Science and Electronic
Engineering, University of Essex
Colchester, United Kingdom
E-mail: poesio@essex.ac.uk

Selzam, Bianca
Institute for German Language and
Literature, TU Dortmund University
Dortmund, Germany
E-mail: bianca.stockrahm@tu-dortmund.de

Sperberg-McQueen, C. M.
Black Mesa Technologies
Española, New Mexico , USA
E-mail: cmsmcq@blackmesatech.com

Stede, Manfred
Applied Computational Linguistics,
University of Potsdam
Potsdam, Germany
E-mail: stede@uni-potsdam.de

Storrer, Angelika
Institute for German Language and
Literature, TU Dortmund University
Dortmund, Germany
E-mail: angelika.storrer@tu-dortmund.de

Stührenberg, Maik
Faculty of Linguistics and Literary Studies,
Bielefeld University
Bielefeld, Germany
E-mail: maik.stuehrenberg@uni-bielefeld.de

Waltinger, Ulli
Faculty of Technology, Bielefeld University
Bielefeld, Germany
E-mail: Ulli.Waltinger@uni-bielefeld.de

Wandmacher, Tonio
Systran S.A.
Paris, France
E-mail: tonio.wandmacher@gmail.com

Witt, Andreas
Institut für Deutsche Sprache,
Zentrale Forschung
Mannheim, Germany
E-mail: witt@ids-mannheim.de

Chapter 1
Introduction: Modeling, Learning and Processing of Text-Technological Data Structures

Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen,
Angelika Storrer, and Andreas Witt

1.1 Textual Units as Data Structures

Researchers in many disciplines, sometimes working in close cooperation, have been concerned with modeling textual data in order to account for texts as the prime information unit of written communication. The list of disciplines includes computer science and linguistics as well as more specialized disciplines like computational linguistics and text technology. What many of these efforts have in common

Alexander Mehler
Computer Science and Mathematics, Goethe-Universität Frankfurt, Senckenberganlage 31,
D-60325 Frankfurt am Main, Germany
e-mail: Mehler@em.uni-frankfurt.de

Kai-Uwe Kühnberger
Institute of Cognitive Science, Universität Osnabrück, Albrechtstraße 28,
D-49076 Osnabrück, Germany
e-mail: kkuehnbe@uos.de

Henning Lobin
Applied and Computational Linguistics, Justus-Liebig-Universität Gießen,
Otto-Behaghel-Straße 10D, D-35394 Gießen, Germany
e-mail: Henning.Lobin@germanistik.uni-giessen.de

Harald Lüngen
Institut für Deutsche Sprache, Programmbereich Korpuslinguistik, R5, 6-13,
D-68161 Mannheim, Germany
e-mail: luengen@ids-mannheim.de

Angelika Storrer
Institute for German Language and Literature, Technische Universität Dortmund,
Emil-Figge-Straße 50, D-44221 Dortmund, Germany
e-mail: angelika.storrer@tu-dortmund.de

Andreas Witt
Institut für Deutsche Sprache, Zentrale Forschung, R 5, 6-13, D-68161 Mannheim, Germany
e-mail: witt@ids-mannheim.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 1–11.
springerlink.com

is the aim to model textual data by means of abstract data types or data structures[1] that support at least the semi-automatic processing of texts in any area of written communication.[2]

Generally speaking, an abstract data type is a mathematical model of a certain range of data together with operations defined on that model in such a way that they can be performed automatically on the data [2]. From this point of view, natural language texts are a very special sort of data that requires very specific data structures for being processed automatically – of course, there is no single data structure that can model all aspects of natural language texts. A central characteristic of this task is structural uncertainty.

Structural Uncertainty

In order to understand this notion, take the example of a tree-like model of text structure as proposed, for example, by *Rhetorical Structure Theory* (RST) [7]. Any text can be made an object of operations (e.g., linkage of text spans) related to this data structure by virtue of interpreting (e.g., delimiting) its constituents (e.g., text spans). However, due to the semiotic nature of natural language texts, this interpretation is, in principle, open, that is, not necessarily determined by the data itself. Moreover, the relevant context that allows for determining this interpretation is not necessarily known in advance, nor fixed once and for all. Take the example of rhetorical structure as modeled in RST: on the one hand, there is disagreement on the range of rhetorical relations that can actually hold between text spans [8]. On the other hand, there is uncertainty about which relation actually holds between a given pair of spans – even if the set of rhetorical relations is fixed in advance – a problem known as inter-annotator disagreement [3]: often, humans diverge in their interpretation of the same data with respect to the same data structure.

In other words, the way textual data is structured is not necessarily clear from the data itself. It may be the result of semantic or even pragmatic interpretations that are ultimately carried out by humans going beyond (and possibly far beyond) the linguistic context of the data to be interpreted. As a consequence, the structure of a given text as an instance of a given data structure can be very *uncertain* as this structure does not need to be reflected by the text's constituents in an obvious way.[3]

Thus, the semi-automatic or even manual annotation of textual data, that is, its informational enrichment to make its structure explicit according to the underlying data structure, is a central task of text technology and related disciplines. This also includes linking textual data with other linguistic resources. Any such annotation and linkage, whether done manually, semi-automatically, or fully automatically, would support various tasks of text processing (e.g., information extraction [10], text categorization [9], text mining [4], text summarization [6], topic modeling

[1] In this chapter, we use these terms interchangeably.

[2] Throughout this volume, we concentrate on written communication.

[3] This sort of structural uncertainty should not be confused with the notion of semi-structured data [1, 10].

[5], or discourse parsing [8]). Any of these tasks requires expressive data structures in conjunction with efficient operations that together allow for moving closer to the goal of *automating* text processing.

This book, "*Modeling, Learning and Processing of Text-Technological Data Structures*", deals with such data structures. Here we focus on theoretical foundations of representing natural language texts as well as on concrete operations of automatic text processing. Following this integrated approach, the present volume includes contributions to a wide range of topics in the context of processing of textual data. This relates to the learning of ontologies from natural language texts, annotation and automatic parsing of texts as well as the detection and tracking of topics in texts and hypertexts. In a nutshell, the book brings together a wide range of approaches to procedural aspects of text technology as an emerging scientific discipline. It includes contributions to the following areas:

- formalizing annotations of textual units
- extracting knowledge and mining ontologies from texts
- building lexical and terminological resources
- machine learning of document structures
- classifying and categorizing texts
- detecting and tracking topics in texts
- parsing discourse

This book addresses researchers who want to get familiar with theoretical developments, computational models and their empirical evaluation in these fields of research. It is intended for all those who are interested in standards of representing textual data structures, the use of these data structures in various fields of application (such as topic tracking, ontology learning and document classification) and their formal-mathematical modeling. In this sense, the volume concerns readers from many disciplines such as text and language technology, natural language processing, computational linguistics and computer science.

1.2 Overview of the Book

1.2.1 Text Parsing: Data Structures, Architecture and Evaluation

Part I of the volume focuses on the automatic analysis of text structure. By analogy to the analysis of sentence structure, it is often called text or discourse parsing. Fundamental aspects of text parsing are the data structures used, principles of system architecture, and the evaluation of these parsing systems.

The first chapter on "*The MOTS Workbench*" by Manfred Stede and Heike Bieler deals with the standardization of processing frameworks for text documents – an important issue for language technology for quite some time. The authors examine one particular framework, the MOTS workbench, and describe the overall architecture, the analysis modules that have been integrated into the workbench, and the user interface. After five years of experience with this workbench, they provide a critical evaluation of its underlying design decisions and draw conclusions for future development.

The second chapter, "*Processing Text-Technological Resources in Discourse Parsing*" by Henning Lobin, Harald Lüngen, Mirco Hilbert and Maja Bärenfänger, investigates discourse parsing of complex text types such as scientific research articles. Discourse parsing is seen from a text-technological point of view as the addition of a new layer of structural annotation for input documents already marked up on several linguistic annotation levels. The GAP parser described in this chapter generates discourse structures according to a relational model of text structure, Rhetorical Structure Theory. The authors also provide an evaluation of the parser by comparing it with reference annotations and with recently developed systems with a similar task. In general, both chapters show that text or discourse parsing is no longer of purely experimental interest, but can yield useful results in the analysis of huge amounts of textual data. This will eventually lead to parsing applications that pave the way to new generations of content-based processing of documents in text technology.

1.2.2 Measuring Semantic Distance: Methods, Resources, and Applications

The determination of semantic distance between lexical units is crucial for various applications of natural language processing; in the context of text technology semantic distance measures were used to reconstruct (and annotate) cohesive and thematic text structures by means of so-called lexical chains. The two contributions of Part II deal with methods and resources to determine semantic distance and semantic similarity in different application contexts.

In their chapter, "*Semantic distance measures with distributional profiles of coarse-grained concepts*", Graeme Hirst and Saif Mohammad first provide an overview of NLP applications using such measures. Then, they group the various approaches to calculate semantic distance into two classes: (1) resource-based measures which determine semantic distance by means of the structure of lexical resources such as thesauruses or word-nets; (2) distributional measures which compare distributional profiles of lexical units generated on the basis of text corpora. The authors list the strengths and limitations of the two measure classes and propose, as an alternative, a hybrid method which calculates distributional profiles not for word forms but for coarse-grained concepts defined on the basis of Roget-style thesaurus categories, disambiguating words attached to more than one concept with a bootstrapping approach. The evaluation results discussed in Section 3 of this chapter indicate that their concept-based hybrid method (using the BNC as a corpus and the Macquarie Thesaurus as a lexical resource) performs considerably better than the word-based distributional approach. However, the performance is still not at the level of the best resource-based measure obtained by using the Princeton WordNet as the lexical resource. However, not all languages dispose of resources with the coverage and quality of the Princeton WordNet. The authors show that for such languages, a good alternative might be an extension of their method which links the concepts of the English thesaurus to a bilingual lexicon with English as the target language. This can then generate concept-based distributional profiles for the lexicon's source language. This extended method was tested for German, using the bilingual lexicon BEOLINGUS and the taz-Corpus as resources. In the comparative evaluation, presented in Section 5.2 of the chapter, the extended method performed even better than the resource-based approach using the German word-net-style resource GermaNet. In their final section, the authors show how another extension of the concept-based method may help to determine different degrees of antonymy between pairs of lexical units in text corpora.

The Princeton WordNet has proven to be a widely-used and valuable lexical resource not only for computing semantic distance but for a broad range of other natural language processing applications. However, some approaches profit from complementing the part-of-speech-specific WordNet relations by cross-part-of-speech links between semantically similar and strongly associated concepts (like [dog] and [to bark], or [sky] and [blue]). In their chapter "*Collecting Similarity Ratings to Connect Concepts in Assistive Communication Tools*", Sonya Nikolova, Jordan Boyd-Graber, and Christiane Fellbaum describe such an application context: the structuring of the vocabulary in assistive communication tools, such that people with aphasia can retrieve words that express the concepts they want to communicate. In the multi-modal visual vocabulary component for these tools, concepts are represented in combination with pictures and sounds. With each concept being mapped to a core set of the Princeton WordNet, navigating through the vocabulary can be improved using WordNet's semantic and lexical relations between words and concepts. The authors identify the need to establish additional links between concepts which are strongly associated with each other in a specific context, based on human judgments on the strength of association between disambiguated words. Since experiments to gain such judgments in controlled studies with trained persons have proven to be time-consuming and expensive, the authors have developed and tested an alternative method using Amazon Mechanical Turk. The results of their experiments indicate that this method is indeed feasible for gathering a large number of association ratings at low cost and in a short amount of time, provided that reliability checks are applied to filter out invalid ratings.

1.2.3 From Textual Data to Ontologies, from Ontologies to Textual Data

Part III, "*From Textual Data to Ontologies, from Ontologies to Textual Data*", contains chapters that focus on semantic issues in text technology. Centered on the current standard of using ontologies for the coding of conceptual knowledge, this part covers semantic resources and cognitive constraints in the process of ontology creation by presenting a formal specification of the semantics of current markup standards and by proposing a general framework for the extraction and adaptation of ontological knowledge in text-technological applications.

Alessandro Oltramari's chapter "*An Introduction to Hybrid Semantics: the Role of Cognition in Semantic Resources*" argues, in a detailed way, for the consideration of cognitive structures and cognitive constraints in semantic technologies and, in particular, the process of ontology creation. The author argues that semantic approaches in text technology need to be enriched by modules that are informed by the cognitive structure of conceptualizations.

The chapter "*Modal Logic Foundations of Markup Structures in Annotation Systems*" by Marcus Kracht shows that it is useful to study connections between Markup languages and logical characterizations of these languages. The chapter shows, for example, that it is possible to derive complexity results of the query language XPath by simply transferring well-known model theoretic results of *Propositional Dynamic Logic* (PDL) to XPath.

The third chapter entitled "*Adaptation of Ontological Knowledge from Structured Textual Data*" by Tonio Wandmacher, Ekaterina Ovchinnikova, Uwe Mönnich, Jens Michaelis, and Kai-Uwe Kühnberger presents a general framework for the extraction of semantic knowledge from syntactically given information. The authors describe the transformation of this information to a logical representation and the adaptation of ontological knowledge using this new information. The framework builds on many well-known technologies and tools as, for example, WordNet and FrameNet. Further, it also builds on reasoning in description logic.

1.2.4 Multidimensional Representations: Solutions for Complex Markup

Text enrichment by the substantial addition of markup is one of the characteristics of the application of text-technological methods. Part IV – "*Multidimensional representations: Solutions for complex markup*" – addresses problems related to the creation, interpretation, and interchange of richly annotated textual resources. This part includes one chapter devoted to general problems of annotated documents, and two case studies that reveal insights into specific aspects of creating and representing annotations. The chapter "*Ten problems in the interpretation of XML documents*" by C. M. Sperberg-McQueen and Claus Huitfeldt examines questions related to the semantics of document markup. The problems addressed in this chapter are related to a formalization of the interpretation of annotations. The aim of this formalization is, as it is quite often in computational linguistics and text technology, to enhance the specification of algorithms for an automatic processing of potentially richly annotated resources. The methodology presented is based on a mapping from XML annotations to predicates of first-order logic. As the title indicates, the main part of the chapter focuses on the problems which arise when transforming markup into formulas of a logical calculus. The problems discussed concern the arity of statements, the form of inference rules, deictic reference to other parts of the document, the inheritance of properties and how to override them, the treatment of a union of properties with conflicting values, the treatment of milestone elements, the definition of the universe of discourse, the occurrence of definite descriptions and multiple references to the same individual, and the recording of uncertainty and responsibility. For each of these items, a description of the problem is presented along with a proposal of a corresponding solution.

The second chapter in Part IV is entitled "*Markup Infrastructure for the Anaphoric Bank: Supporting Web Collaboration*". The authors Massimo Poesio, Nils Diewald, Maik Stührenberg, Jon Chamberlain, Daniel Jettka, Daniela Goecke, and Udo Kruschwitz focus on the creation of specific high-quality annotations, and their publication. A collaborative approach is taken for the annotation process described in this chapter. Two different methods have been developed to support annotators. The first method is based on the special purpose editor Serengeti, a web application for modern browsers. Tools like this facilitate the process of annotation considerably, but they require expert knowledge. Since non-expert users could also provide high-quality annotations, a second method was also used. The annotations of the non-experts are produced by means of a game in which, e.g., a user tries to find the correct antecedents of an anaphora. All the annotated corpora are accessible through the web. The structure of the annotated resources is based on an XML-conformant markup scheme that makes use of the stand-off technique. This format is imported into a relational database system that allows for fast access to the data.

Part IV closes with the chapter "*Integrated Linguistic Annotation Models and their Application in the Domain of Antecedent Detection*" by Andreas Witt, Maik Stührenberg, Daniela Goecke, and Dieter Metzing. It deals with potential benefits of using information about the logical document structure for the task of anaphora resolution. Even though the investigations of these effects show only a weak influence of the logical document structure for the task (as reported in this chapter), the findings give insights into complex markup and multidimensional representations. The ways to integrate different information types in text resources are discussed at length. The discussion is based on a corpus study with a focus on logical document structure and anaphoric relations in texts. To do this study, the texts in the corpus were annotated according to the annotation schemes for these two different levels of information. The level of logical document structure was annotated according to a document grammar that uses elements from the wide-spread annotation schemes XHTML and DocBook. The other level covers the annotation of anaphoric elements, antecedents and the types of anaphoric relations. The semi-manual annotation of this anaphora level was carried out with the help of the special purpose editor Serengeti. From the point of view of multidimensional representation by means of complex markup, this chapter on the one hand presents techniques that allow for the integration of heterogeneous types of annotations in a single representation. On the other hand it presents a corpus study that investigates the interaction between diverse levels of information. The methodology described could also be adapted to examine the existence of interrelations between different linguistic levels.

1.2.5 Document Structure Learning

The chapters in Part V of the volume deal with document structure learning. One of the chapters focuses on "classical" texts, all other chapters deal with web documents. In this way, Part V includes models of learning the structure of texts and of hypertexts. In both cases, the *Logical Document Structure* (LDS) of a textual unit is used as a reference point for learning. In the case of texts, the LDS can be identified with their hierarchical division into sections, subsections, etc. down to the level of sentences and their lexical constituents. In the case of hypertexts, things are more complicated since hyperlinks give rise to non-hierarchical, network-like structures.

Note that the LDS is used as the reference point of many text-linguistic models that focus, for example, on rhetorical, argumentative or thematic structures. However, approaches to information retrieval that aim to go beyond the bag-of-words approach are likewise in need of models of logical document structure that can be easily induced from text instances. The chapter "*Machine Learning for Document Structure Recognition*" by Gerhard Paaß and Iuliu Konya describes approaches in this field of research. Based on the notion of a *Minimum Spanning Tree* (MST), the chapter describes an algorithm of layout-based document analysis that processes images of document pages to identify their LDS. As a matter of fact, this approach to *logical layout analysis* is highly important in the field of digitizing historical documents. However, a central problem of structure recognition relates to the variability of layout-based cues of document structuring. In order to tackle this challenge, approaches to machine learning are needed that deal with the uncertainty of layout-related manifestations of the LDS. Following this idea, the chapter reviews and describes approaches to document structure recognition that utilize *Conditional Random Field*s (CRF) as a learning method. Both classes of approaches, the MST- and the CRF-based approaches, are discussed in terms of their F-measure-related evaluation.

The variety of layout structures is one source of the uncertainty about the structure of non-digitized documents. A related problem concerns the variety of formats that are used to represent already digitized documents, say, by means of (X)HTML, XML-DTD, or XML Schema. Moreover, for a wide range of semi-structured documents on the web, which have multiple sources and undergo frequent changes, one has no access to the underlying document schema (if such a schema exists at all). The chapter "*Corpus-Based Structure Mapping of XML Document Corpora: A Reinforcement Learning based Model*" by Francis Maes, Ludovic Denoyer, and Patrick Gallinari addresses this kind of structural variety. Starting from a document-centric perspective, they introduce an algorithm for automatically mapping documents that vary only by their format onto a mediating schema, which expresses the structural unity of these input documents. The algorithm for aligning documents by their structure works on a set of pairs of input-output documents and, thus, is supervised. The chapter provides an extensive evaluation of this approach by means of five different corpora including a corpus of documents from Wikipedia. These experiments show that generic models of learning web document structure are possible. This, in turn, focuses on one of the challenges of exploring information from the web that is restricted by the variety of document formats and schemata in use.

What makes the web unique in terms of document structure is its hyperlink-based structuring. That is, as instances of webgenres (i.e., types of web documents by analogy to text types), websites usually consist of several pages that are connected by hyperlinks. From a formal point of view, such documents can be seen as a special class of graphs with an internal hierarchical structure that is superimposed by graph-inducing links. Starting from this notion, the chapter "*Learning Methods for Graph Models of Document Structure*" by Peter Geibel, Alexander Mehler and Kai-Uwe Kühnberger describes two approaches to learning web document structures. First, it describes a range of kernel-based approaches that utilize structure-related kernels to build supervised classifiers of webgenres. The chapter then adopts quantitative structure analysis in order to arrive at an unsupervised classifier of the same range of webgenres. Using a corpus of three webgenres, the chapter provides empirical evidence into the learnability of hyperlink-based document structures on the level of websites.

A central aspect of learning web document structures is given by their internal and external structuring. More specifically, instances of webgenres are manifested by page-level units as well as by units across the border of single pages. Consequently, when trying to automatically delimit instances of webgenres, one has to process document-internal as well as document-external structures, whether hyperlink-based or not. The chapter "*Integrating Content and Structure Learning: A Model of Hypertext Zoning and Sounding*" by Alexander Mehler and Ulli Waltinger tackles this task. By integrating web content and structure mining, it introduces a classifier of page-internal, webgenre-specific staging together with an unsupervised algorithm of content tagging that utilizes Wikipedia as a social-semantic resource. In this way, the chapter focuses on the task of hypertext zoning, that is, of delimiting webgenre instances based on their content and structure. The chapter ends by outlining an approach to estimate bounds of thematic sounding in Wikipedia.

1.2.6 Interfacing Textual Data, Ontological Resources and Document Parsing

The three chapters in Part VI deal with the extraction of semantic relations from text, the evaluation of measures of the semantic relatedness of lexemes using linguistic resources, and the modeling of WordNet-like resources in a formalism for the representation of ontologies. Semantic relations include lexical-semantic relations like antonymy and meronymy, but also a more general semantic relatedness relation sometimes called association, or evocation.

The chapter "*Exploring Resources for Lexical Chaining: A Comparison of Automated Semantic Relatedness Measures and Human Judgments*" by Irene Cramer, Tonio Wandmacher, and Ulli Waltinger gives an overview of 16 different measures of semantic relatedness using four types of linguistic resources in their calculations. They categorize the measures according to three types, that is, net-based measures, distributional measures, and Wikipedia-based measures, and evaluate them by comparing their performance with human judgments on two lists of word pairs. The results show that among the three types of measures, distributional measures perform better than the other two types, while in general the correlations of all 16 measures with human judgments are not high enough to accurately model lexical cohesion as perceived by humans. In their conclusion, Cramer et al. argue that more research is needed in order to come up with a sounder theoretical foundation of semantic relatedness and to determine what kind of resource should be employed for what kind of task. To further these directions of research, they argue for the definition of a shared task by the research community.

The chapter "*Learning Semantic Relations from Text*" by Gerhard Heyer describes semantic relations in terms of the traditional structuralist notions of syntagmatic and paradigmatic relations between words. Accordingly, wordforms stand in a paradigmatic relation if their global contexts (statistically relevant syntagmatic relations captured in a co-occurrence set) are similar according to some (distributional) similarity measure. Semantic relations such as the hyponymy relation can then be derived by applying linguistic filters or constraints on the similarity measure. Two applications of iterative filtering and refining global contexts of words are exemplified, namely word sense disambiguation, and the identification of (near) synonymy. Heyer also sketches a language-independent, modular, web-service-oriented architecture for interactively learning semantic relations.

One type of resource frequently used in the calculation of semantic relatedness is lexical-semantic networks such as the Princeton WordNet for English. Recently, suggestions have been made to represent wordnets in formalisms designed for the representation of ontologies in order to provide better interoperability among lexical-semantic resources and to make them available for the Semantic Web. In their chapter – "*Modelling and Processing Wordnets in OWL*" – Harald Lüngen, Michael Beißwenger, Bianca Selzam, and Angelika Storrer argue that when modeling wordnets in OWL, it has to be decided whether to adhere to either a class model, an instance model, or a metaclass model. They discuss and compare the features of these three models by the example of the two resources GermaNet and TermNet. In several experiments, each model is assessed for its performance when querying and processing it in the context of automatic hyperlinking. Because of its compatibility with notions from traditional lexical semantics, they favor the metaclass model.

Acknowledgement

Like its predecessor "*Linguistic Modeling of Information and Markup Languages*" [11], this book is based on contributions and numerous discussions at a corresponding conference at the *Center for Interdisciplinary Research* (*Zentrum für interdisziplinäre Forschung*, ZiF) at Bielefeld University. In the case of the present volume, this relates to the conference on "*Processing Text-Technological Resources*", organized by the research group "*Text-technological*

Information Modeling"[4] that was funded from 2002 to 2009 by the German Research Foundation (DFG).

The editors gratefully acknowledge financial support by the DFG and by the ZiF. We also thank the series editor and all reviewers who helped to shape this book by means of their invaluable hints. Last but not least we gratefully acknowledge all authors for their contributions to this book.

References

[1] Abiteboul, S.: Querying semi-structured data. In: Afrati, F.N., Kolaitis, P.G. (eds.) ICDT 1997. LNCS, vol. 1186, pp. 1–18. Springer, Heidelberg (1996)

[2] Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Data Structures and Algorithms. Computer Science and Information Processing, Addison-Wesley, Reading, Massachusetts (1983)

[3] Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics 22, 249–254 (1996)

[4] Feldman, R., Sanger, J.: The Text Mining Handbook. In: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, Cambridge (2007)

[5] Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W.: Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates, Mahwah (2007)

[6] Mani, I.: Automatic Summarization. John Benjamins, Amsterdam (2001)

[7] Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8, 243–281 (1988)

[8] Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge (2000)

[9] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)

[10] Soderland, S.: Learning information extraction rules for semi-structured and free text. Machine Learning 34(1), 233–272 (1999)

[11] Witt, A., Metzing, D. (eds.): Linguistic Modeling of Information and Markup Languages. Springer, Dordrecht (2010)

[4] www.text-technology.de

Part I
Text Parsing: Data Structures, Architecture and Evaluation

Chapter 2
The MOTS Workbench

Manfred Stede and Heike Bieler

Abstract. Standardization of processing frameworks for text documents has been an important issue for language technology for quite some time. This paper states the motivation for one particular framework, the MOTS workbench, which has been under development at Potsdam University since 2005 for purposes of research and teaching. We describe the overall architecture, the analysis modules that have been integrated into the workbench, and the user interface. Finally, after five years of experiences with MOTS, we provide a critical evaluation of the design decisions that were taken and draw conclusions for future development.

2.1 Introduction and Overview

The development of general frameworks for quickly configuring natural language processing (NLP) applications started in the 1990s, with the release of GATE in 1996 being the most significant first milestone [6]. Nowadays, several of such frameworks are in successful use; we will provide a brief overview in Section 2.2. Our aim in this paper is to provide a description and critical review of our own framework, which has been implemented at Potsdam University since 2005. The MOTS (MOdular Text processing System) Workbench was devised as a framework[1] for integrating diverse NLP modules by means of a specific standoff XML

Manfred Stede · Heike Bieler
Applied Computational Linguistics, EB Cognitive Science, University of Potsdam,
Karl-Liebknecht-Str. 24-25, D-14476 Golm, Germany
e-mail: {stede,bieler}@uni-potsdam.de

[1] We use the term 'framework' in the same sense as [5]: a software infrastructure that supports the effective re-use of services for a particular domain (here: language processing). In Computer Science, a more specific term emphasizing the integration of heterogeneous components is 'middleware'. When graphical tools for configuring and testing systems are being added, the framework turns into a 'development system'; MOTS provides a first step here but does not go all the way, as will become clear later.

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 15–34.
springerlink.com

format that serves as “interlingua” or “pivot” format to mediate between modules. An important goal in designing this format (called PAULA for ‘Potsdamer Austauschformat für Linguistische Annotation’) was to serve both the worlds of manual annotation and of automatic processing:

- Manual annotation, in our setting, primarily serves the needs of a large collaborative research center on information structure (IS), where data of very different kinds and in different languages was hand-annotated with different types of information that can support IS research. *Heterogeneity* thus is a key concept here. The different annotation needs are best suited by different existing tools, which can be used to label the same data from different viewpoints. The challenge then is to merge the output formats of the annotation tools into a single linguistic database, where the data can be queried across the layers in order to check for correlations of features. This setting is described in detail in [4]
- Automatic processing, the focus of the present paper, was conceived in the same way: analysis modules contribute separate layers of annotation to a text document, and – in addition – a module in charge of a “complex” task should be able to access the results of modules that have already solved a less complex task. Thus layers of analysis are being stacked upon each other, and each layer can be stored in the PAULA format.

The basic idea therefore was to realize a layer-based annotation of text data, where layers might either be independent or in their annotations refer to other layers, and where the origin of the layer (manual annotation or automatic analysis) would not matter. Our emphasis on an XML-based pivot format results on the one hand from the need to make the output of manual annotation tools (such as MMAX2, EXMARaLDA, RSTTool, ANNOTATE) inter-operable, and on the other hand from the desire to integrate automatic analysis tools written in arbitrary programming languages – at present, MOTS includes modules written in Java, C, C++, Lisp, Perl, and Python.

A common problem for integrative approaches of the kind described here is to ensure that all annotations refer correctly to the intended spans of text. Our approach is to submit each incoming document to a preprocessing phase where sentence splitting and tokenisation is performed, and the basic layout of the document represented in a light-weight XML “logical document structure” format (henceforth called ‘LDOC’). The sentence and token boundaries are then enforced for all other modules – which can be difficult to do when integrating external modules that perform their own tokenization, thus generating the need for synchronization. One advantage of the standardized LDOC base format is that different types of documents (XML, HTML, plain text) can all be translated to LDOC during preprocessing, so that all subsequent analysis modules are not being bothered by idiosyncratic data formats.

The first task that MOTS was designed for was automatic text summarization. Our user interface thus was given the job to not only show all the intermediate analysis results (for debugging purposes) but also to allow for a flexible visualization of summaries (extracts) of different lengths. From the outset, we targeted a web-based UI, as development of the text summarizer involved external project partners who had to have quick access to our results. This decision was also supported (and proved very helpful) by the desire to use MOTS also for teaching purposes: In classes on NLP and text analysis, results can be shown and compared, and new modules implemented by students be integrated into the framework with relatively little effort. In this sense, we see MOTS as a 'workbench': a flexible environment for combining NLP modules and showing results. However, MOTS was not conceived as a framework for building real-time or even industry-standard applications – which is why we use the term 'NLP workbench' rather than 'Language Engineering (LE) workbench'.

Having described the "starting position" of the MOTS development, we now turn first to a brief review of related approaches along with a comparison to MOTS (section 2.2). Afterwards, we provide a description of the PAULA XML format (section 2.3) and give overviews of the overall architecture of the processing pipeline (section 2.4), the analysis modules we have integrated so far (section 2.5), and the user interface (section 2.6). Finally, we mention some ongoing work on MOTS, and provide a critical evaluation of the design decisions in the light of several years of experience with the system (section 2.7).

2.2 Background: Natural Language Processing Frameworks

The central task of an NLP framework is to support an effective re-use of processing modules – usually called *components* when part of a framework – and possibly also of linguistic (data) resources, and their flexible re-combination to systems made up of a particular set of components. Thus, a component would be in charge of a particular subtask, and the composition of the subtasks to the overall application is being designed with the help of the framework, which ensures the inter-operability of components. One of the first major steps to the design of such frameworks for language processing was the TIPSTER architecture [13], conceived at the time primarily for the task of information extraction. In general, we can distinguish between frameworks aiming primarily at document processing and handling rather diverse types of information, and those focusing more specifically on linguistic analysis, emphasizing the role of unifying *formalisms* for representing information and performing computations. A recent example for the latter is 'Heart of Gold' [19]; our focus in this paper, however, is on the former group.

When devising GATE[2], [5] characterized the basic design decision as one between the 'embedded' and the 'referential' type of annotation (p. 98), where the former was represented by SGML "inline" markup in the document itself, and the latter by a database model where columns in the table represent annotations referring to sequences of characters in the source document – i.e., the TIPSTER approach. GATE adopted the database solution, mainly because it seemed to provide

[2] `http://gate.ac.uk`

more efficient access in comparison to the inherently sequential data storage using SGML, and because of the possibility to represent graph structures in the annotations. Nowadays, SGML inline markup has developed to XML standoff markup, where each layer of annotation resides in a separate file, the source text document remains unchanged, and annotations merely *refer* to it (borrowing from TIPSTER's idea of the database table). This allows for more flexibility in the annotations (e.g., it is possible to represent overlapping spans) and it also allows for representing graph structures; this has been suggested in the Linguistic Annotation Format (LAF, [15]) as well as in PAULA, and it is the approach pursued in MOTS, as will be explained in the next section.

Apart from this difference in basic design, MOTS obviously is a much more modest endeavour than GATE. For one thing, at the moment we provide only single-document processing, so there is no corpus management facility. For the time being, MOTS offers neither a systematic description of component behaviour nor a treatment of linguistic non-processing resources (e.g., lexicons), as it has been introduced in version 2 of GATE. More importantly, as will become clear in Section 2.5, a unified access to the analysis results of the components is only now becoming available in the form of a Java API; apart from that, components need to parse the standoff representations and extract the required data themselves.

Another important issue for NLP frameworks is their providing for distributed and/or parallel processing. Regarding distribution, the introduction of service oriented architectures and in particular web services was a very influential development in recent years, and it is beginning to play a more important role in language processing as well. The possibility of flexibly coupling services that do not have to be locally installed can speed up the prototyping of NLP applications significantly (but, on the other hand, it of course may slow down execution speed); notable examples are the web services provided by the project Deutscher Wortschatz at Leipzig University.[3] The MOTS approach of exchanging analysis results via standoff XML annotations makes the integration of external services quite straightforward – they can be wrapped in much the same way as modules running on the local machine.

Non-linear execution, on the other hand, is as of today not a standard feature in NLP frameworks. Various proposals had been made in the 1990s, among them the PVM-based ICE manager [1] in the Verbmobil project. Today, however, the well-known and widely-used frameworks LT-XML2[4] and GATE realize strictly sequential "pipe/filter" architectures. So does MOTS, and at the moment it gives the user only limited means to influence the processing chain: In contrast to, for example, the possibility of dynamically adding components and menu-based construction of processes in GATE, we merely allow for selecting the modules that are to take part in the process from a hard-wired list – the user can thus activate modules, while MOTS ensures the well-formedness of the resulting chain (keeping track of dependencies).

[3] `http://wortschatz.uni-leipzig.de/Webservices/`

[4] `http://www.ltg.ed.ac.uk/software/ltxml2`

Parallel processing *is* enabled in implementations of the Unstructured Information Management Architecture (UIMA)[5]. This is the most ambitious approach, in principle targeting not only text documents but other kinds of unstructured information as well (audio, video), with the basic mission to turn unstructured into structured information (e.g., database tables). Though sharing many similarities with GATE, UIMA is more explicitly oriented to scalable processing of vast amounts of data, hence to the professional construction of language engineering applications. In contrast to GATE, the basic unit of processing in UIMA is a general feature structure rather than an annotated text (which would be just one special kind of feature structure). Also, these structures are strongly typed, requiring very explicit descriptions of components' interfaces. This clearly facilitates sharing of components (a number of repositories already exist) but at the same time is not trivial and requires negotiation between parties interested in using the components, possibly for quite different purposes.

With its much more modest goals, the idea of MOTS is to provide a lean, easy-to-use framework for effective development and testing of modules for single-document processing, in the spirit of rapid prototyping. It is aimed explicitly at research and teaching and not at building applications, so there is more emphasis on simplicity than on performance.

2.3 The PAULA XML Format

The rationale behind our representation format PAULA[6] (a German acronym for 'Potsdam interchange format for linguistic annotation') is the integration of different annotation structures, whether resulting from manual or from automatic annotation. With respect to manual annotation, we provide conversion tools that map the output of the following tools to PAULA: *annotate* for syntax annotation; *Palinka*[7] and *MMAX2*[8] for discourse-level annotations such as co-reference; *EXMARaLDA*[9] for dialogue transcription and various layer-based annotations. The conversion scripts are publicly available via the PAULA webpage: Users can upload their data and annotations, and the data is converted automatically to PAULA. The mappings from the tool outputs to our format are defined such that they only transfer the annotations from one format into another without *interpreting* them or adding any kinds of information.

[5] `http://incubator.apache.org/uima/`

[6] See [7] and `http://www.sfb632.uni-potsdam.de/projects/d1/paula/doc/`

[7] `http://clg.wlv.ac.uk/projects/PALinkA/`

[8] `http://mmax2.sourceforge.net`

[9] `http://exmaralda.org`

2.3.1 PAULA: Logical Structure

The conceptual structure of the PAULA format is represented by the PAULA Object Model (POM). It operates on a labeled directed acyclic graph. Similar to the NITE Object Model [12, NOM] and the GrAF data model [16], nodes correspond to annotated structures, and edges define relationships between independent nodes. Both nodes and edges are labeled, and generally, labels define the specifics of the annotation. Nodes refer to other nodes, or point to a stream of primary data.

Besides labels that define concrete annotation values, a specialized set of labels serves to indicate the *type* of an edge or a node. For a specific set of pre-defined edge labels, POM defines the semantics of the relation expressed by the corresponding edge. For instance, the *dominance* relation is characterized as a transitive, non-reflexive, antisymmetric relation, which requires that the primary data covered by the dominated node is covered by the dominating node as well. On the basis of these dominance relations, tree structures can be represented, e.g. syntactic trees.

Another pre-defined edge type is *reference*, a non-reflexive, antisymmetric relation. Reference relations may occur with different annotation-specific labels. Reference relations with the same label, e.g. 'anaphoric_link', or 'dependency_link' are also transitive. Reference relations serve to express, for instance, dependency trees, coreference relations, or alignment of text spans in multilingual data.

The PAULA Object Model differs from related proposals, e.g. GrAF, in the definition of explicit semantics for certain edge types. The specifications of the dominance relation are comparable to the NITE Object Model, but while NOM focuses on hierarchical annotation, POM also formulates the semantics of pointing relations.

On the basis of this general object model, annotation-specific data models are then defined with reference to POM.

2.3.2 PAULA: Physical Structure

The elements of the PAULA representation format along with their corresponding POM entities are given in Table 2.1. For illustration, Figure 2.1 shows a sample annotation data set, as it is distributed across different layers (and files). The token layer, via xlink, identifies sequences of characters in the primary text files, and thereby provides the ultimate reference objects (in POM, terminal nodes) for other levels of annotation. We call such objects 'markables', and hence the token layer is of type 'mark list'. The POS (part of speech) layer, in contrast, does not define new markables but merely provides labels to existing ones; its type therefore is 'feat list'. The sentence layer, not surprisingly, provides objects that are sequences of tokens; these can then also be given attributes, such as the term relevance values in our example (on the Term layer). Paragraphs are then represented as sequences of sentences in the Div layer (see Section 2.4.1) and can in turn receive attributes, as by the Zone layer in the example (see Section 2.5).

Table 2.1 Predefined structure elements in the PAULA Object Model.

PAULA element	POM entity
`tok`(en)	terminal node
`mark`(able)	non-terminal node (containing *references* to nodes)
`struct`(ure)	non-terminal node (containing *dominance relations* to nodes)
`rel`(ation)	within `struct`: *dominance*, otherwise *reference* relation
`feat`(ure)	annotation label
`multiFeat`(ure)	bundles of annotation labels

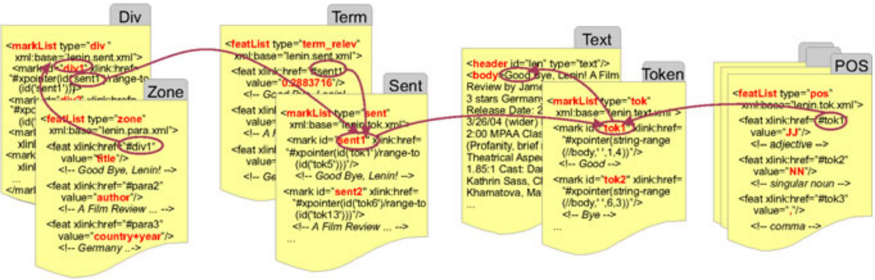


Fig. 2.1 Illustration of PAULA standoff annotation.

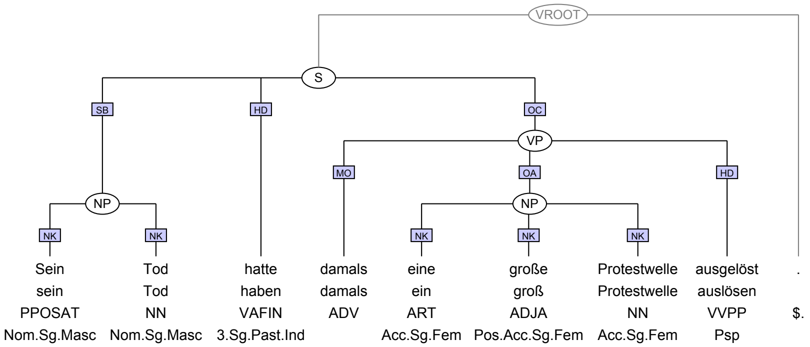


Fig. 2.2 Example of a TIGER tree.

For the encoding of hierarchical structures, including labeled edges, PAULA provides the specific elements `struct` and `rel`. Like markables, a `struct` element represents a node in POM, but in this case a node which is the parent node of a *dominance* relation. The dominance relation is expressed by the `rel` element. An annotation example with hierarchical syntax annotation in the TIGER format is shown in Figure 2.2. A PAULA `struct` element with its daughters corresponds to a local TIGER subtree, i.e. a mother node and its immediate children. For instance, the subtree dominated by the first NP in Figure 2.2, *sein Tod*, 'his death', is represented by a `struct` element that, via `rel` elements, embeds the daughter tokens with IDs `tok_26/27` (these are stored in a separate file called "tiger.ex.tok.xml"). The NP subtree itself is dominated by another `struct` element, with ID `const_14`. `feat` elements encode the categorial status of these subtrees, "NP" and "S" respectively, and their grammatical functions. For example, the `rel` element with ID `rel_39`, which connects the subtree of S with the subtree of the NP, is marked as "SB" relation by the `feat` element pointing to `#rel_39`.

File *tiger.TIG49796.const.xml*:

```
...
<struct id="const_11">
   <rel id="rel_30" type="edge" xlink:href="tiger.ex.tok.xml#tok_26"/>
      <!-- Sein -->
   <rel id="rel_31" type="edge" xlink:href="tiger.ex.tok.xml#tok_27"/>
      <!-- Tod -->
</struct>
<struct id="const_14">
  <rel id="rel_38" type="edge" xlink:href="tiger.TIG49796.tok.xml#tok_28"/>
      <!-- hatte-->
  <rel id="rel_39" type="edge" xlink:href="#const_11"/>
  <rel id="rel_40" type="edge" xlink:href="#const_13"/>
</struct>
...
```

File *tiger.TIG49796.const˙cat.xml*:

```
...
<feat xlink:href="#const_11" value="NP"/>
<feat xlink:href="#const_14" value="S"/>
...
```

File *tiger.TIG49796.const˙func.xml*:

```
...
<feat xlink:href="#rel_30" value="NK"/><!-- Sein -->
<feat xlink:href="#rel_31" value="NK"/><!-- Tod -->
<feat xlink:href="#rel_39" value="SB"/>
...
```

A consequence of the decision to have annotations point – possibly by transitivity – to tokens is that the information cannot be directly read off in the opposite directions, i.e., for a particular token we do not represent explicit links to all its annotations. If needed, this has to be computed by traversing the various annotation layers, which is a functionality provided by our Java API (see Section 2.7).

2.4 Processing Pipeline

We now turn to the description of the MOTS workbench itself, which is realized as a pipeline architecture.

When a text document is submitted, a preprocessing stage first transforms it into the PAULA standoff format, which will be explained below. Then, during the "proper" processing stage it is enriched with further layers by the analysis components (see Section 2.5). Finally, all resulting PAULA layers are being merged into a standard inline XML representation, which is used for visualization purposes (see Section 2.6.1).

The pipeline is implemented as a shell script. It manages the flow of processing, starting from the input document and leading to an output representation that can be shown to the user, while the analysis steps can be flexibly switched on and off. Each component in the processing pipeline constitutes a distinct application and thus can also be executed outside of MOTS. The components (to be described in the next section) were developed in various programming languages; some of them are external off-the-shelf solutions, others were developed in-house by our research group and students.

MOTS offers a set of parameters for configuring the analysis pipeline. One class of parameters, as mentioned above, serves to de-/activate analysis components; violations of dependencies are detected automatically. An important parameter is the *genre* of the text, as several of our components provide genre-specific information tailored to, for example, news articles, film reviews, or court decisions. Other parameters reflect document properties such as the language (German or English) and the technical format of the input (see below). A parameter specific to the summarization task is the desired definition of 'term' (wordform, lemma, Porter stems, character based n-grams), which is then used for calculating sentence relevance. Certain other, more technical, parameters defining the use of meta information and output directories can be used on the command line, but are hidden in the regular user interface (the GUI page for setting these parameters is shown in Figure 2.6, Section 2.6).

A common problem for integrative NLP platforms is character encoding. All our intermediate representations are encoded in UTF-8, but because some of the modules work only with ISO 8859-1, we use only characters compatible to this encoding. In a first step, the input document is converted to ISO 8859-1. This conversion uses some intelligent features, such as mapping Cyrillic letters to their ISO 8859-1 transcriptions. Afterwards we convert the ISO 8859-1 file back to UTF-8. For the modules using ISO 8859-1, we use *gnu iconv* for converting the input from UTF-8 to ISO 8859-1 and the output back from ISO 8859-1 to UTF-8.

The preprocessing, i.e., the first phase of the pipeline, is performed obligatorily for each input document. It consists of three steps: conversion to a normalized format, tokenization, and conversion to PAULA standoff. The result is a PAULA representation of the input document with layout information, tokens and sentence boundaries. Next, we discuss the three steps in turn, and afterwards describe the integration of "real" analysis modules.

2.4.1 Normalized Input Format: LDOC

The system accepts input in various forms: plain text, some XML formats, and HTML. In the first preprocessing step, the input document is converted into our "normalized" XML format LDOC [21], which provides markup for layout features. The conversion identifies headers, paragraphs and highlighted text, and it extracts metadata from XML or HTML headers.

```
<?xml version="1.0" encoding="utf-8"?>
<ldoc id="text.utf8.in">
<body>
 <div id="div_1" type="heading" typeConf="high" >
      29. Dezember 2005
 </div>
 <div id="div_2" type="heading" typeConf="high">
     Bube, Damenopfer, König, As und Sieg
 </div>
 <div id="div_3" type="heading" typeConf="high">
      Match Point
 </div>
 <div id="div_4" type="paragraph" typeConf="medium">
     Ein junger Mann gerät unversehens in die High Society.
     Aufgrund seines gestiegenen Selbstbewusstseins verkalkuliert er sich
     im Privaten und schreckt schließlich vor einem Doppelmord nicht zurück,
     um seine Stellung zu verteidigen.
 </div>
 ...
 <div id="div_22" type="paragraph" typeConf="medium">
     © filmrezension.de
 </div>
</body>
</ldoc>
```

Fig. 2.3 LDOC representation of a film review.

```
<html>, <meta>, <head>, <body>
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>
<p>, <div>, <span>
<a>, <img>, <q>, <abbr>
<table>, <menu>, <ul>, <ol>, <dl>
<th>, <td>, <tr>, <li>, <dt>, <dd>
<i>, <b>, <u>, <strike>, <big>, <small>, <sub>, <sup>, <em>,
<strong>
```

Fig. 2.4 HTML tags considered for LDOC.

The LDOC format is defined and validated with a RELAX NG specification. Figure 2.3 shows an excerpt from a sample LDOC file. The general structure is as follows. A <header> tag encloses meta information, and the <body> tag encloses the document content. The layout is marked by <div> and <span>. <div> stands for *division* and marks the document structure, i.e., headings and paragraphs. <span> is used to mark smaller highlighted units within a <div>. Both tags have attributes. The *type* attribute, for instance, determines whether a <div> is a heading or a paragraph. Other attributes can encode the confidence value assigned by the converter. For instance, if a line in the text contains only one or two words, it is very likely that this line is a heading, while the confidence value for a longer line to be a header is lower.[10]

Our prototypical converter from HTML to LDOC resulted from a students project, where rules for all those HTML tags that are relevant for LDOC were

[10] These confidence values play a role predominantly for converting plain text documents, where quite a bit of guesswork can be involved.

developed; these tags are listed in Figure 2.4. All other tags are ignored. The converter works on XHTML, which is first produced from the HTML input.

2.4.2 Tokenization

The second preprocessing step is tokenization. Our tokenizer accepts an LDOC document as input, determines the character positions of individual tokens, and assigns a label to each token, which gives its type. Some of the types we use are XMLTAG, WORD, PUNCT, DATE, SBOUND, ABBREV, QUOTE, BRACE-OPEN, FLOAT, MIXEDSTRING.

Another task of the tokenizer is sentence boundary detection. Using lists of abbreviations, full stops are identified, disambiguated, and labeled accordingly. For German texts, the presence of upper-case letters at the beginning of sentences is taken into consideration: A determiner starting with an upper-case letter after an abbreviation or ordinal number marks the beginning of a sentence, while other tokens do not.

The tokenizer follows a decidedly "surface-level" approach and does not recognize any multi-word expressions such as proper names. This step is left to a dedicated named-entity recognition component that can be adapted to specific domains, while tokenization is a domain-independent task.

2.4.3 Conversion to PAULA

Tokenized LDOC is the input for the last preprocessing step: the conversion to PAULA standoff. The output is a set of PAULA layers: markables for text, token, sentence, div, span; features for div and span. The conversion distinguishes between XML and text tokens: XML tokens mark the layout, text tokens contain the original text. As indicated above, the PAULA 'text' layer is a sequence of all text tokens, while the 'token' layer records the character positions of each token. Each token is attached to a sentence in the corresponding layer, where headings are also regarded as sentences. Sentence boundaries are determined by full stops as well as closing division boundaries (</div>). Other markable layers are 'span' (referring to tokens) and 'div' (referring to sentences). For each attribute of <div> and <span>, a separate feature layer is produced.

Now the preprocessing is completed and the basic PAULA layers are available for the "real" analysis components in the pipeline, whose job it is to add further layers to the PAULA set.

2.4.4 Integrating Analysis Components

The flexible part of the pipeline can be configured for each execution by selecting the active processing modules. The pipeline script then manages the order of processing and resolves all module dependencies. Input and output of the modules in general are one or more layers of the PAULA standoff set. On the output side, in addition to PAULA layers, most modules also provide a human-readable format, which can be used for debugging purposes.

Any external, off-the-shelf tools that are to be integrated into MOTS need to be wrapped by converters that generate input for the tool from PAULA, and create PAULA layers from the output. The effort needed for building a converter depends on the type of the annotation. Writing a converter for a tool just annotating tokens with features is easily done. For annotations with complex structures, the task becomes more difficult. The efficiency of a converter at runtime depends on the complexity of annotations as well. For example, when running the Tree Tagger[11] and chunker [20], the time needed to convert from and to PAULA is $1/3$ of the overall runtime of the component. For this reason, we skip the conversion steps when components depend directly on another's output.

Another well-known issue with integrating a variety of off-the-shelf components is their tokenization behavior. Many "black box" components read their input as text and tokenize it by their own rules. In such cases, the converter has to align the tokens produced by the component with the "standard" MOTS tokenization performed in preprocessing. Usually, the differences are restricted to the interpretation of spaces, punctuation and sentence boundaries, but some tools also identify multiword tokens (such as *Golden Gate Bridge*) and portmanteau words (such as German *im*, which is evaluated as *i+m = in dem*). In such cases, the features of the additional or reduced tokens are determined by heuristic rules.

The central advantage of using PAULA standoff in MOTS is the flexibility in the architecture and pipeline configuration. Components such as taggers or parsers can be easily replaced, or both run on the same text for evaluation purposes. The results, always in the shape of different annotation layers on the same text, can be visualized quite straightforwardly (see Section 2.6). On the other hand, the substantial XML "packing and unpacking" at runtime comes with a cost that may be prohibitive for "real-time" applications; we will return to this issue at the end of the paper.

2.5 Analysis Components

In this section, we provide an overview of the analysis components that we have integrated into MOTS so far, sorted (roughly) by the level of analysis to which they apply: token, clause/sentence, and discourse.

Token Level

To provide a basic level of analysis for other components to build on, we integrated the Tree Tagger with the off-the-shelf models for English and German, along with its chunking mode. Thus two separate levels of analysis are created, one for part-of-speech tags and one for NP- and PP-chunks. Also operating on the level of tokens,

[11] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

a named entity recognizer combines gazetteer-based lookup of person's first names, location and company names with a set of rules that hypothesize the presence of a last name after a first name or a title, and the like [11]. Similarly, a component in charge of temporal information [17] identifies German time and date expressions (including their manifold linguistic variants such as: *11.25, fünfundzwanzig nach elf, fünf vor halb zwölf, ...*) and assigns corresponding formal representations in the *Temporal Expression Language* [10], which is similar in spirit to TimeML[12].

On the level of (sub-)tokens, we compute weights based on TF/IDF with respect to a genre-specific reference corpus. This is one of the points where the user's choice of genre for the input text plays a role. (If no genre is selected, a "generic" corpus is used.) These weights are later used for computing sentence weights (for summarization/extraction) as well as for isolating 'topic' identifiers in text tiles (see below). As for the unit of analysis, MOTS allows for selecting regular tokens as well as character n-grams. As shown by [2], different definitions are better-suited for different purposes.

Clause and Sentence Level

As a preparatory step for discourse-level analysis of coherence relations, we provide the Brill tagger with a model that we incrementally trained in order to perform disambiguation of certain connectives as to their sentential versus discourse reading. Based on the part-of-speech context, this tagger tries to distinguish the readings of, for example, German *darum* (causal connective versus verbal particle) – as we had reported in [8], off-the-shelf taggers for German often do not provide this information correctly, sometimes because of errors and sometimes because the part-of-speech tag is the same for both the discourse and the sentential reading.

Using the sentence boundaries and the part-of-speech tags, a first version of a subjectivity analysis component employs a rule-based approach to recognizing opinions in text. It was designed for the genre of student's evaluations of their classes and uses lists of subjective vocabulary tailored to this task. After identifying positive/negative lexemes from the lists, it checks for the presence of negations (using a simple fixed-size word window) and if appropriate, applies rules to reverse the polarity of the opinion. On the basis of these results, sentences are labelled as positive, negative, or neutral.

To enable deeper linguistic analyses, we integrated the Connexor syntax parser[13] for German. It delivers dependency structures that are required by the Rosana anaphora resolution (see below), and we also use them as the basis for other discourse-level tasks. In addition, we built a wrapper for the BitPar parser[14], which delivers constituent structures of sentences in the TIGER format.

[12] `http://www.timeml.org`

[13] `http://www.connexor.com`

[14] `http://www.ims.uni-stuttgart.de/tcl/SOFTWARE/BitPar.html`

Discourse Level

The anaphora resolution component *Rosana* was originally developed by [22] for English, and later (in collaboration with our group) also for German. Based on the Connexor dependency structures, Rosana tries to resolve pronouns, proper names and definite descriptions. A significant gap for the coreference analysis (especially for the genre of newspaper texts) is the handling of named entities. We are currently designing an approach to fusing the algorithms employed by Rosana with the results of our named-entity recognizer, working towards an integrated approach that would cover the whole variety of (direct) nominal anaphora.

For statistical text tiling and topic detection, we implemented algorithms by [14] and [24], which determine topic boundaries in the text (either in correspondence with or in ignorance of the paragraphs encoded in our LDOC layer) and also compute keywords that are representative for the particular "tile". For this purpose, we also make use of the aforementioned genre-specific reference corpora.

As an investigation into the 'rhetorical structure' of paragraphs, we developed a component for local coherence relations, which specifically identifies causal relations in German text. On the basis of a declarative lexicon, the presence of a relation is established by analyzing connectives and, if necessary, performing disambiguation (using rules operating on the part-of-speech context). Then, the spans related by the connective are hypothesized on the grounds of the syntactic dependency trees. This works quite well for conjunctions and prepositions; whereas for adverbial connectives, we can only guess that the two neighbouring sentences are in fact the related spans.

Finally, operating on the level of the complete text, our *IDOC* component performs an analysis of the functional role of the individual LDOC portions of the text (i.e., lines and paragraphs), similar to the "argumentative zoning" approach by [23]. The target of this approach are semi-structured documents, which display regularities as to the inventory of labels needed to describe the role of paragraphs and the linear order of the elements. Also, some of the zones need to be recognizable on the basis of surface features (words, length of paragraph, etc.). Our approach performs two steps: In the first phase, regular surface patterns are matched using LAPIS [18], thus identifying a certain number of zones reliably. In the second phase, the zones already found are used to hypothesize the presence of other zones in the remaining material, using likelihoods derived from surface features and from the neighbourhood of zones already found. So far, we implemented the approach for two genres: film reviews [3] and court decisions.

2.6 User Interface

When the execution of the pipeline script is complete, and all analyses are accomplished, the PAULA standoff set contains a lot of XML files that the average user would not want to inspect. Hence, the MOTS user interface is a convenient way to evaluate and compare all the results of the analysis pipeline.

```
<div _id="id_1893" _org_id="div_2" type="heading" typeConf="high"
zone="tagline|title">
 <sent _id="id_1101" _org_id="s_2" W5g="0.028237089618056307" Wxdoc="0.64">
  <tok _org_id="tok_4" _id="id_4" pos="NN" lemma="Bube">Bube</tok>
  <tok _org_id="tok_5" _id="id_5" pos="$," lemma=",">,</tok>
  <tok _org_id="tok_6" _id="id_6" pos="NN"
  lemma="Damenopfer">Damenopfer</tok>
  <tok _org_id="tok_7" _id="id_7" pos="$," lemma=",">,</tok>
  <tok _org_id="tok_8" _id="id_8" pos="NN" lemma="Koenig">Koenig</tok>
  <tok _org_id="tok_9" _id="id_9" pos="$," lemma=",">,</tok>
  <tok _org_id="tok_10" _id="id_10" pos="NN" lemma="As">As</tok>
  <tok _org_id="tok_11" _id="id_11" pos="KON" lemma="und">und</tok>
  <tok _org_id="tok_12" _id="id_12" pos="NN" lemma="Sieg">Sieg</tok>
 </sent>
</div>
```

Fig. 2.5 Inline representation.

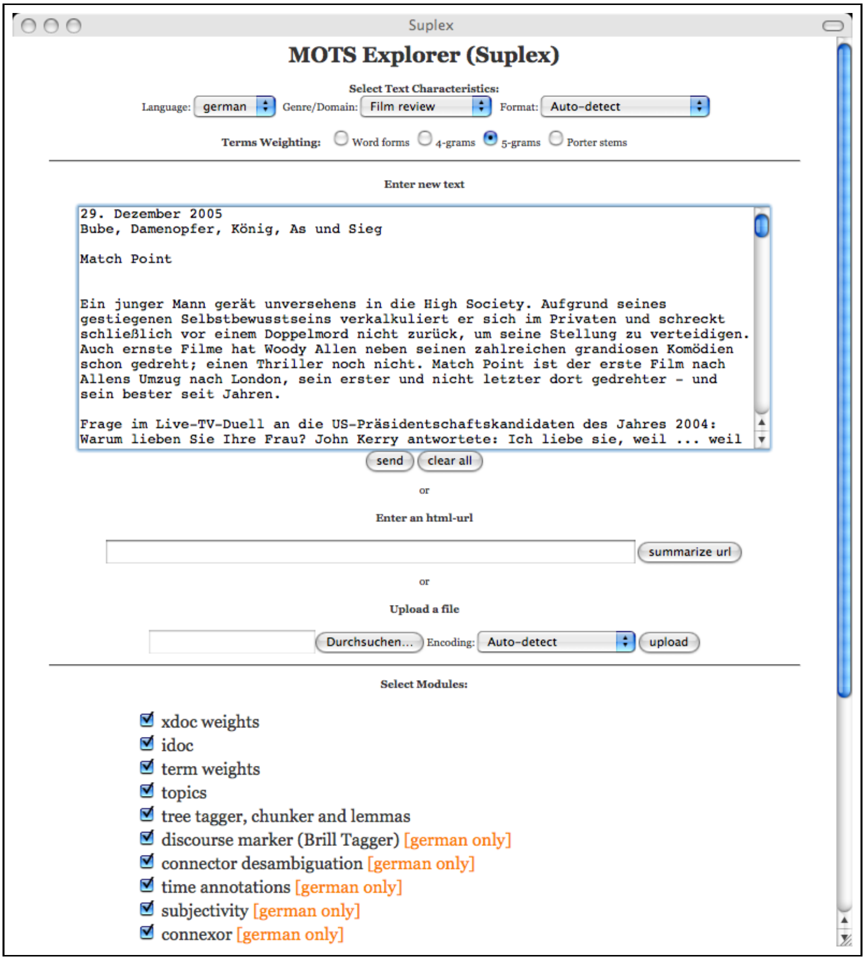
2.6.1 XML Inline Representation

The input to the visualization component is a standard inline-XML file whose format is called ‘PAULA-inline’. It results from merging all the PAULA layers into a single file – a step needed solely to facilitate the graphical presentation of the output. The inline-XML document contains an XML element for each markable (token or span). All features referring to this markable are annotated as attributes of this element. Smaller spans of markables are added as children of wider spans.

Figure 2.5 gives an extract of the inline representation of the beginning of a text we had shown earlier (Figure 2.3). The example shows a headline, which is also annotated as sentence. The tokens contained therein are represented as children of the sentence element. Tokens are annotated with *pos* and *lemma*. PAULA markables do not have any dominance relation between each other; they just mark spans in the text. If two markables cover exactly the same span of text, in the inline document one is arbitrarily chosen to embed the other. Thus in the example, the division element could also be included into the sentence element.

To mark “real” embedding in structures such as trees (PAULA <struct> elements), we use the special element <_rel>, which explicitly encodes the dominance relation in PAULA-inline. This allows us to annotate edges between nodes.

XML embedding cannot be used for the representation of overlapping segments. For such data, we use the strategy of fragmentation: One of the overlapping elements is broken into smaller units and an attribute *gid* (‘group id’) is added to the fragmented elements to explicitly mark elements that belong together. For further details on creating the PAULA-inline representation, see [9].



MOTS Explorer (Suplex)

Select Text Characteristics:

Language: german   Genre/Domain: Film review   Format: Auto-detect

Terms Weighting: ○ Word forms ○ 4-grams ● 5-grams ○ Porter stems

Enter new text

```
29. Dezember 2005
Bube, Damenopfer, König, As und Sieg

Match Point


Ein junger Mann gerät unversehens in die High Society. Aufgrund seines
gestiegenen Selbstbewusstseins verkalkuliert er sich im Privaten und schreckt
schließlich vor einem Doppelmord nicht zurück, um seine Stellung zu verteidigen.
Auch ernste Filme hat Woody Allen neben seinen zahlreichen grandiosen Komödien
schon gedreht; einen Thriller noch nicht. Match Point ist der erste Film nach
Allens Umzug nach London, sein erster und nicht letzter dort gedrehter - und
sein bester seit Jahren.

Frage im Live-TV-Duell an die US-Präsidentschaftskandidaten des Jahres 2004:
Warum lieben Sie Ihre Frau? John Kerry antwortete: Ich liebe sie, weil ... weil
```

Enter an html-url

Upload a file

Encoding: Auto-detect

Select Modules:

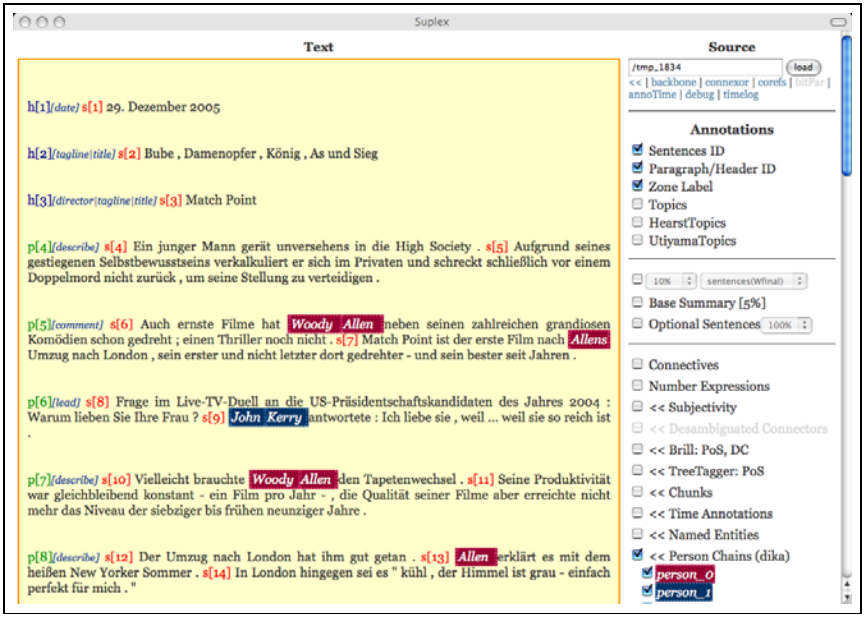
☑ xdoc weights
☑ idoc
☑ term weights
☑ topics
☑ tree tagger, chunker and lemmas
☑ discourse marker (Brill Tagger) [german only]
☑ connector desambiguation [german only]
☑ time annotations [german only]
☑ subjectivity [german only]
☑ connexor [german only]

Fig. 2.6 Graphical user interface: parameter selection.

2.6.2 *Visualization*

The user interface is a PHP script accessible with a web browser – see Figure 2.6. It is to be used in two steps: First, the user defines the input and the processing parameters. The input can be directly entered as text, uploaded as a file, or defined by a URL. Then, the pipeline script is started with the 'send' button. When processing is complete, the resulting PAULA-inline document is converted by an XSLT script to HTML and thus available for viewing with PHP and Javascript.

In the second step, the user can browse the results and compare various analyses (Figure 2.7). All annotations on a specific token are shown on a mouse-over. Other annotations on larger spans are highlighted upon request. Some component-specific


Suplex
Text
Source
load
<< | backbone | connexor | corefs | bitPar |
annoTime | debug | timelog

h[1]*[date]* s[1] 29. Dezember 2005

h[2]*[tagline|title]* s[2] Bube , Damenopfer , König , As und Sieg

h[3]*[director|tagline|title]* s[3] Match Point

p[4]*[describe]* s[4] Ein junger Mann gerät unversehens in die High Society . s[5] Aufgrund seines gestiegenen Selbstbewusstseins verkalkuliert er sich im Privaten und schreckt schließlich vor einem Doppelmord nicht zurück , um seine Stellung zu verteidigen .

p[5]*[comment]* s[6] Auch ernste Filme hat *Woody Allen* neben seinen zahlreichen grandiosen Komödien schon gedreht ; einen Thriller noch nicht . s[7] Match Point ist der erste Film nach *Allens* Umzug nach London , sein erster und nicht letzter dort gedrehter - und sein bester seit Jahren .

p[6]*[lead]* s[8] Frage im Live-TV-Duell an die US-Präsidentschaftskandidaten des Jahres 2004 : Warum lieben Sie Ihre Frau ? s[9] *John Kerry* antwortete : Ich liebe sie , weil ... weil sie so reich ist .

p[7]*[describe]* s[10] Vielleicht brauchte *Woody Allen* den Tapetenwechsel . s[11] Seine Produktivität war gleichbleibend konstant - ein Film pro Jahr - , die Qualität seiner Filme aber erreichte nicht mehr das Niveau der siebziger bis frühen neunziger Jahre .

p[8]*[describe]* s[12] Der Umzug nach London hat ihm gut getan . s[13] *Allen* erklärt es mit dem heißen New Yorker Sommer . s[14] In London hingegen sei es " kühl , der Himmel ist grau - einfach perfekt für mich . "

/tmp_1834

Annotations

☑ Sentences ID
☑ Paragraph/Header ID
☑ Zone Label
□ Topics
□ HearstTopics
□ UtiyamaTopics

□ 10% sentences(Wfinal)
□ Base Summary [5%]
□ Optional Sentences 100%

□ Connectives
□ Number Expressions
□ << Subjectivity
□ << Desambiguated Connectors
□ << Brill: PoS, DC
□ << TreeTagger: PoS
□ << Chunks
□ << Time Annotations
□ << Named Entities
☑ << Person Chains (dika)
☑ *person_0*
☑ *person_1*

Fig. 2.7 Graphical user interface: browsing results.

visualization is being realized. Sentence and paragraph boundaries and headings are marked by a tag at the beginning of the text range. Tile boundaries are marked more prominently and are completed by keywords describing the topic. Most annotations assign labels to ranges of text (mostly tokens). The labels appear at the right frame of the page ordered by the kind of annotation and can be selected for colored highlighting. In addition to the merged graphical presentation, the output of each component is accessible in its original form by following hyperlinks; likewise we provide debugging and runtime information on separate pages. Also, there is a button for downloading the PAULA files that have been produced, so that they can be used as input for other software.

2.7 Current Developments and Conclusion

The file-centric approach of MOTS, managing interfaces through a fairly generic standoff annotation format, resulted on the one hand from the dual needs of (i) bridging between manual annotation tools and (ii) automatic processing. At the same time, it provides a simple way of allowing for interoperability of components that enables rapid prototyping, which has been utilized in a number of students' projects and diploma theses. One advantage here is the independence on programming languages: Existing modules in script languages or more traditional languages (e.g., Lisp) can be added quite easily. In this way, MOTS enables quick experiments with combining modules for new tasks, without right away taking the step to work with a more complex system such as GATE or UIMA.

In this spirit of a lean, "low-cost" framework, we are currently making several additions to MOTS, whose primary goal is to somewhat reduce the effort of XML processing in a pipeline of components. A Java API is being developed that will be in charge of parsing PAULA files and providing convenient access to the data via the PAULA Object Model (cf. Section 2.3.1). One perspective then is to replace the shell script managing the processing pipeline with a Java application, so that PAULA processing can be restricted to the interfaces of non-Java components.

For the time being, we will hold on to the menu-based de-/activation of components when the pipeline is started. For most purposes of text processing, it has proven sufficient, and moving to a fully-flexible component management system would amount a leap toward GATE-style frameworks, which we are not intending. However, a slightly more generic way of describing component behavior (using simple configuration files) could be added, so that integrating new modules can be done more systematically than by changing the shell script. Likewise, it should be possible to integrate annotations that have been produced manually with a suitable tool. This can be interesting when a higher-level component builds on the output of a lower-level one and for testing purposes, "perfect results" can be provided in place of the lower-level component's output.

Our approach to visualising results currently creates problems with long texts holding many annotations, as Javascript execution becomes prohibitively expensive. While this can be attended to with limited effort, a more substantial problem is the conversion from (possibly many) standoff files to inline-XML, which our GUI is currently based on. This merging step is an inherently complex task, and rather than trying to improve our current merging solution, it is probably more effective to try to circumvent it altogether and base the visualisation on the standoff files. This would amount to a general overhaul of the output side of the GUI, offering the opportunity to establish a more generic solution that maps types of annotations to their visual counterparts (both menu items for clicking annotations on/off and the annotation visualization itself).

Our PAULA tools are being made available via the webpage mentioned in Section 2.3, and in conjunction with our ANNIS linguistic database (focusing on the scenario of manual annotation) [4]. Likewise, the MOTS software is available to interested parties for research or teaching.

Acknowledgements

Part of the work reported in this paper was funded by Bundesministerium für Bildung und Forschung (BMBF), grant 03WKH22.

The following researchers (listed in alphabetical order) have contributed to the development of the MOTS workbench and the PAULA format and tools: Christian Chiarcos, Stefanie Dipper, Michael Götze, Peter Kolb, Uwe Küssner, Julia Ritz, Johannes Schröder, Arthit Suryiawongkul. Also, many of our Computational Linguistics students helped building conversion tools or analysis components.

We are grateful to two anonymous reviewers for their helpful comments on an earlier version of this paper.

References

[1] Amtrup, J.: Ice - intarc communication environment user guide and reference manual version 1.4. Tech. rep. Universität Hamburg (1995)

[2] Bieler, H., Dipper, S.: Measures for term and sentence relevances: an evaluation for german. In: Proceedings of the 6th LREC Conference, Marrakech (2008)

[3] Bieler, H., Dipper, S., Stede, M.: Identifying formal and functional zones in film reviews. In: Proceedings of the Eighth SIGDIAL Workshop, Antwerp (2007)

[4] Chiarcos, C., Dipper, S., Götze, M., Ritz, J., Stede, M.: A flexible framework for integrating annotations from different tools and tagsets. In: Proc. of the First International Conference on Global Interoperability for Language Resources, Hongkong (2008)

[5] Cunningham, H.: Software architecture for language engineering. PhD thesis, University of Sheffield (2000)

[6] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002)

[7] Dipper, S.: XML-based stand-off representation and exploitation of multi-level linguistic annotation. In: Eckstein, R., Tolksdorf, R. (eds.) Proceedings of Berliner XML Tage, pp. 39–50 (2005)

[8] Dipper, S., Stede, M.: Disambiguating potential connectives. In: Butt, M. (ed.) Proceedings of KONVENS 2006, Konstanz, pp. 167–173 (2006)

[9] Dipper, S., Götze, M., Küssner, U., Stede, M.: Representing and querying standoff XML. In: Proceedings of the Biennial GLDV Conference 2007. Data Structures for Linguistic Resources and Applications, Narr, Tübingen (2007)

[10] Endriss, U., Küssner, U., Stede, M.: Repräsentation zeitlicher Ausdrücke: Die Temporal Expression Language. Verbmobil Memo 133, Technical University Berlin, Department of Computer Science (1998)

[11] Ernst, C.: Auffinden von Named Entities in Nachrichtentexten. Diplomarbeit, Institut für Linguistik, Universität Potsdam (2008)

[12] Evert, S., Carletta, J., O'Donnell, T., Kilgour, J., Vögele, A., Voormann, H.: The nite object model. version 2.1. Tech. rep., University of Edinburgh, Language Technology Group (2003)

[13] Grishman, R.: Tipster architecture design document version 2.3. Tech. rep., DARPA (1997),
http://www.itl.nist.gov/div894/894.02/related_projects/tipster/

[14] Hearst, M.A.: Multi-paragraph segmentation of expository text. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Las Cruces/NM, pp. 9–16 (1994)

[15] Ide, N., Romary, L.: International standard for a linguistic annotation framework. Natural Language Engineering 10(3-4), 211–225 (2004)

[16] Ide, N., Suderman, K.: Graf: A graph-based format for linguistic annotation. In: Proceedings of The Linguistic Annotation Workshop (LAW), Prague (2007)

[17] Luft, A.: Automatisches Tagging von zeitlichen Ausdrücken. Diplomarbeit, Institut für Informatik, FH Mittweida (2006)

[18] Miller, R.C.: Lightweight structure in text. PhD thesis, Carnegie Mellon University (2002)

[19] Schäfer, U.: Integrating deep and shallow natural language processing components - representations and hybrid architectures. PhD thesis, Universität des Saarlandes (2007)

[20] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester, pp. 44–49 (1994)

[21] Stede, M., Suriyawongkul, A.: Identifying logical structure and content structure in loosely-structured documents. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages - Contributions to Language Technology, pp. 81–96. Springer, Dordrecht (2010)

[22] Stuckardt, R.: Design and enhanced evaluation of a robust anaphor resolution algorithm. Computational Linguistics 27(4), 479–506 (2001)

[23] Teufel, S., Moens, M.: Summarizing scientific articles – experiments with relevance and rhetorical status. Computational Linguistics 28(4), 409–445 (2002)

[24] Utiyama, M., Isahara, H.: A statistical model for domain-independent text segmentation. In: Proceedings of the ACL/EACL Conference, Toulouse (2001)

Chapter 3
Processing Text-Technological Resources in Discourse Parsing

Henning Lobin, Harald Lüngen, Mirco Hilbert, and Maja Bärenfänger

Abstract. Discourse parsing of complex text types such as scientific research articles requires the analysis of an input document on linguistic and structural levels that go beyond traditionally employed lexical discourse markers. This chapter describes a text-technological approach to discourse parsing. Discourse parsing with the aim of providing a discourse structure is seen as the addition of a new annotation layer for input documents marked up on several linguistic annotation levels. The discourse parser generates discourse structures according to the Rhetorical Structure Theory. An overview of the knowledge sources and components for parsing scientific journal articles is given. The parser's core consists of cascaded applications of the GAP, a *Generic Annotation Parser*. Details of the chart parsing algorithm are provided, as well as a short evaluation in terms of comparisons with reference annotations from our corpus and with recently developed systems with a similar task.

3.1 Introduction

Relational discourse theories like RST [*Rhetorical Structure Theory*, [29], [31]], D-LTAG [*Lexicalized Tree-Adjoining Grammar for Discourse*, [43]], ULDM [*Unified Linguistic Discourse Model*, [33], [34]], or SDRT [*Segmented Discourse Representation Theory* [2], [3]] provide text-type independent principles for analysing coherence relations between parts of a text of different sizes. For some of these theories, discourse parsers have been implemented, notably for RST. Two features of RST make it especially favourable for an automatisation of discourse analyses: RST utilises trees (not graphs like SDRT) as a data structure for discourse representation.

Henning Lobin · Harald Lüngen · Mirco Hilbert · Maja Bärenfänger
Applied and Computational Linguistics, Justus-Liebig-Universität Gießen,
Otto-Behaghel-Straße 10D, D-35394 Gießen, Germany
e-mail: {henning.lobin,harald.luengen,mirco.hilbert,
maja.baerenfaenger}@germanistik.uni-giessen.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 35–58.
springerlink.com

(It is in fact controversial whether graph-based representations are actually necessary for the representation of discourse structures [cf. e.g. [12].) And while in the definition of rhetorical relations in the original theory, references to the beliefs and intentions of speakers and hearers abound [cf. [29]), in the different approaches to RST-based discourse parsing it has been shown that automatic discourse analysis can also be achieved by applying mainly surface-oriented discourse markers (*cues*) [cf. [31], [36], [21]]. In the following, we give a brief overview of previous RST approaches to discourse parsing.

[30, 31] presented several alternative algorithms for the RST parsing of unrestricted texts. One prerequisite for rhetorical parsing formulated by Marcu is the *compositionality principle* for RST structures, which states that a rhetorical relation holding between two text constituents (*spans*) also exists between their respective most salient subconstituents. According to Marcu, another prerequisite for the parsing of unrestricted texts is a feature-based description of discourse markers based on an extensive corpus analysis. Discourse markers are utilized both for the segmentation and identification of related discourse units and for the assignment of a particular rhetorical relation.

Corston-Oliver's [8] automatic Rhetorical Structure Analyser RASTA bases its rhetorical analyses on fully-fledged syntactic analyses of the sentences of a text (in this case articles from the Microsoft Encarta Encyclopedia). Thus the cues that indicate rhetorical relations comprise discourse connectives as well as syntactic and morphological features. A further novelty introduced in this approach is the association of *cue*:*relation* pairs with weights that are based on linguistic intuition. They are used to build up more plausible discourse representations before less plausible ones.

An extension of Corston-Oliver's algorithm is the symbolic RST parser for English developed by Le Thanh [20, 21, 23, 22]. It performs an automatic discourse segmentation into elementary discourse units, sentence-level discourse parsing using syntactic information and cue phrases, and finally, text-level discourse parsing using a beam-search algorithm. To reduce the search space, heuristic scores are used as constraints on textual adjacency and textual organisation. The parser was evaluated on a test corpus from the *RST Discourse Treebank* [7].

As an alternative approach [35, 36] implemented discourse parsing according to RST as a quantitative approach as a series of text classification decisions. Classification instances from a training corpus are represented as feature vectors and associated with rhetorical relation schemata. The linguistic features used include the occurrence of discourse markers in certain segment positions, concepts introduced by definite noun phrases, punctuation, POS tagging and lexical similarity [35]. Support vector machines (SVM) are used as a classification algorithm. As a representation format for RST trees, the XML application URML [*Underspecified Rhetorical Markup Language,* cf. [37] is chosen. In an URML document, alternative tree structures can be presented as well. Besides, URML is used to represent partial results in the parsing process, similar to a chart in chart parsing.

URML is also used in the approach by [14], employing a feature-based RST grammar with a rule hierarchy. The grammar also includes robust rules that

Table 3.1 Annotations in the SemDok corpus.

XML annotation layer	# annotated articles
Logical document structure (DOC)	47+2
Morphological and syntactic structure (CNX) (using the tagger *Machinese Syntax* from Connexor Oy)	47 +2
Discourse markers (DMS)	47
Rhetorical structure (RST-HP)	5+2
Discourse segments (SEG)	5+2
Anaphoric structure (CHS) (from Sekimo project)	3+2
Lexical chains (LC) (from HyTex project)	1+2
Genre-specific text type structure (TTS)	47

combine subtrees when no discourse marker is found. Using this grammar, a standard chart parsing algorithm is applied for discourse parsing. The chart is extended to accommodate parse forests, and URML is used for their representation.

In several of the above described projects, collections of newspaper articles were used as test corpora [36, 38, 19]. The goal of the SemDok project was to design and implement a new RST parser for the text type[1] of (German) *scientific journal articles* as a text-technological application.[2] Scientific articles form a more complex text type than newspaper articles – primarily due to their deeply nested logical document structure. A discourse parser therefore has to resolve a higher number of potential relational combinations of text segments. Besides the traditional discourse markers such as lexical cues, grammatical features and punctuation, features derived from analyses of text and document structures need to be included as well. An overview of the requirements for such an approach in terms of linguistic foundations, resources, and an application scenario is given in [27]. The linguistic resources are made available for the SemDok parser using text-technological (XML-based) standards, formalisms, methods and tools and are described in the following sections.

3.2 Corpus

A corpus of German linguistic journal articles, which was created between 2002 and 2008, served as a development corpus. It provides XML annotations on various linguistic and text-structural analysis layers. The corpus contains 47 German articles of the online journal *Linguistik Online* (`www.linguistik-online.de`) from between 2000 and 2003 (comprising approx. 360,000 word forms). One newspaper article (from the German weekly *Die Zeit*) and one web-published article on

[1] The term *text type* is used as an equivalent for *genre*.

[2] SemDok was a project within the DFG research group 437 *Text-technological modelling of information*. The discourse parser was developed in the project's second funding phase (2005-2008) called *Generic document structures in linearly organised texts*.

hypertext were added (from the corpora compiled in the projects *Sekimo*[3] and *HyTex*[4], cf. “+2” in Table 3.1).

An overview of the SemDok corpus and its different annotation layers is given in Table 3.1. The different annotation layers were added according to the framework of *XML-based multi-layer annotation* [44]: Each annotation layer is stored in a separate XML document, i.e. the primary (text) data are copied several times. Since in all annotation layers, the primary data are absolutely identical, relations holding between elements on different annotation layers can be analysed using the so-called Sekimo tools[5] [45].

The corpus was semi-automatically annotated according to a modified DocBook format (“DOC”, [cf. 41, 24]). For the annotation of morphology and syntax (“CNX”) we employed the commercial software *Machinese Syntax* from Connexor Oy. Machinese Syntax yields dependency trees according to the *Functional Dependency Grammar* [FDG, 39] as an XML-like annotation. For an annotation of lexical discourse markers (“DMS”), a tagger was developed which basically performs lexical insertions according to the SemDok discourse marker lexicon in combination with some context checking [cf. 27]. The initial discourse segmentation (“SEG”) was achieved by a segmentation program also developed in SemDok, which segments a document according to criteria on punctuation, grammar, and logical document structure [cf. 25]. Anaphoric structure (or referential structure) represented by the annotation layer “CHS”, was added to the SemDok documents corpus in the

```
<para xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="hypo-para.xsd" relname="Contrast" id="i12">
  <n id="i13">
    <hypo id="i14" relname="Elaboration-example">
      <n id="i1">
        <t id="ti1">In der Schrift hat die Sprachpflege einen etwas besseren
         Erfolg als im Gespräch gehabt.</t>
      </n>
      <s id="i2">
        <t id="ti2">In öffentlichen Dokumenten ist man z.B. darauf bedacht,
         dass die Termini dem Gebrauch in Schweden entsprechen.</t>
      </s>
    </hypo>
  </n>
  <n id="i4">
    <t id="ti4">Trotzdem enthalten sowohl Sachtexte als auch die
         Belletristik
     sprachliche Züge, die den Schweden fremd vorkommen.</t>
  </n>
</para>
```

Listing 3.1 RST analysis in RST-HP.

[3] http://www.text-technology.de/Sekimo/

[4] http://www.hytex.info

[5] The Sekimo tools operate on a stand-off Prolog fact base format and include tools for merging different annotation layers, for transforming between XML and the Prolog format and for checking relations holding between elements of single annotations layers.

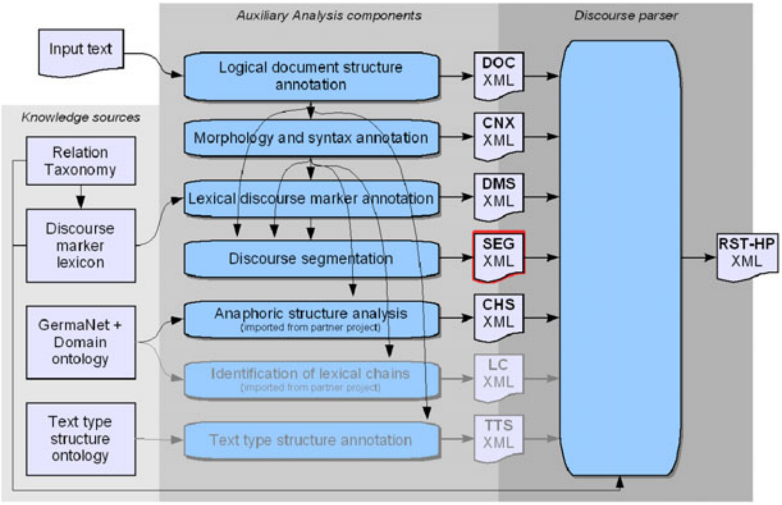

Auxiliary Analysis components
Discourse parser
Input text
Knowledge sources
Relation Taxonomy
Discourse marker lexicon
GermaNet + Domain ontology
Text type structure ontology
Logical document structure annotation
Morphology and syntax annotation
Lexical discourse marker annotation
Discourse segmentation
Anaphoric structure analysis (imported from partner project)
Identification of lexical chains (imported from partner project)
Text type structure annotation
DOC XML
CNX XML
DMS XML
SEG XML
CHS XML
LC XML
TTS XML
RST-HP XML

Fig. 3.1 Discourse Parsing Architecture.

partner project Sekimo [cf. 10]. The SemDok parser is also projected to handle two further annotation layers: an annotation of lexical chains ("LC"), as provided by the partner projects HyTex [cf. 9] and IndoGram [cf. 42], and an annotation of the genre-specific text type structure ("TTS"), available from the first funding period of SemDok [cf. 4].

Apart from the above annotation layers, which serve as auxiliary annotations in the discourse parsing process, several corpus articles were also provided with reference annotations according to the Rhetorical Structure Theory in RST-HP. RST-HP is an XML application developed in the SemDok project to represent RST trees in XML [27, 17]. "HP" stands for `<hypo>` and `<para>`, which are two element types to label RST subtrees as either hypotactic or paratactic (relations). An example of an RST analysis in RST-HP is given in Listing 3.1. RST-HP is also the target format of the SemDok discourse parser.

3.3 Architecture

For the technical realisation of our discourse parsing approach we chose a pipeline architecture, in which linguistic analyses of one text document at different linguistic levels are provided by auxiliary analysis components (preprocessors) implemented as part of the SemDok project, or by project-external software. These components were also used in creating the SemDok corpus (Section 3.2).

The pipeline architecture is shown in Figure 3.1. The auxiliary analysis components are shown in the middle part, on the left-hand side there are four knowledge sources that are used by some of these preprocessors. In this chapter, we do not discuss these knowledge sources, but see [6] for information on the *Relation Taxonomy RRSet*, [27] on the discourse marker lexicon, and [26] for techniques of combining GermaNet with a domain ontology.

As well as in the corpus, the results of the auxiliary analyses are represented as XML annotations according to the principles of *XML-based multi-layered annotation* [44]. To evaluate them, the SemDok parser was provided with an interface to the Sekimo tools [45]. Using the Sekimo tools, configurations of elements and attributes on the different XML annotation layers are tested in the condition parts of reduce rules (see Section 3.4.3.2). Finally, the parser puts out an XML document in RST-HP format.

3.4 Parser

The discourse parser is realised as a cascade, where the input document is processed in bottom-up fashion along its document levels (see Figure 3.2). The information about document levels is provided by the annotation layer SEG, which contains the initial segmentation into elementary and complex discourse segments [cf. 25]. SEG is the only obligatory annotation layer, i.e. all other layers are optional in terms of input requirements.

In each cascade step, a parsing iteration is activated for the associated document level. At the first level, called “EDS+”, for each containing element of type “SDS” (sentential discourse segment), “EDSes”, (elementary discourse segments – mostly clauses) are recursively combined to form structures on SDS level. At the second level, called “SDS+”, SDSes are combined to form complex discourse segments on block level (mostly paragraphs, “CDS$_{block}$”), on the third level, the block-level segments are combined to form CDSes of type “division” (sections and the like, “CDS$_{div}$”), and finally, the CDS$_{div}$ are combined up to the level of the complete document (i.e. one CDS$_{doc}$). In each cascade step, the respective higher level elements are called the *containing elements*, they thus act as top-down constraints in the otherwise bottom-up parsing process.

3.4.1 Initialisation

The chart parser’s internal representation “HPX” [cf. 17] is an extended version of the target format RST-HP that can accommodate underspecified information, cf. Section 3.4.3. During the parsing process, the HPX chart is gradually augmented by new information in the form of new chart edges. Hence, in an initialisation phase, the elementary discourse segmentation SEG is converted to HPX by changing each EDS into a terminal edge element `<t>` and assigning it an underspecified nuclearity in the form of an `<undefined>` element [cf. 31, 152ff]. That way, a sequence of adjacent `<undefined>` elements forms the basis of the parsing process.

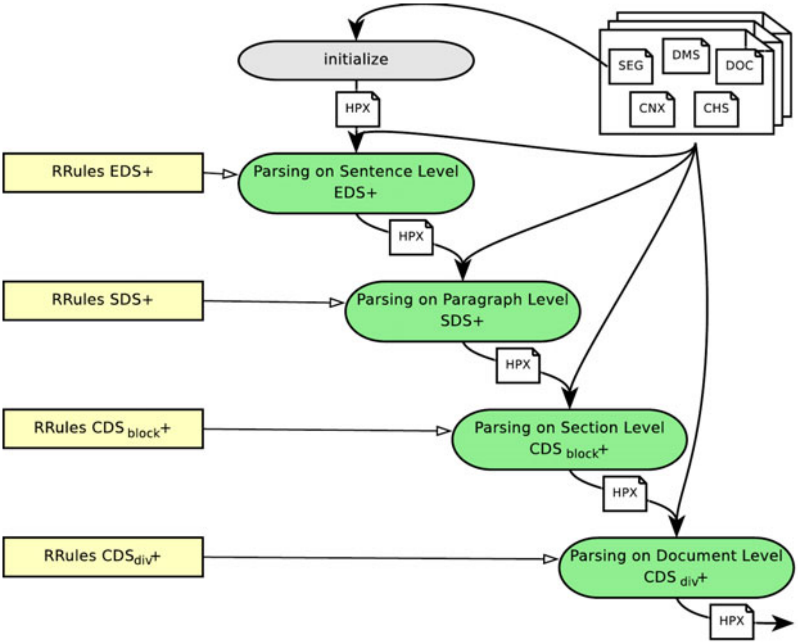

initialize
HPX
SEG
DMS
DOC
CNX
CHS
RRules EDS+
Parsing on Sentence Level EDS+
HPX
RRules SDS+
Parsing on Paragraph Level SDS+
HPX
RRules CDS block+
Parsing on Section Level CDS block+
HPX
RRules CDSdiv+
Parsing on Document Level CDS div+
HPX

Fig. 3.2 Parsing Cascade.

The actual parsing is achieved by consecutive calls of the central parsing component *GAP* (*Generalized Annotation Parser*) for each containing element on the SEG layer. The GAP is described in Section 3.4.3.

3.4.2 Cascade Step

An example of the operation of the algorithm in one cascade step is illustrated in Figure 3.3. The invocation of a cascade step depends on several parameters.

- The *Base Layer Type* and the *Containing Layer Type* specify the document level to be parsed. On the level of "EDS+", for example, RST structures are built whose leaf nodes correspond to segments of the base layer type "EDS" and whose root node completely covers the associated segment of the containing layer type "SDS".
- The *Reduce Rule Set* is the third mandatory parameter. It contains the rules stating how a set of adjacent segments may be linked by a rhetorical relation and combined to form a bigger segment.

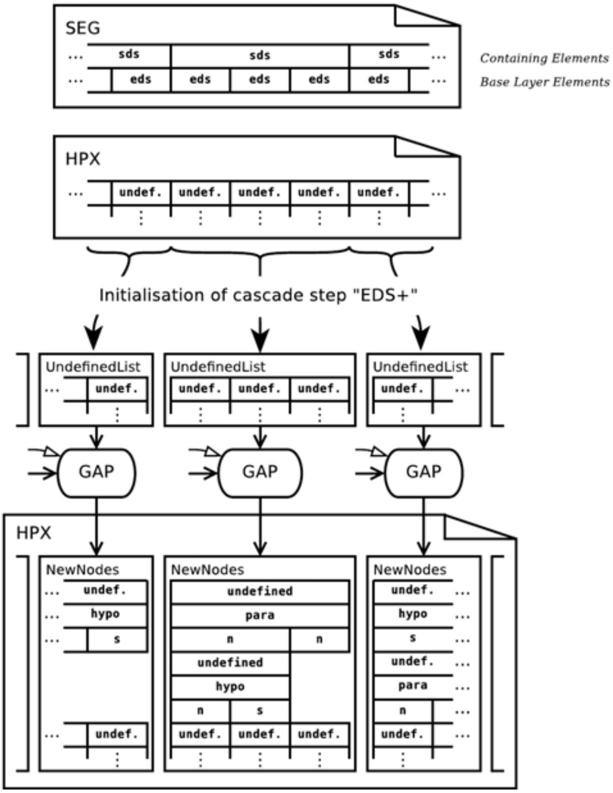


Fig. 3.3 Initialisation and execution of one cascade step by the example of the document level "EDS+" (*Sentence Level*).

Two further parameters modify and extend the effect of the reduce rules: the *Rhetorical Relation Set* and the *Default Relation*.

- The SemDok *Rhetorical Relation Set* (RRSet) forms a taxonomy of rhetorical relations suitable for our application purpose [cf. 6]. A full set consisting of 44 relations and a reduced set consisting of 30 relations with less specific relations at the leaves of the taxonomy are available. Depending on the RRSet variant selected for a cascade step, the reduce rules are customised to label a combined segment with a more specific (e.g. ELABORATION-CONTINUATION-OTHER) or a more general relation (e.g. ELABORATION).
- In the *Default Relation* parameter, a relation is specified that combines two adjacent segments when no regular reduce rule matches the configuration and its features. As values, the three relations most frequently found in the SemDok corpus can be specified: the general multi-nuclear coordination relation LIST-COORDINATION, and two multi-nuclear relations indicating thematic progression, ELABORATION-DRIFT and ELABORATION-CONTINUATION-OTHER.

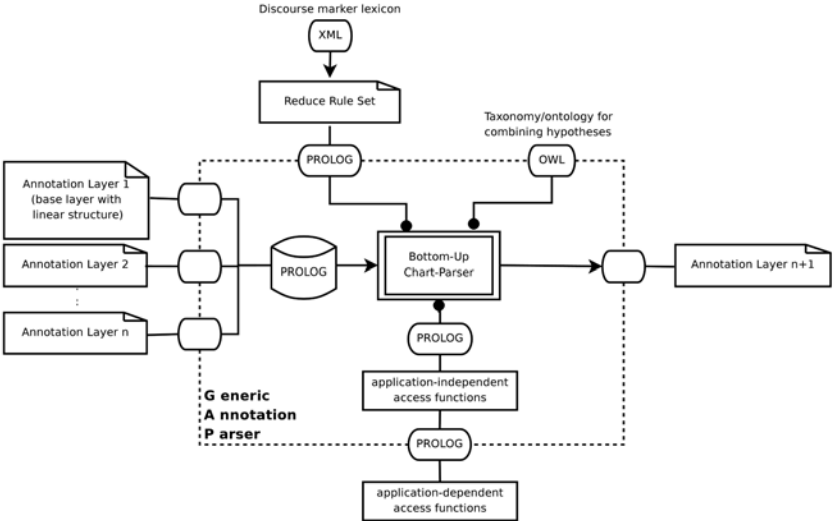

Discourse marker lexicon
XML
Reduce Rule Set
Taxonomy/ontology for combining hypotheses
PROLOG
OWL
Annotation Layer 1 (base layer with linear structure)
Annotation Layer 2
Annotation Layer n
PROLOG
Bottom-Up Chart-Parser
Annotation Layer n+1
PROLOG
application-independent access functions
G eneric
A nnotation
P arser
PROLOG
application-dependent access functions

Fig. 3.4 GAP.

Within a cascade step, a list of `<undefined>` elements is generated first for each segment of the *Containing Layer Type* (*Containing Element*). Each `<undefined>` element corresponds to an element of the *Base Layer Type* (*Base Element*) (for an example cf. the initialisation of cascade step "EDS+" in Figure 3.3).

Iterating over all *Containing Elements*, the GAP is then invoked for each list with multiple `<undefined>` elements.[6]

The GAP attempts to build a partial HPX chart which spans the current *Containing Element*, applying the reduce rules, which make reference to information available on the auxiliary annotation layers. If the HPX chart built fails to completely span the current *Containing Element*, an artificial "RST collection" edge is inserted, which spans the current *Containing Element* by connecting a sequence of longest adjacent `<undefined>` elements in the `<undefined>` list. This ensures an output of connected RST trees and robustness of discourse parsing.

3.4.3 Generic Annotation Parser (GAP)

The Generic Annotation Parser (GAP), as displayed in Figure 3.4, has been conceived as an abstract parsing system augmenting n input annotation layers with identical primary text data by one new, $n+1$st output annotation layer representing a constituency structure. The building of the output annotations is based on the reduce rule set that contains reduce rules to add new annotations starting from the

[6] The GAP is not invoked for lists with only one `<undefined>` element, as that already covers the respective *Containing Element*.

base annotation layer. In the reduce rules, reference can also be made to information on further input annotation layers.

The base annotation layer must consist of a sequence of adjacent elements that covers the whole text and provides the basis for building the new, output structure. In the case of discourse parsing in SemDok, these are the `<undefined>` elements on the annotation layer HPX, which can be changed into `<n>` or `<s>` elements as exemplified in Figure 3.3.

The rule set contains declarative reduce rules which are selected based on the sequences (mostly pairs) of matching elements on the base layer and on additional conditions referring to other annotation layers.

Since element configurations may be ambiguous i.e. match with multiple rules and associated relations, the quality of each possible output structure is evaluated by means of a scoring function. Scoring is discussed more fully in Section 3.4.3.6, further details of the reduce rules are given in Section 3.4.3.2.

In the following section, we describe the bottom-up passive chart parser that constitutes the core of the GAP.

3.4.3.1 Chart Parser

A discourse parser needs a strategy to handle ambiguities, arising e.g. due to discourse markers which are ambiguous according to the discourse relation they indicate, or their scope. Chart parsing is designed to efficiently store and retrieve information about partial structures that have already been parsed [cf. 16].

The chart of the SemDok Parser is defined by minimally extending the Sekimo fact base format [45] for RST-HP-compliant XML documents such that SemDok chart edges are always potential Prolog node/5 facts representing the XML elements of a RST-HP result document. The chart format is therefore called HPX, for *HP extended*. An HPX chart edge has the following components:

1. Layer: name of the base Layer; in the discourse parsing application: HPX
2. Start: PCDATA offset of the begin of the text range spanned by the chart edge (XML element)
3. End: PCDATA offset of the end of the text range spanned by the chart edge (XML element)
4. NodeID: edge ID of the chart edge (node ID of the XML Element)
5. LoLoI: *List of Lists of child node IDs*, pointers to the child edges
6. Score: score according to probabilistic parsing, cf. Section 3.4.3.6
7. DM-ID-List: list of those discourse markers that have already been used in the parsing history, cf. Section 3.4.3.7
8. Element: XML element name; in the application of discourse parsing, one of `n|s|para|hypo|embed|t|undefined|rstCollection`

The algorithm we employ for discourse parsing is based on *bottom-up passive chart parsing* [cf. 32, 1]. In our variant, no edge sequence is analysed more than once, because in the loop over the current set of edges, each edge is checked only against those right-adjacent edges that have been inserted in the previous loop.

Node sequences of n nodes (as opposed to node pairs of two nodes) are currently analysed in two reduce rules for recognising embedded discourse segments (`<embed>` constructions cf. [27]).

3.4.3.2 Rule Components

Reduce Rules

For each cascade step (EDS+, SDS+, CDS$_{block}$+, and CDS$_{div}$+), an individual set of reduce rules is loaded. The reduce rules presently used in the SemDok parser have been acquired in two ways:

1. rules generated from the SemDok Discourse Marker Lexicon [27] using an XSLT style sheet
2. rules manually encoded in Prolog

Naturally, the rules which were generated from the Discourse Marker Lexicon mainly refer to the annotation layer DMS (cf. Section 3.2). The manually encoded rules typically refer to other layers and are based on findings from qualitative and quantitative corpus analyses, e.g. rules for assigning the Elaboration relation and several of its subtypes which refer to the CHS layer as described in [5].

Listing 3.2 shows the entry for *dagegen* (“in contrast”) in the SemDok Discourse Marker Lexicon. Listing 3.3 shows the Prolog reduce rule generated from it.

The “corpus score” specified in the attribute `@corpusScore` is derived from an XML database consisting of 564 RST tree instances from an annotated subcorpus, in which the discourse markers that indicate the relation are annotated as well. The corpus score in Listing 3.2 specifies that 56% of all occurrences of *dagegen* in the subcorpus were marked to indicate the relation Contrast-multi. The simple score in the attribute `@score`, on the other hand, represents the a priori probability

```
<dm id="c333" typ="lexical">
  <cue>
    <text>dagegen</text>
    <lemma pos="ADV">dagegen</lemma>
    <position>
      <vorfeld>+</vorfeld>
      <mittelfeld>+</mittelfeld>
    </position>
  </cue>
  <rels default="Contrast-multi">
     <relation corpusScore="0.55556" score="1" relname="Contrast-multi"
      skopus="sds+" typ="n" beds-richtung="l"/>
  </rels>
</dm>
```

Listing 3.2 Entry in the discourse marker lexicon.

```
%-------------------------------------------
% Case sentence adverb/coordinating conjunction "dagegen"
% Type N-N with simple lexical DM in N2

reduce_rule(dmlist1, [N2, N1|[]], [Np|[]], L_new_undefined) :-

   gap_baselayer(BaseLayer),

   node(BaseLayer, _Start1, _End1, N1, _, _Score1, _DML1, element('
        undefined')),
   node(BaseLayer, Start2, End2, N2, _, _Score2, DML2, element('undefined')
        ),

   % Constraint:
   one_relation(inclusion_B_in_A, 'undefined', BaseLayer, N2, dm, 'DMS',
        N_d, Start2, End2, Start_d, End_d),

   % Constraint:
      attr('DMS', Start_d, End_d, N_d, 'lemma', 'dagegen'),
   attr('DMS', Start_d, End_d, N_d, 'pos', 'ADV'),


   % Get DMID and check DMID in DML2:
   attr('DMS', Start_d, End_d, N_d, 'id', ID),
   nonvar(ID), nonvar(DML2),
      not(member(('DMS', ID), DML2)),

   reduce_to_N_N(N1, N2, 'Contrast-multi', Np, L_new_undefined, 0.00829, [(
        'DMS', ID)]).
```

Listing 3.3 Reduce in Prolog, generated from the discourse marker lexicon.

of the discourse marker derived from its ambiguity in the Discourse Marker Lexicon. In the entry in Listing 3.2, the simple score is 1, because in the lexicon the discourse marker is specified to indicate only one relation.

A reduce rule as in Listing 3.3 is processed in the following way: First, all components of the current node sequence are retrieved from the chart via their IDs. (In the example in Listing 3.3, the variables $N1$ and $N2$ represent the IDs of the current node sequence.) Then it is checked whether an occurrence of `<dm>dagegen</dm>` on the annotation layer 'DMS' is included in the text span of node $N2$ on the base layer. Finally, it is checked whether the discourse marker has not already been used in the parsing history of the segment represented by $N2$ (using the DM-ID (discourse marker identifier) list $DML2$). When all constraints are satisfied, the required components will be passed to the reduce schema `reduce_to_N_N` which is invoked to insert the edges representing a new multi-nuclear relation into the chart.

The generated score of the rule (0.00829) is not identical with the conditional probability in the `@corpusScore` attribute in the discourse marker lexicon entry in Listing 3.2, since it has been combined with the a priori probability of the relation CONTRAST-MULTI (also acquired from the corpus) when generating the Prolog rule.

Reduce Schemas

When a reduce rule and its constraints match XML annotations of the current node sequence, the edges forming a new RST subtree are inserted into the chart. For this purpose, five reduce rule application schemata are available (similar to the RST application schemata in [29]). Depending on the schema, either a `<hypo>`, a `<para>` or an `<embed>` edge is generated, as well as the corresponding `<n>` and `<s>` edges, and one new `<undefined>` edge, which will be available in the subsequent parsing process. The following schemas are available.

1. `reduce_to_N_S`: mono-nuclear schema
2. `reduce_to_S_N`: mono-nuclear schema
3. `reduce_to_N_N`: bi-nuclear schema
4. `reduce_to_N_N_List_Add`: schema for a "tree-adjoining" construction of multi-nuclear structures. Proper multi-nuclear structures (with more than two nuclei, such as potentially occurring with the relations LIST, SEQUENCE, and their subtypes) will first be initialised by an application of Schema 3 to the first two nuclei. The remaining nuclei will then be added by iteratively applying Schema 4 in the parsing loops to follow (when the constraints of the rule match). This procedure represents one way to derive multinuclear structures using only binary rules. Within Schema 4, incomplete intermediate multinuclear structures are removed from the chart; this is the only possible destructive action during chart building. Nevertheless, some incomplete (from the viewpoint of a correct reference annotation) multi-nuclear structures may still be kept in the chart, since Schema 3 may be applied to non-initial elements of a multi-nuclear structure as well.
5. `reduce_to_embed`: schema for embedded satellite constructions. This schema has actually two components, one for two embedded satellites, and one for three embedded satellites within a nucleus. As `<embed>` constructions can only be built over EDSes, they could alternatively be parsed in a separate function to be invoked before the first call of the GAP. The chart parsing algorithm would then operate on node pairs instead of node sequences, because all the remaining reduce rules in the SemDok parser are binary.

3.4.3.3 Ranking of Reduce Rules

In the SemDok parser, each rule is ranked so that rules are grouped into specific ones, less specific ones, and default rules. The parser tests a node sequence against rules in a group with a more general rank only when no rules in the more specific groups matched. Rule ranking allows for a prioritisation of rules. Another method of prioritising rules employed in the SemDok parser is scoring (cf. Section 3.4.3.6). The two methods supplement one another as ranking represents a discrete gradation of rules that may lead to an absolute exclusion of certain rules, as opposed to scoring which represents a continuous measurement of the quality of parse trees. In the current version, three named ranks are defined, representing the following groups of rules.[7]

1. 'dmlist1': Rules based on lexical discourse markers. If one of these matches, do not continue with the following rule groups.
2. 'elab': Rules based on the annotation of anaphora, mostly indicating ELABORATION. If one of these matches, do not continue with the following rule group.[8]
3. 'list': A group containing one default rule.

3.4.3.4 Supplementary Functions

To check the constraints formulated in the reduce rules a set of *application-independent functions* (i. e. Prolog predicates) is available. It contains predicates for querying information about the general configuration of XML elements on the multiple annotation layers as stored in the Prolog fact base, such as "Are the nodes in *L* children of node *N*?", or "Are the nodes in *L* all the children of node *N*"?

Furthermore, a set of *application-dependent functions* (i. e. discourse parsing-specific predicates) is available. For the most part, it contains query predicates referring to the grammatical annotation, such as: "Does segment '*N1* correspond to a complete sentence?", "Is discourse marker *D1* contained in the first sentence of *N2*?" or "Is the anaphoric expression *A1* the subject of the first sentence in *N2*?"

The purpose of separating application-independent and application-dependent functions is to be able to simply exchange the module containing the application-dependent functions when the GAP is employed for a different application such as sentence parsing.

3.4.3.5 Node Packing

A situation where the parser finds two or more analyses for the same local text range is called *local ambiguity*. Local ambiguity occurs when more than one reduce rule is applicable to an edge sequence, or because a discourse marker is ambiguous, or because the current segments contain several discourse markers indicating different rhetorical relations for the combination of the segments, or because *n* different chart edge sequences were combined to form *n* new chart edges with identical range. Local ambiguity has to be distinguished from cases where the current segments contain several discourse markers indicating *the same* rhetorical relation for the combination of the segments, these are treated by an adjustment of the score, cf. Section 3.4.3.6.

In the original chart parsing algorithm, when *n* analyses are found over the same text span, *n* `<undefined>` edges will be inserted in the chart, and the overall

[7] Not all ranks are used for each document level/in each cascade step.

[8] Rules in which anaphora indicate (a type of) ELABORATION have a default character with respect to rules based on lexical discourse markers, cf. [6].

ambiguity grows exponentially. To eliminate this kind of ambiguity, we use *packed representations* in the line of [40]:[9]

For a set of edges with an identical range (start and end offsets) but different analyses (their relation labels or sets of child edges), only one new `<undefined>` edge will be inserted into the chart. Note that in discourse parsing, the type of an edge (the RST tree type plus its relation label) is irrelevant for its combination with another segment in a rule application, and consequently edges can be packed regardless of their type. In syntax parsing, in contrast, no edges with identical range but different types (e.g. AP, NP, VP) may be packed.

3.4.3.6 Scoring

The sixth argument of a chart edge is its *score*, which assigns a rating to the RST tree it represents. Its purpose is to make competing hypotheses of rhetorical relations comparable. The score of an edge depends on the context in which it is inserted into the chart. In principle, we distinguish two cases in the calculation of a new score. When two or more adjacent discourse segments are combined to form a larger segment (case "children2parent"), a new score is computed for the edge representing the larger segment [cf. 28, 23]. Similarly, when several alternative analyses are combined in a packed edge (case "alternative", cf. Section 3.4.3.5), a new, averaged score is computed for the packed edge. For both cases of score combinations, a number of different mean calculations have been implemented: product, geometric mean, arithmetic mean, quadratic mean, and maximum of the involved scores.

Calculating the product is the common method to combine two independent probability values. [28] alternatively suggested the geometric mean in order to reduce the influence of very low partial scores on the total score, e.g. on account of very few occurrences of a discourse marker in a corpus. The arithmetic mean is the classic average calculation, treating each partial score equally. Using the square mean, good scores have a higher influence on the resulting score than bad ones in comparison with the arithmetic mean. The maximum of the underlying scores is another possible heuristic for the combination of alternative analyses, meaning that a packed edge will get the score of the best-scored edge among the alternatives it represents.

A score newly computed using one of these methods is set off against the score of the rule that has been applied for inserting the edge. The rule score is the a priory probability of the rhetorical relation propagated by the rule combined with the conditional probability of the relation given the discourse marker that was tested in the rule. The probabilities used have been estimated by calculating the percentages of relation occurrences and discourse marker occurrences in the SemDok corpus (cf. Section 3.4.3.2).

Which one of the mean calculations is to be chosen in the two cases can be parametrised in the main call of the SemDok parser, so that the best settings for the parameters can be determined in test runs.

[9] Packed representations were originally introduced as *shared forests* of parse *trees*, as [40] is not a chart parsing approach.

A special case occurs when several reduce rules indicate one and the same rhetorical relation for the same segment combination, which means that a matching rule would lead to a set of new edges that are already in the chart. In that situation, no new RST tree is inserted in the chart but the score of the existing chart edges is updated by increasing it by the rule score of the newly matched rule. That way, the score for a relation will be rated higher, the more cues for the relation are found.

3.4.3.7 DM-ID Lists

The seventh component of a chart edge is a list of identifiers (i.e. values of XML ID attributes of XML elements representing the discourse markers in the linguistic annotations of the input document) of those discourse markers that were used in the parsing history of the RST tree represented by the edge, cf. Section 3.4.3.1. In the derivation of *one* RST tree, *one* discourse marker must have induced exactly *one* relation. Hence, one has to keep account of those discourse markers that have already led to rule applications in the bottom-up parsing process. This purpose is served by the DM-ID list. The set of already consumed discourse markers is indicated on the chart edges of the types `para`, `hypo`, or `embed`. When a new edge is inserted in the chart as the result of a successful rule application, the DM-ID list of the edge for the new RST tree edge consists of the union of the DM-ID lists of its child edges plus the DM-ID(s) of the discourse marker(s) that matched in the current rule application.

A conflict occurs when *packed edges* (cf. Section 3.4.3.5) are inserted in the chart. Like all chart edges, packed edges are specified for exactly one DM-ID list. However, they do not represent one unique, but several tree derivations. Thus, if A and B are two derivations packed in one packed edge, in derivation A, a discourse marker may have been applied that has not been applied in derivation B. To capture all the possible cases, one would have to employ disjunctions of DM-ID lists, which would of course counteract the purpose of packed representations. Hence, in the SemDok parser, the following three heuristics are implemented for combining two DM-ID lists: intersection, union, and the so-called majority union.

When intersection is chosen as the combination method, some DM-IDs may get lost so that the associated discourse markers might (falsely) be applied once again in the subsequent parsing process. When the DM-ID lists of the edges $K_1, K_2 \ldots K_n$ to be packed are combined by the union operation, the packed edge does represent the derivation associated with the edge K_i, but its DM-ID list possibly also contains discourse markers of the derivation associated only with the edge K_j. In the subsequent parsing process, these will (falsely) not be available any more for the continued derivation associated with K_i. The "majority union" of $K_1, K_2 \ldots K_n$ is defined such that the DM-ID list of the packed edge will contain only DM-IDs contained in least 50% of the DM-ID lists associated with $K_1, K_2 \ldots K_n$. It is then possible that some discourse markers may be falsely applied a second time in the subsequent parsing process, however in less cases than with regular list intersection. It is also possible that some discourse markers are falsely not available anymore, however in less cases than with regular list union.

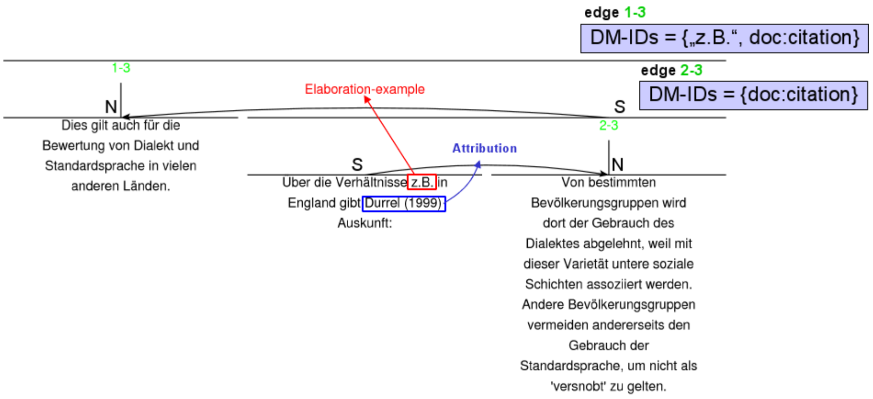

edge 1-3
DM-IDs = {„z.B.“, doc:citation}
edge 2-3
DM-IDs = {doc:citation}
1-3
N
Elaboration-example
S
2-3
Dies gilt auch für die Bewertung von Dialekt und Standardsprache in vielen anderen Länden.
S
Attribution
N
Uber die Verhältnisse z.B. in England gibt Durrel (1999) Auskunft:
Von bestimmten Bevölkerungsgruppen wird dort der Gebrauch des Dialektes abgelehnt, weil mit dieser Varietät untere soziale Schichten assoziiert werden. Andere Bevölkerungsgruppen vermeiden andererseits den Gebrauch der Standardsprache, um nicht als 'versnobt' zu gelten.

Fig. 3.5 Consumed discourse markers in the DM-ID list.

When calling the SemDok parser, one of the three methods to combine DM-ID lists for packed chart edges needs to be specified. Since node packing itself can also be selected or deselected on the parser's top-level call [cf. [18]], it can be determined in evaluation suites which of the three methods yields the best results and how high the error rate is in comparison with parsing without node packing, i.e. with unmanipulated DM-ID lists.

3.4.3.8 Relaxation of the Compositionality Criterion

In [31], a *compositionality criterion* for rhetorical structures is formulated stating that a relation that holds between two text segments also holds between at least two of the nuclei among their embedded segments. Consequently, according to Marcu, in discourse parsing it is sufficient to consider discourse markers only in the top nuclei of the segments concerned. However, numerous counterexamples to this claim can be found in the SemDok corpus, one of which is illustrated in Figure 3.5. Between segment 2 (the satellite) and segment 3 (the nucleus) in the lower RST subtree, an ATTRIBUTION relation holds. This relation is indicated by a citation marker (XML element `<doc:citation>` on the annotation layer DOC) in segment 2, strictly speaking in combination with the colon and the predicate “gibt Auskunft” (“provides information”). In the top-level RST tree (segments 1-3) an ELABORATION-EXAMPLE relation holds, indicated by the discourse marker “z. B.” (“e.g.”) in segment 2, a lower-embedded satellite from the perspective of the top-level tree. Thus, in case of the ELABORATION-EXAMPLE relation in Figure 3.5, a consideration of the embedded nuclei is not sufficient because the relevant discourse marker occurs in the subordinated satellite. The observation that the compositionality criterion is not sufficient was made previously by [20], and many more examples of this kind can be found in the SemDok corpus. Consequently, discourse markers occurring in *any* subsegment of the candidate segments are considered in the reduce rules for cascade step EDS+. For the higher segment levels SDS+, CDS$_{block}$+, and CDS$_{div}$+, however, lexical discourse markers are only considered in the first sentence (SDS) of the second segment, as, with the exception of list contexts, no instance of a lexical discourse marker indicating a relation and occurring in a non-first sentence of its second segment was found in the SemDok corpus.

3.4.4 Traversing the Chart

Our result chart is equivalent to a parse forest [cf. 40, 14], i.e. the chart edges represent the nodes of subtrees connected by the ID pointers in the edges' LoLoIs, representing sets of alternative child node lists (see Section 3.4.3.1). Starting from the "root edge" (the one that spans the complete document and becomes the root note of any result tree), the chart can be traversed along the LoLoIs to generate RST result trees. In the case of a packed edge, the scores of its sub-edges are used to select only the best-rated alternative(s).

Our chart traversion algorithm produces a set of x best-rated parse trees and stores each in a separate Prolog fact base. They can subsequently be exported into RST-HP XML documents using the Sekimo tool prolog2xml[10].

The exact number of the best-rated parse trees cannot be determined beforehand but results from the number of alternative branches traversed on account of the scores found. To better be able to manipulate the number of result trees, it is intended to implement the possibility of specifying a relative threshold ϑ, according to which alternatives can be selected by comparing it with the edge scores instead of automatically selecting all of the x best alternative edges.

For the visualisation and exploration of result trees, a graphical web interface was developed which displays an RST tree structure as well as the related document structure and the text, and also provides the possibility to navigate both structures and between both structures [cf. 18].

3.5 Evaluation

In a parsing experiment, six documents from our development corpus were parsed on the sentence and block level (EDS+ and SDS+). Two of them were not scientific articles, but one web-published article on hypertext and one newspaper article from our partner projects HyTex and Sekimo, cf. Section 3.2.

[10] `http://coli.lili.uni-bielefeld.de/Texttechnologie/Forschergruppe/sekimo/python/`

Table 3.2 Comparative evaluation.

	#relation set	precision	recall
SemDok Exp 1 block level	44 relations	11.53	32.71
SemDok Exp 2 block level	30 relations	12.22	36.64
LeThanh 1 text level	22 relations	38.5	39.6
LeThanh 2 text level	14 relations	39.3	40.5

The six articles contained between 1235 and 9988 wordforms, altogether 26325 wordforms. Their manually built reference annotations of their RST structure contained 2219 elementary discourse segments. The automatically generated initial segmentation was post-edited with respect to faulty segmentations that were introduced on account of errors in the morphology/syntax analysis but otherwise not adapted to the reference annotation.

In the evaluation, the parser's output RST analyses were compared with the reference annotation. Only completely agreeing RST subtrees counted as a match, i.e. RST subtrees had to agree with respect to their relation label and the text range of the combined segment, the text ranges of the constituent segments and the nuclearity labels of the constituent segments.[11] We evaluated two parser runs on the six documents, one in which the full RRSet was used [cf. 6], and one in which a reduced RRSet with 30 (partly underspecified) relation labels was used. The results of the two runs are shown in the first two lines of Table 3.2.

Many relations present in the reference annotations are still not indicated by surface-related discourse markers as are currently analysed in the SemDok approach. Their absence is the main reason for the recall values of 32.71% and 36.64%. Note also that discourse analysis in terms of RST is not an unambiguously feasible task for human annotators, either. While producing the reference annotation of the corpus articles, our human annotators achieved agreements between $\kappa = 0.47$ and $\kappa = 0.81$ for RST analyses of the sentence and block level.[12]

The low precision values of 11.53% and 12.22% arise from the remaining ambiguities of many discourse markers with respect to the relation they indicate and to the scope of a relation, and also by a suboptimal performance of the export of the n-best subtrees from the chart using the scores in the traversal algorithm.

A recent symbolic rhetorical parser is the one by Le Thanh [20, 21, 23, 22, 19] for English, which represents an extension of Corston-Oliver's [8] system. It was evaluated on a test corpus of 20 documents from the *RST Discourse Treebank* [7]

[11] We would like to thank Daniela Goecke of Bielefeld University for implementing the first version of the evaluation program.

[12] Three annotators annotated the same three corpus articles using the full RRSet consisting of 44 relation categories. This setting resulted in $3 * 3 = 9$ agreement ratings. The number of RST subtrees of that annotator which had identified the most RST subtrees was taken as N in the κ formula. A match of RST subtrees (an agreement) was defined as explained for precision and recall above.

that contained between 30 and 1284 word forms [23]. This parser also performs an automatic segmentation of the input text into elementary discourse segments.

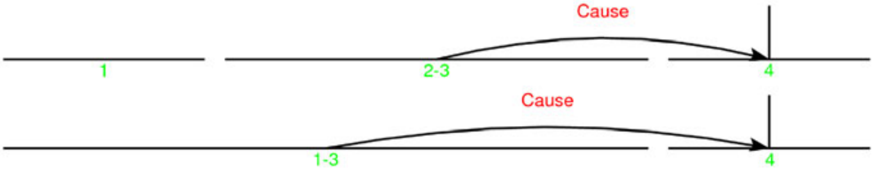
A comparison with this parser also suggests that recall values above 40% are generally hard to achieve on discourse segments that are bigger than sentences. The third and fourth line in Table 3.2 shows Le Thanh's [23] results for discourse parsing on the level of the entire text, which like the CDS$_{block}$ level in the SemDok parser uses sentences as base segments. The results of her parser are better than those of the SemDok parser, but note that her texts are substantially shorter than the scientific articles of the SemDok corpus and that both versions of her relation set are substantially smaller.

3.6 Conclusion

In syntax parsing, the Earley algorithm [11], which combines the basic bottom-up approach with top-down predictions is usually applied to avoid a combinatorial explosion of the number of edges to be inserted in the chart. In discourse parsing, however, categorial top-down constraints corresponding to the phrase labels NP, AP, VP etc. are not available. Hence, in the SemDok approach, elements of the logical document structure of a text are used as top-down constraints by applying the central parsing component multiple times in a cascade for each containing element identified on the logical document structure annotation level of the input text. Further techniques implemented in the SemDok parser to reduce the hypothesis space are the representations of parse forest, node packing and a ranking of reduce rules. Moreover, rule scoring is applied during parsing and evaluated in the chart traversal to reduce the search space. Still the number of possible parses of the scientific articles of our corpus is quite high, as the precision figures of our evaluation indicate. Additionally, many correct analysis (according to reference annotations) are still not found because the rule component lacks certain types of rules especially relevant for higher-level segment types. In the following, we give an overview of the major error types that we identified through an analysis of RST annotations generated by the SemDok parser. On their basis, we point out future enhancements that would further increase its performance.

Disambiguation of discourse markers. The sample corpus consisting of 564 RST subtree annotations together with their indicating discourse markers is actually too small to reliably disambiguate the 96 readings of discourse markers accounted for in the discourse marker lexicon. A bigger corpus with annotations of both rhetorical relations and their indicating discourse markers thus should be acquired.

Cues for higher segment levels. Although the SemDok parser was projected for discourse parsing of text types with a more complex structure, so far only cues from the logical document structure (and partly anaphoric structure) are implemented to analyse rhetorical structures on higher levels than the block level. Particularly, analyses of cues from lexical chaining analyses and text type structure analyses have been prepared for inclusion [4] but not been implemented yet.



1. Aufmerksamkeit gilt als notwendige Voraussetzung für erfolgreiches Lernen.

2. Zwar wurde in der Fremdsprachenerwerbsforschung im Zusammenhang mit der noticing-Hypothese (vgl. Schmidt 1990 und 1995) die Rolle der auf den Input gerichteten Aufmerksamkeit untersucht.

3. Die Funktion der lernerseitigen Aufmerksamkeit für den Output im L2-Erwerb blieb bisher jedoch weitgehend unberücksichtigt.

4. Mit entsprechenden Aufgabenstellungen soll **daher** die Ausrichtung der Aufmerksamkeit auf verschiedene Aspekte der L2-Sprachproduktion manipuliert werden.

Fig. 3.6 Reference annotation above and output of the SemDok parser.

Syntactic and morphological annotation. The morphological and syntactic annotations of the SemDok corpus were produced using the commercial software *Machinese Syntax* from Connexor Oy. For the fairly complex sentences of the scientific articles in the SemDok corpus, however, the software frequently yields false analyses primarily because much of the domain-specific vocabulary occurring in the corpus is apparently not included in the Machinese lexicon. Furthermore, the performance of discourse parsing on sentence level is negatively affected by the frequently missing or erroneous identifications of the embedding structure of paratactic sentences.

Identification of higher-level discourse segments and scope of discourse markers. A top-down identification of higher-level discourse segments other than those predicted by the logical document structure is currently lacking in the SemDok parser. Presently, all RST subtrees constructed during bottom-up parsing yield new complex discourse segments. This leads to an overgeneration of chart edges that currently cannot be disambiguated adequately by the scoring routine. Instead, a top-down prediction of further higher-level, complex segments would be desirable and would also help identify the *scope* of discourse markers more efficiently. Figure 3.6 shows that in the reference annotation (the above structure), the adverb *daher* ("hence") led to a connection of the nucleus segment 4 with a satellite consisting of segments 2-3 by the Cause relation. In contrast, the SemDok parser identified segment 1-3 as the best-rated satellite of the Cause relation indicated by the discourse marker in segment 4. Apart from the fact that such an analysis seems to express an alternative but maybe also correct interpretation of segments 1-4, an independent identification of only 1-3 as a complex discourse segment could have avoided this failed match. For this purpose, an initial thematic segmentation of an input document should be deployed, e.g. according to the algorithms of described in [15] or [13].

References

[1] Allen, J.: Natural Language Understanding, 2nd edn. Benjamin/Cummings, Redwood City (1994)

[2] Asher, N., Lascarides, A.: Logics of Conversation. Cambridge University Press, Cambridge (2003)

[3] Asher, N., Vieu, L.: Subordinating and coordinating discourse relations. Lingua 115(4), 591–610 (2005)

[4] Bärenfänger, M., Hilbert, M., Lobin, H., Lüngen, H., Puskàs, C.: Cues and constraints for the relational discourse analysis of complex text types - the role of logical and generic document structure. In: Sidner, C., Harpur, J., Benz, A., Kühnlein, P. (eds.) Proceedings of the Workshop on Constraints in Discourse, National University of Ireland, Maynooth, Ireland, pp. 27–34. (2006)

[5] Bärenfänger, M., Goecke, D., Hilbert, M., Lüngen, H., Stührenberg, M.: Anaphora as an indicator of elaboration: A corpus study. JLCL - Journal for Language Technology and Computational Linguistics, 49–72 (2008)

[6] Bärenfänger, M., Lobin, H., Lüngen, H., Hilbert, M.: OWL ontologies as a resource for discourse parsing. LDV-Forum GLDV-Journal for Computational Linguistics and Language Technology 23(2), 17–26 (2008)

[7] Carlson, L., Marcu, D., Okurowski, M.E.: RST discourse treebank (2002), `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07` (visited 20.01.2009), Linguistic Data Consortium

[8] Corston-Oliver, S.H.: Identifying the linguistic correlates of rhetorical relations. In: Proceedings of the ACL Workshop on Discourse Relations and Discourse Markers, pp. 8–14 (1998)

[9] Cramer, I., Finthammer, M.: An evaluation procedure for word net based lexical chaining: Methods and issues. In: Proceedings of the Global WordNet Conference 2008, Szeged, Hungary (2008)

[10] Diewald, N., Stührenberg, M., Garbar, A., Goecke, D.: Serengeti – Webbasierte Annotation semantischer Relationen. JLCL - Journal for Language Technology and Computational Linguistics, 74–94 (2008)

[11] Earley, J.: An efficient context-free parsing algorithm. Communications of the Association for Computing Machinery 13(2), 94–102 (1970)

[12] Egg, M., Redeker, G.: Underspecified discourse representation. In: Benz, A., Kühnlein, P. (eds.) Constraints in Discourse, Pragmatics & Beyond, Benjamins, Amsterdam, pp. 117–138 (2008)

[13] Green, S.J.: Lexical semantics and automatic hypertext construction. ACM Computing Surveys 31(4) (1999)

[14] Hanneforth, T., Heintze, S., Stede, M.: Rhetorical parsing with underspecification and forests. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL), Edmonton, Canada (2003)

[15] Hearst, M.A.: TextTiling: A quantitative appraoch to discourse segmentation. Technical Report UCB:S2K-93-24 (1993), http://people.ischool.berkeley.edu/hearst/tiling-about.html (visited 20.01.2009)

[16] Hellwig, P.: Parsing natürlicher Sprachen: Grundlagen und Parsing natürlicher Sprachen: Realisierungen. In: Bátori, I.S., Lenders, W., Putschke, W. (eds.) Computational Linguistics. An International Handbook on Computer Oriented Language Research and Applications, Handbücher zur Sprach- und Kommunikationswissenschaft, de Gruyter, Berlin, pp. 348–431 (1989)

[17] Hilbert, M., Lüngen, H.: RST-HP - Annotation of rhetorical structures in SemDok. Interne Reports der DFG-Forschergruppe 437 "Texttechnologische Informationsmodellierung", Justus-Liebig-Universität Gießen, Fachgebiet ASCL (2009)

[18] Hilbert, M., Lüngen, H., Bärenfänger, M., Lobin, H.: Demonstration des SemDok-Textparsers. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds.) Proceedings of the 9th Conference on Natural Language Processing (KONVENS 2008), pp. 22–28. Ergänzungsband Textressourcen und lexikalisches Wissen, Berlin (2008)

[19] Le Thanh, H.: An approach in automatically generating discourse structure of text. Journal of Computer Science and Cybernetics, Vietnam 23(3), 212–230 (2007)

[20] Le Thanh, H., Abeysinghe, G.: A study to improve the efficiency of a discourse parsing system. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 104–117. Springer, Heidelberg (2003)

[21] Le Thanh, H., Abeysinghe, G., Huyck, C.: Using cohesive devices to recognize rhetorical relations in text. In: Proceedings of the 4th Computational Linguistics UK Research Colloquium (CLUK-4). University of Edinburgh, UK (2003)

[22] Le Thanh, H., Abeysinghe, G., Huyck, C.: Automated discourse segmentation by syntactic information and cue phrases. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), Innsbruck, Austria (2004)

[23] Le Thanh, H., Abeysinghe, G., Huyck, C.: Generating discourse structures for written texts. In: Proceedings of COLING 2004, Geneva, Switzerland (2004)

[24] Lenz, E.A., Lüngen, H.: Dokumentation der Annotationsschicht: Logische Dokumentstruktur. Internal Report, Universität Dortmund, Institut für deutsche Sprache und Literatur/ Justus-Liebig-Universität Gießen, Fachgebiet ASCL (2004), http://www.uni-dortmund.de/hytex/hytex/publikationen.html

[25] Lüngen, H., Puskás, C., Bärenfänger, M., Hilbert, M., Lobin, H.: Discourse segmentation of german written texts. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 245–256. Springer, Heidelberg (2006)

[26] Lüngen, H., Kunze, C., Lemnitzer, L., Storrer, A.: Towards an integrated OWL model for domain-specific and general language wordnets. In: Proceedings of the Fourth Global WordNet Conference (GWC 2008), Szeged, Hungary, pp. 281–296 (2008)

[27] Lüngen, H., Bärenfänger, M., Hilbert, M., Lobin, H., Puskàs, C.: Discourse relations and document structure. In: Metzing, D., Witt, A. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, Text, Speech and Language Technology. Springer, Dordrecht (2010)

[28] Magerman, D.M., Marcus, M.P.: Pearl: A probabilistic chart parser. In: Proceedings of the European ACL Conference, pp. 40–47 (1991)

[29] Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: Toward a functional theory of text organisation. Text 8(3), 243–281 (1988)

[30] Marcu, D.: The rhetorical parsing, summarization, and generation of natural language texts. PhD thesis, University of Toronto (1997)

[31] Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge (2000)

[32] Naumann, S., Langer, H.: Parsing. Teubner, Stuttgart (1994)

[33] Polanyi, L., Culy, C., van den Berg, M., Thione, G.L., Ahn, D.: A rule based approach to discourse parsing. In: Proceedings of the 5th Workshop in Discourse and Dialogue, Cambridge, MA, pp. 108–117 (2004)

[34] Polanyi, L., Culy, C., van den Berg, M., Thione, G.L., Ahn, D.: Sentential structure and discourse parsing. In: Proceedings of the ACL 2004 Workshop on Discourse Annotation, Barcelona, pp. 49–56 (2004)

[35] Reitter, D.: Rhetorical analysis with rich-feature support vector models. Master's thesis, University of Potsdam (2003)

[36] Reitter, D.: Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. In: Seewald-Heeg, U.: (ed) Sprachtechnologie für die multilinguale Kommunikation. Textproduktion, Recherche, Übersetzung, Lokalisierung. Beiträge der GLDV-Frühjahrstagung, Köthen, LDV-Forum, vol. 18(1,2), pp. 38–52 (2003)

[37] Reitter, D., Stede, M.: Step by step: Underspecified markup in incremental rhetorical analysis. In: Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC 2003) at the EACL, Budapest (2003)

[38] Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), Edmonton, Canada (2003)

[39] Tapanainen, P., Järvinen, T.: A non-projective dependency parser. In: Proceedings of the 5th Conference on Applied Natural Language Processing, Association for Computational Linguistics, Washington D.C., pp. 64–71 (1997)

[40] Tomita, M.: An efficient augmented-context-free parsing algorithm. Computational Linguistics 13(1-2), 31–46 (1987)

[41] Walsh, N., Muellner, L.: DocBook: The Definitive Guide. O'Reilly, Sebastopol (1999)

[42] Hilbert, M., Lüngen, H., Bärenfänger, M., Lobin, H.: Demonstration des SemDok-Textparsers. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds.) Proceedings of the 9th Conference on Natural Language Processing (KONVENS 2008), pp. 22–28. Ergänzungsband Textressourcen und lexikalisches Wissen, Berlin (2008)

[43] Webber, B.: D-LTAG: Extending Lexicalized TAG to Discourse. Cognitive Science 28(5), 751–779 (2004)

[44] Witt, A.: Multiple hierarchies: New aspects of an old solution. In: Proceedings of the Extreme Markup Languages, Montreal (2004)

[45] Witt, A., Lüngen, H., Goecke, D., Sasaki, F.: Unification of XML documents with concurrent markup. Literary and Linguistic Computing 20(1), 103–116 (2005)

Part II
Measuring Semantic Distance: Methods, Resources, and Applications

Chapter 4
Semantic Distance Measures with Distributional Profiles of Coarse-Grained Concepts

Graeme Hirst and Saif Mohammad

Abstract. Although semantic distance measures are applied to words in textual tasks such as building lexical chains, semantic distance is really a property of concepts, not words. After discussing the limitations of measures based solely on lexical resources such as WordNet or solely on distributional data from text corpora, we present a hybrid measure of semantic distance based on distributional profiles of concepts that we infer from corpora. We use only a very coarse-grained inventory of concepts—each category of a published thesaurus is taken as a single concept—and yet we obtain results on basic semantic-distance tasks that are better than those of methods that use only distributional data and are generally as good as those that use fine-grained WordNet-based measures. Because the measure is based on naturally occurring text, it is able to find word pairs that stand in non-classical relationships not found in WordNet. It can be applied cross-lingually, using a thesaurus in one language to measure semantic distance between words in another. In addition, we show the use of the method in determining the degree of antonymy of word pairs.

4.1 Semantic Distance

Many applications in natural language processing can be cast in terms of **semantic distance between words** in one way or another. For example, word sense disambiguation can be thought of as finding the sense of the target word that is semantically closest to its context [27]. Real-word spelling errors can be detected

Graeme Hirst
Department of Computer Science, University of Toronto,
Toronto, Ontario, Canada M5S 3G4
e-mail: gh@cs.toronto.edu

Saif Mohammad
Institute for Information Technology, National Research Council Canada,
Ottawa, Ontario, Canada, K1A 0R6
e-mail: saif.mohammad@nrc-cnrc.gc.ca

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 61–79.
springerlink.com

Table 4.1 Some NLP applications that have used semantic distance measures [18].

Cognate identification
Coreference resolution
Document clustering
Information extraction
Information retrieval
Multi-word expression identification
Paraphrasing and textual entailment
Question answering
Real-word spelling error detection
Relation extraction
Semantic similarity of texts
Speech recognition
Subjectivity determination
Summarization
Textual inference
Word prediction
Word sense disambiguation
Word-sense discovery
Word-sense dominance determination
Word translation

by identifying words that are semantically distant from their context and the existence of a spelling variant that is semantically much closer [8]. Word completion and prediction algorithms may rank those candidate words higher that are semantically close to the preceding context [14]. Table 4.1 lists a number of applications of NLP identified by Mohammad [18] that have been attempted with semantic distance measures.

In particular, semantic distance measures are important in any application that involves finding **lexical chains** in a text – that is, sequences of identical or semantically close words in a text. Lexical chains arise naturally in text that is coherent and cohesive, and thus they can be good indicators of the topic structure of a text.

Table 4.2 Intuitions of semantic distance.

Semantically close	**Semantically distant**
bank–money	doctor–beer
apple–fruit	painting–January
apple–banana	money–river
tree–forest	apple–penguin
pen–paper	nurse–bottle
hot–cold	pen–river
mistake–error	clown–tramway
car–wheel	car–algebra
dog–bark	faint–porpoise
bread–butter	asphalt–chocolate

Some examples of word pairs that intuitively are semantically close and semantically distant are shown in Table 4.2. People's intuitions of semantic distance are remarkably consistent. In experiments in which subjects are asked to judge the semantic distance of word pairs on a scale of 0 to 4, correlation between subjects is around .9 [29, 17].

We say that two terms are **semantically related** (or **semantically close**) if either there is a lexical semantic relation between them, such as synonymy, hyponymy, meronymy, or troponymy, or a non-classical relation [26], such as role-filler of action, causal relation, co-occurrence, or even just a strong association. We say that two semantically close terms are **semantically similar** if the relation between them is synonymy, hyponymy, or troponymy. For example, the pairs *dog–paw* and *dog–bark* are semantically related but not similar; the relationships are meronymy and typical-action, respectively. The pair *dog–golden retriever* is not only semantically related by hyponymy but is also semantically similar.

The metaphor of semantic distance implies that the measure of relatedness of two words is a continuous function with metric properties yielding a real number in the interval $[0,\infty)$, where 0 means identity and larger values imply larger distances or less relatedness. On the other hand, similarity implies a continuous function yielding a real number in $[0,1]$, where 1 means identity and 0 means maximal dissimilarity. It is necessary, therefore, to keep in mind which view is being taken at any particular time, and map between them as necessary.

Lexical ambiguity is a serious complication for these intuitive ideas of semantic distance. Relations are defined on words yet they depend on senses or concepts; two words may be related with respect to some of their senses but unrelated with respect to others. In this paper, we will take word senses and concepts to be much the same thing – we need not be concerned with the distinctions between them nor with concepts that are unlexicalized (that have no word) – and we will implicitly take a word to be, more precisely, a *lexical unit* composed of a surface string and a sense. However, in many instances, if a word is ambiguous, we know only the surface string and not its particular sense in the instance. In our exposition below, we will use the word *word* sometimes to refer to a complete lexical unit and sometimes to refer to just the surface string; the intent will be clear from the context in each case.

4.2 Measures of Semantic Distance

There are many ways that semantic distance can be computed in NLP applications. The first class of methods is **resource-based measures** that use the lexicographers' judgments that are implicit in thesauri, dictionaries, or wordnets. In a thesaurus, for example, the semantic distance between two words can be defined as the length of the path between them through the thesaurus's category structure and/or cross-references and index [25, 10]. In a dictionary or wordnet it can be the number of words that occur in the definitions of both target words and possibly, in the case of a wordnet, their neighbours [1]. In a wordnet, it can be the length of the path from one word-sense (synset) to the other (possibly with scaling factors to account for change

Table 4.3 Correlations of several resource-based measures of semantic relatedness with data on human judgments from experiments by Miller and Charles [17] (*M&C*) and by Rubenstein and Goodenough [29] (*R&G*). Based on a table by Budanitsky and Hirst [4], with additional data.

Measure	M&C	R&G
Hirst and St-Onge [9]	.744	.786
Jiang and Conrath [11]	.850[a]	.781[a]
Leacock and Chodorow [13]	.816	.838
Lin [15]	.829	.819
Resnik [28]	.774	.779
Roget-as-tree [10]	.878	.818
Gloss overlaps [1]	.67	.60
Latent semantic analysis [2]	.73	.64

[a] Absolute value of correlation coefficient.

in grainedness with depth [9, 13, 31]); or it can be the amount of information shared by both nodes [11, 28, 16]. Most of these methods are reasonably successful in that they correlate well with the human judgments observed in experiments [4]; see Table 4.3. However, they also have serious limitations:

- Each measure is only as good as the resource it depends on. And most wordnet measures use only the noun portion of the wordnet and only the hyponymy relation.
- The measures typically do not work across parts of speech; that is, one can compare nouns only to other nouns, verbs only to other verbs, and so on.
- Non-similarity relationships are not well covered.
- High-quality resources are not available for many languages.
- The role of context is not accounted for.

An alternative to resource-based measures that overcomes these limitations is to use a **distributional measure** as a proxy for ‘real’ semantics. These methods look only at surface strings of words without regard to their sense. In this class of methods, e.g., [15, 6, 30], we say that two words are semantically related or similar if they tend to co-occur with similar word contexts – that is, if they have similar distributions among other words. The distance between two words is thus defined as the distance between the distributions of the contexts in which they occur. For example, if our target word is *credit* and we see the phrase *a rise in credit and the money supply* in the corpus, we will add 1 to our count of occurrences of *credit* in contexts of *rise*, of *money*, and of *supply*, building a **distributional profile of the word**. Later, we might observe that *debit* tends to occur with many of the same context words, and hence has a distributional profile similar to that of *credit*. Within this idea, there are many definitions of context (e.g., a window of *n* tokens or a syntactic argument relationship), many definitions of “tend to co-occur” (e.g., conditional probability or pointwise mutual information), and many measures of distributional similarity (e.g., α-skew divergence, cosine, Jensen-Shannon divergence, Lin's similarity measure). To define a specific measure, a choice must be made for each of these parameters. See Mohammad and Hirst [19] for a detailed survey of these methods.

These methods overcome some of the limitations of the resource-based approaches. Being corpus-based, they reflect true language usage for which a corpus is available and they are not limited to any particular part of speech or lexical relationship. Moreover, by their very definition they take into account at least a local view of context.

But these methods have limitations too, the most serious of which is that they don't actually work. Their performance is mediocre to awful; Weeds [30] experimented with a number of measures and found their correlation with human data to be between .26 and .62; one of the poorer measures that she experimented with returned this list as the ten words most similar to *hope*: *hem, dissatisfaction, dismay, skepticism, concern, outrage, break, warrior, optimism, readiness.* Moreover:

- The measures are based only on the occurrence of the surface forms of words, not meanings; hence ambiguity is a confound. For example, *credit* has both financial and non-financial senses (… *credited with the invention of the sextant*), but contexts of the different meanings will be conflated in the word's distributional profile. This leads both to attenuation of the measures in the case of true relatedness and to spuriously higher measures between unrelated words.
- They rely on inter-substitutability, which is far too strict a criterion for similarity, let alone relatedness.
- They require enormous corpora to gather sufficient data. Weeds [30] found that the 100M-token British National Corpus was adequate for gathering data for only 2000 word-types. Yet their use in tasks such as real-word spelling correction requires distributional data for a very large vocabulary. This is especially a problem for applications in specific domains and in low-resource languages.

4.3 A Hybrid Method for Semantic Distance Measures

We propose a solution to the limitations of these two classes of methods of measuring semantic distance: a hybrid method that uses both distributional information and a lexicographic resource [21, 18]. Our goal is to gain the performance of resource-based methods and the breadth of distributional methods. The central ideas are these:

- In the lexicographical component of the method, concepts are defined by the category structure of a **Roget-style thesaurus**.
- In order to avoid data sparseness, the concepts are very **coarse-grained**.
- The distributional component of the method is based on concepts, not surface strings. We create **distributional profiles of concepts**.

A Roget-style thesaurus classifies all lexical units into approximately 1000 **categories**, with names such as CLOTHING, CLEANNESS, and DESIRE. Each category is divided into paragraphs that classify lexical units more finely.[1] We take these thesaurus categories as the coarse-grained concepts of our method. That is, for our semantic distance measure, there are only around 1000 concepts (word-senses) in the world; each lexical unit is just a pairing of the surface string with the thesaurus category in which it appears.

In the distributional component of the method, we look at the distribution of these concepts in word contexts. For example, when we see in the corpus *a rise in credit and the money supply*, and given that *credit* appears in category 729 FINANCE in the thesaurus, it's now the count for category 729 that we increment for the context words *rise, money*, and *supply*. To implement this idea, just as for the word-distribution methods, we must choose a definition of context, a measure of strength of association, and a measure of distributional similarity. Given these distributional profiles of concepts, we then define the distance between two concepts as the distance between the distributions of the contexts in which they occur.

But what if a word is ambiguous – appears in more than one thesaurus category? An inability to cope with lexical ambiguity, after all, was one of the limitations of the distributional method that we described earlier. We resolve the ambiguity by bootstrapping as follows. On the initial pass, we count a word for all its categories. This gives a noisy result, but, unlike the word-distribution case and as a consequence of the coarse-grainedness of the concepts, the signal shows through because there are many words in each category. On the second pass, we disambiguate each word by taking the greatest strength of association from the first pass. (We found that additional passes don't increase accuracy.) We define the distance between two lexical units as the distance between their closest senses.

Thus the method is still primarily distributional at heart; its use of lexicographic information is solely for mapping words to the coarse-grained set of concepts. Therefore, we cannot expect it to have the fine performance of measures that are based on rich lexical resources. Nonetheless, the distributional component will give it the breadth that is presently lacking in measures based on those resources.

4.4 Evaluation in Monolingual Applications

We carried out several task-oriented monolingual evaluations of our hybrid method. Our corpus was the British National Corpus, our online thesaurus was the *Macquarie Thesaurus* [3], and context was defined to be a ±5-word window, We defined four different versions of the method by choosing four combinations of measures of strength of association and distributional similarity that are frequently used in the literature on the simple word-distance measures described in section 4.2 above:

- Conditional probability (cp) with
  - α-skew divergence (ASD_{cp});

[1] We do not use other characteristics of Roget-style thesauri, such as the hierarchical structure of the category system, the index, the cross-references, and the further subdivision of paragraphs.

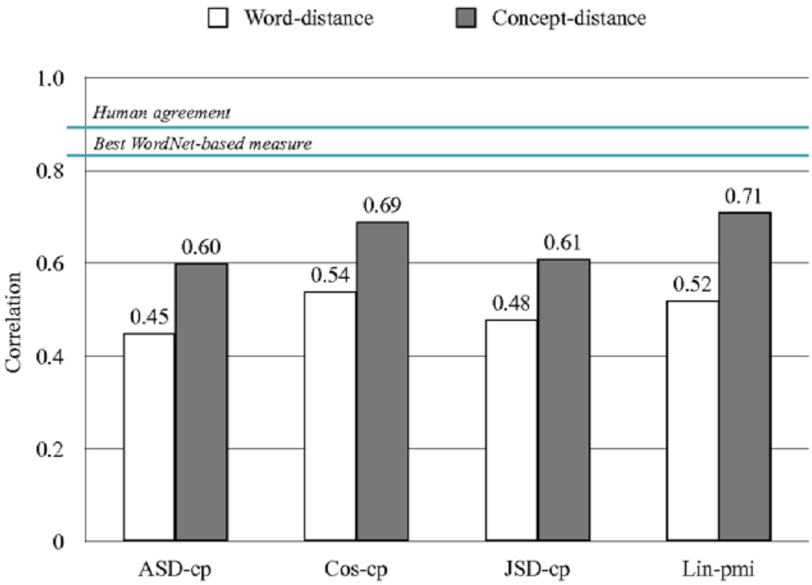


Fig. 4.1 Performance of four distributional concept-distance measures (grey bars) compared with the corresponding word-distance measures (white bars) on the task of ranking word-pairs by semantic distance (correlation with human judgments).

  - Jensen-Shannon divergence (JSD_{cp});
  - Cosine similarity (Cos_{cp}).

- Pointwise mutual information (*pmi*) with Lin's [16] distributional similarity[2] (Lin_{pmi}).

We then compared these four distributional concept-distance measures with distributional word-distance measures using the same four choices.

Our first evaluation was simply to compare the measures' ranking of word-pair distances with human norms [21]. The results are shown in Figure 4.1. In each case, using concepts instead of words improved the results markedly. Nonetheless, as we would expect, the performance is not at the level of the best WordNet-based measures (shown in Table 4.3).

Our second evaluation was to use the measure in correcting real-word spelling errors. Hirst and Budanitsky [8] presented a semantic-distance method for finding and correcting real-word spelling errors in a text, and used it to compare six WordNet-based semantic distance measures. We tried our four measures in the method, along with the corresponding four word-distance versions, with the results shown in Figure 4.2 [21]. The y-axis shows the **correction ratio** for each method, which is a statistic that takes into account both the number of errors corrected and the number

[2] Lin's distributional similarity measure [16] should not be confounded with his WordNet-based semantic distance measure [15], which was mentioned in section 4.2 above.

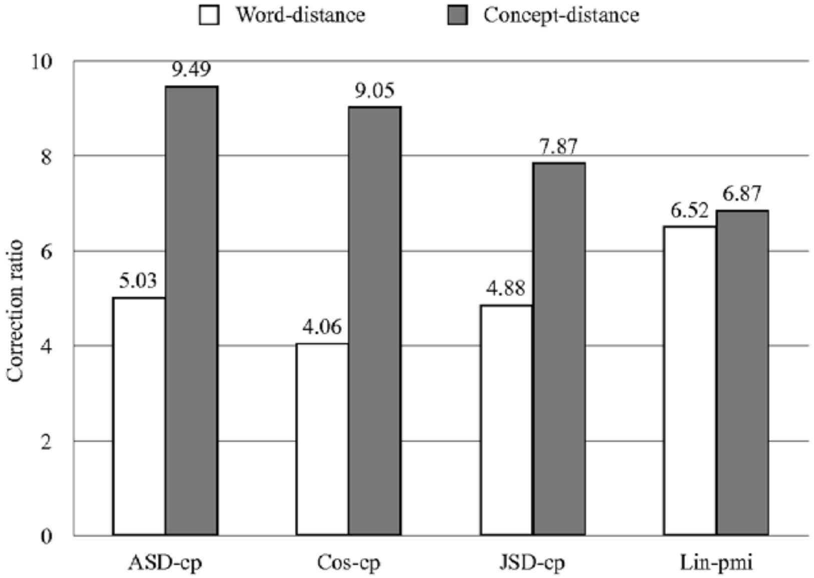


Fig. 4.2 Performance of four distributional concept-distance measures (grey bars) compared with the corresponding word-distance measures (white bars) on the task of real-word spelling-error correction.

of non-errors flagged as errors (false positives). Again, concept-distance measures give better results than word-distance measures, and except for Lin_{pmi}, the difference is quite large. In fact, here the performance of the two best concept-distance measures exceeded that of all but one of the WordNet-based measures as well – though the WordNet-based measure that did better, that of Jiang and Conrath [11], did *much* better, with a score of 12.91 [21]; the second-best WordNet-based measure scored 8.48.

It should be noted that the Rubenstein and Goodenough word-pairs used in the ranking task and the real-word spelling errors in the correction task are all nouns. We expect that the WordNet-based measures will perform less well when other parts of speech are involved, as those hierarchies of WordNet are not as extensively developed. Further, the various hierarchies are not well connected, nor is it clear how to use these interconnections across parts of speech for calculating semantic distance. On the other hand, our hybrid measures do not rely on any hierarchies (even if they exist in the thesaurus) but on sets of words that unambiguously represent each sense. And because our measures are tied closely to the corpus from which co-occurrence counts are made, we expect the use of domain-specific corpora to give even better results.

Our other two monolingual evaluations involved word senses. In the task of determining which sense of a word is dominant in a text, we achieved near upper bound results [20]. And using the measures in word sense disambiguation with an unsupervised naive Bayes classifier, we achieved respectable results in SemEval 2007 [23].

4.5 Extension to Cross-Lingual Applications

4.5.1 Method

It is not necessary in our method that the corpus of text used to determine the distributional profiles of concepts be in the same language as the thesaurus used to define the concepts. In particular, the thesaurus may be in English (E) while the corpus is in a lower-resource language L that has no Roget-style thesaurus. All that is necessary to make this work is a **bilingual dictionary** from L to E that can map the words of the corpus from L to their thesaurus concepts in E. Of course, there will be ambiguity in the translation that creates spurious candidate senses, but this is background noise, as before, that can be eliminated by bootstrapping [22].

Figure 4.3 illustrates the method with two examples in which German plays the role of the low-resource language (see section 4.5.2 below). The first example, *Stern*, is mapped by the bilingual dictionary to *star*, which has additional senses in English; the second example, *Bank*, is ambiguous in German and is mapped to two different English words, *bank* and *bench*, in its different senses (Figure 4.3*(a)*). Concepts, that is thesaurus categories, are obtained for each of the English words (Figure 4.3*(b)*), at which point the English words themselves can be ignored (Figure 4.3*(c)*); observe that some of the concepts are spurious, relative to the original German words, being artifacts of the intermediate English (Figure 4.3*(d)*). However, on the next iteration in the bootstrapping process, these spurious concepts can be identified and removed (Figure 4.3*(e)*) because of their relatively low strength of association with the original German words.

4.5.2 Evaluation

We evaluated the method on two tasks, with German playing the role of the low-resource language L. Of course, German is not really a low-resource language, but the logic of the evaluation requires that the test language L actually have sufficient resources that our method can be compared with resource-based monolingual methods in L. The two tasks were ranking German word pairs for relatedness and solving "Word Power" problems (which require finding the word semantically closest to the target word from a choice of four alternatives) from the German edition of *Reader's Digest*. Our aim was not to perform better than the monolingual method but merely to obtain results that are not markedly poorer; after all, the cross-lingual method is inherently noisy, and is intended for situations only when the resources for monolingual methods are not available at all.

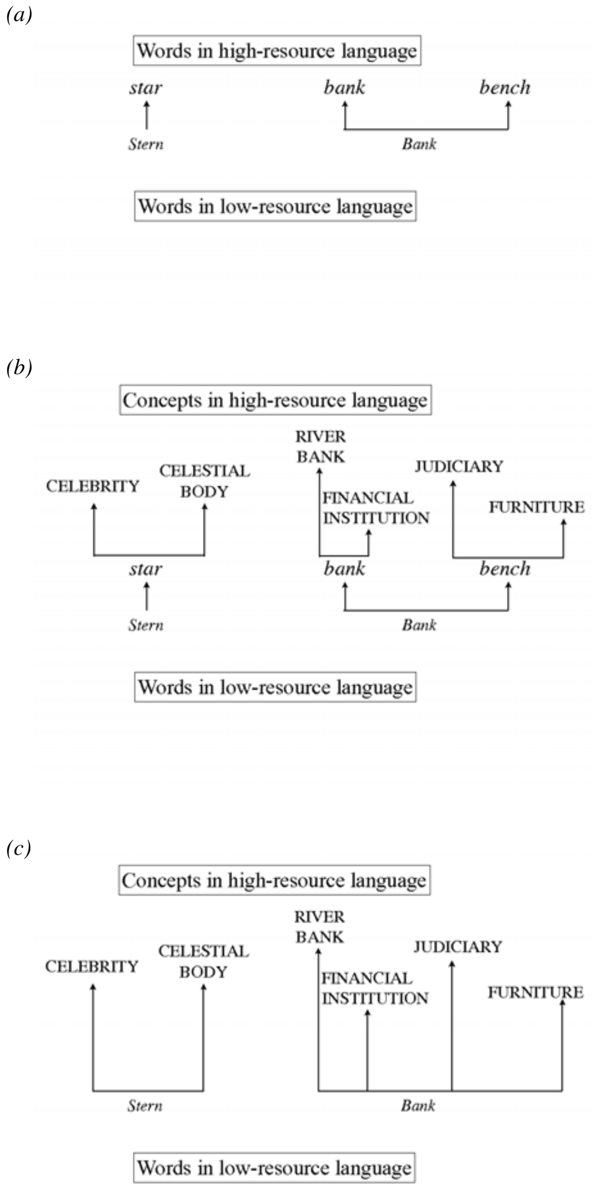


Fig. 4.3 Cross-lingual examples (German to English) demonstrating how bootstrapping removes the artifacts of lexical ambiguity. *(a)* The bilingual dictionary maps the words from German to English. *(b)* The English words are then mapped to thesaurus concepts. *(c)* The English words can now be ignored. [*Figure continues on next page.*]

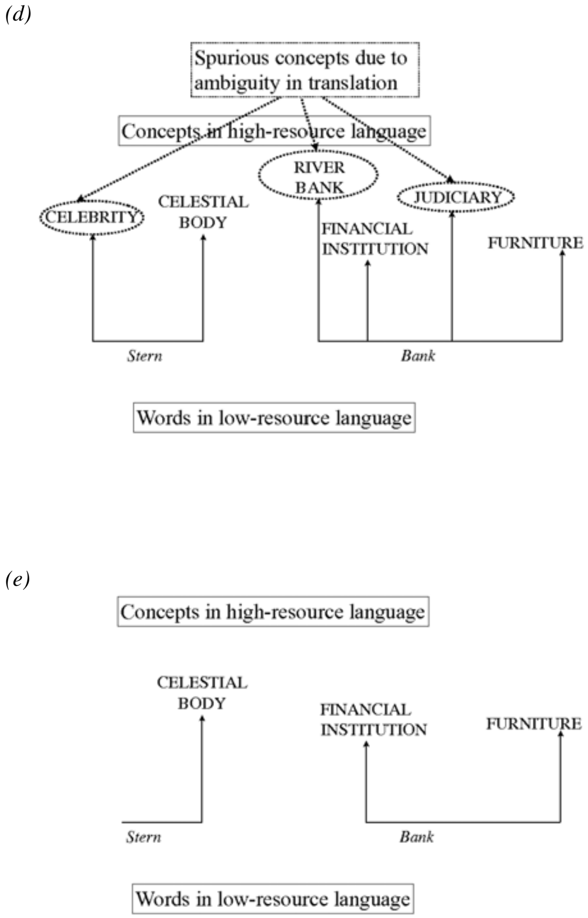

(d)
Spurious concepts due to ambiguity in translation
Concepts in high-resource language
RIVER BANK
JUDICIARY
CELEBRITY
CELESTIAL BODY
FINANCIAL INSTITUTION
FURNITURE
Stern
Bank
Words in low-resource language
(e)
Concepts in high-resource language
CELESTIAL BODY
FINANCIAL INSTITUTION
FURNITURE
Stern
Bank
Words in low-resource language

Fig. 4.3 (cont.) *(d)* Some of the concepts are spurious artifacts of the intermediate English. *(e)* In the bootstrapping process, these spurious concepts can be identified and removed.

As resources for the cross-lingual measure, we used the German newspaper corpus *taz* and the German–English bilingual lexicon BEOLINGUS. As before, the English thesaurus was the *Macquarie Thesaurus*. We tried the same four versions of the method that we used in the evaluations of section 4.4 above. Our benchmark for comparison as a monolingually based semantic distance measure in the same tasks was WordNet-style measures (see section 4.2 above) with GermaNet as the resource; in addition to the measures of Jiang and Conrath, Lin, and Resnik (see Table 4.3) we also used two pseudo-gloss-based measures proposed explicitly for GermaNet by Gurevych [7].

We found the cross-lingual method to be not just the equal of the GermaNet-based monolingual methods but better in both tests. Figure 4.4 illustrates the results. The upper histogram shows Spearman rank correlations with human rankings of the best of our cross-lingual measures (which was Lin_{pmi}) and the best of the GermaNet measures (which was Jiang and Conrath's); the former achieves a notably better result. The lower series of histograms shows results on the "Word Power" problems for the best methods of each type; for the GermaNet methods, this was one of Gurevych's, and for the cross-lingual method this was JSD_{cp} and Lin_{pmi} equally, with Cos_{cp} only a tiny amount behind. Although the cross-lingual measures have a lower precision than the best monolingual measure, they have higher recall and overall a slightly better F-score. The higher recall implies that the bilingual dictionary had a better coverage of the vocabulary of the "Word Power" problems than GermaNet did.

In addition to these tests, we tried the cross-lingual method out in a Chinese–English setting in the SemEval 2007 task of choosing the best English translation for an ambiguous Chinese word in context, and we achieved good results with an unsupervised naive Bayes classifier [23].

4.6 Antonymy and Word Opposition

In this section, we show that our method for semantic distance can be extended to solve the related problem of finding words that are antonyms or, more generally, pairs of words whose meanings are contrasting or opposed to one another [24]. Thus we want to go beyond the conventional kinds of antonymy (*wet–dry, open–closed, life–death*), which are already well-recorded in lexical resources such as WordNet, to a more-general notion of contrast in meaning (*closed–accessible, flinch–advance, cogent–unconvincing*) which is largely unrecorded. This has application in tasks such as detecting contradictions and differences in opinion, and detecting paraphrases in which one alternative is negated (*caught–not evaded*).

We base our approach on two hypotheses:

- **The co-occurrence hypothesis** (Charles and Miller [5]): Antonyms co-occur more often than chance.
- **The distributional hypothesis** (after Justeson and Katz [12]): Antonyms tend to occur in similar contexts.

By comparing 1000 randomly chosen antonym pairs from WordNet with a control set of 1000 randomly chosen (non-antonymous) word pairs, we showed [24] that both of these hypotheses are correct: antonym pairs have a higher strength of co-occurrence (by pointwise mutual information) than random word pairs ($p < .01$) and are distributionally more similar (by Lin's measure [16]) than random pairs ($p < .01$). The same is true, of course, of semantically similar and semantically related words. So these two hypotheses alone are not sufficient to identify contrasting word pairs.

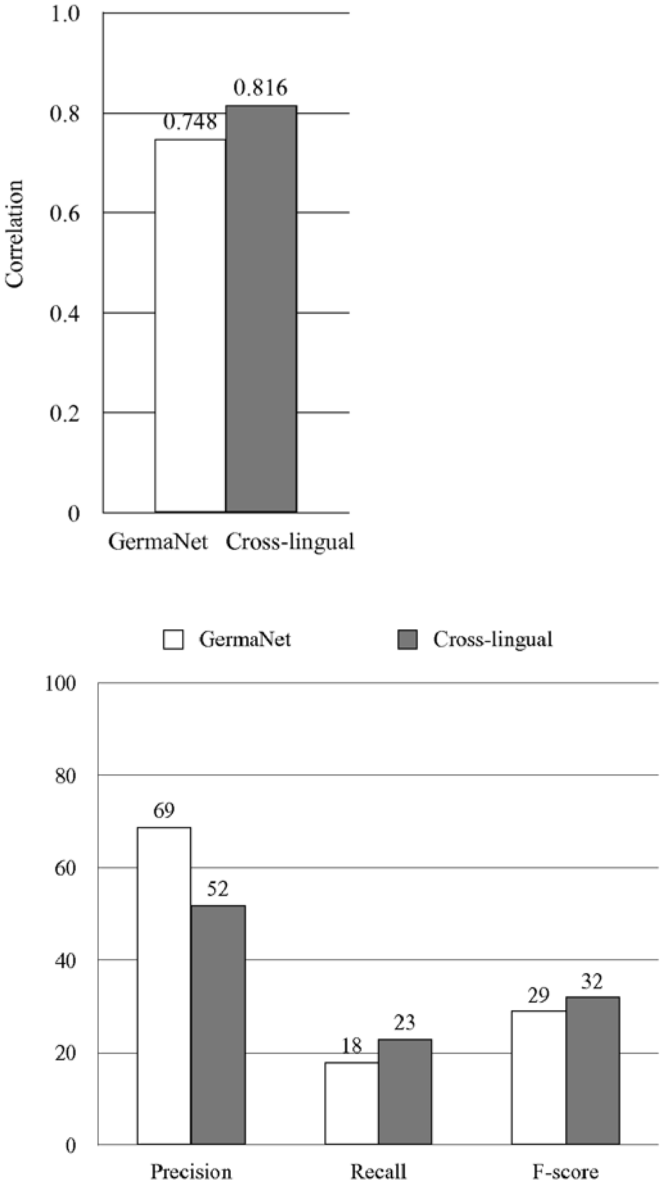


Fig. 4.4 Performance of the cross-lingual method (grey bars) compared with monolingual GermaNet-based method (white bars) on ranking word-pairs by distance (correlation with human judgments) *(top)* and on *Reader's Digest* "Word Power" problems (precision, recall, and F-measure) *(bottom)*.

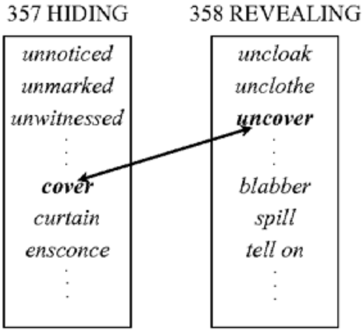


Fig. 4.5 Two thesaurus categories are assumed to be contrasting if each contains one member of a pair from the seed set of antonyms.

The central ideas of our method are these: First, we identify **contrasting category pairs** in the thesaurus using the structure of the thesaurus and a set of seed antonym pairs. We then determine the *degree of antonymy* between a pair of words, one from each of a pair of contrasting categories, using the two hypotheses mentioned above and, again, the structure of the thesaurus.

4.6.1 Contrasting Categories

We have two heuristics for recognizing contrasting categories. First, thesaurus lexicographers often explicitly place contrasting categories adjacent to each other; for example, the LOVE category may follow the HATE category. So we assume that all adjacent category pairs are contrasting. This is obviously untrue in general; for example, the other category adjacent to HATE may be INDIFFERENCE. Second, we manually create a list of 16 affixes that tend to generate antonyms, such as X–*anti*X (*clockwise–anticlockwise*), X*less*–X*ful* (*harmless–harmful*), and *im*X–*ex*X (*implicit–explicit*) and we use this list to generate a seed set of about 2600 pairs of likely antonyms.[3] We then assume that a pair of thesaurus categories containing a word pair in the seed set is contrasting (see Figure 4.5); this is our second heuristic. Additionally, we also use antonym pairs from WordNet to find contrasting categories where possible; WordNet contains 10,800 antonym pairs for which both words were in our thesaurus.

[3] The affix list obviously overgenerates (*part–depart; tone–intone; sect–insect; coy–decoy*), but this has little effect on the results.

‘hardened in feelings’	‘resistant to persuasion’	‘persistent’
obdurate:	**obdurate:**	**obdurate:**
a. *meager*	a. *yielding**	a. *commensurate*
b. *unsusceptible*	b. *motivated*	b. *transitory**
c. *right*	c. *moribund*	c. *complaisant*
d. *tender**	d. *azure*	d. *similar*
e. *intelligent*	e. *hard*	e. *uncommunicative*

Fig. 4.6 GRE-style multiple-choice closest-opposite questions using the same prompt in different senses. The correct answer is marked with an asterisk.

4.6.2 Degree of Antonymy

We can now determine the **degree of antonymy** between two thesaurus categories, and from that between two lexical units (a word and its thesaurus category), and from that between two words:

- **Categories:** Following the distributional hypothesis for antonyms, we stipulate that the degree of antonymy between two *contrasting* categories is proportional to the semantic closeness of the two categories as measured by our hybrid semantic-distance measure (section 4.3 above).
- **Lexical units:** We assign four discrete levels of antonymy. If the units do not occur in contrasting categories, then they have ZERO antonymy. Otherwise, if each occurs in its respective category in the same paragraph as one of the seeds that is the basis for the contrast between the categories, then antonymy is HIGH. Otherwise, following the co-occurrence hypothesis, the antonymy is MEDIUM or LOW depending on the strength of co-occurrence between the categories.
- **Words:** We take the degree of antonymy of two words to be that of their most antonymous pair of senses.

4.6.3 Evaluation

We evaluated the method on 950 GRE-style multiple-choice closest-opposite questions. Each question contains a prompt word and five alternatives from which the closest opposite to the prompt must be chosen. Typically the alternatives will include as distractors both another close opposite and a near-synonym of the prompt. An ambiguous word may appear in more than one question in different senses; Figure 4.6 shows three questions all using the prompt *obdurate* in different senses. (Of course, the system is not informed of the intended sense.)

The results are shown in Figure 4.7. The baselines for our evaluation are simple random choice from the five alternatives, and looking for the answer in WordNet but choosing at random if none of the alternatives are listed as an antonym of the prompt. In fact, the answer is so rarely found in WordNet that it scarcely improves on random choice.

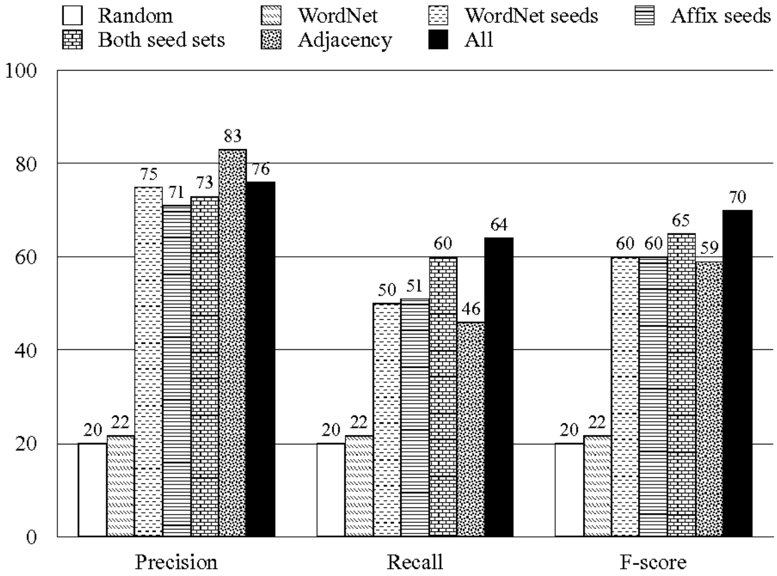


Fig. 4.7 Results of evaluation of method for determining degree of antonymy. In each group, the bars show, from left to right: a random-choice baseline; random choice except using WordNet antonyms where possible; the method using only WordNet-generated seed-pairs; the method using only affix-generated seed-pairs; the method using both seed sets; the method using only the category-adjacency heuristic; and the method using all heuristics.

We also tried the heuristics individually as well as in combination. The relatively small set of affix-generated seed-pairs performed almost as well by itself as the larger set of WordNet-generated seed-pairs; but the two together performed better than either alone. The simple adjacency heuristic achieved better precision than this combination; however, its recall was much lower. The highest F-score was achieved by a combination of all three heuristics.

4.7 Conclusion

There have been many prior proposals for measuring semantic distance: measures based on lexicographical resources and measures based on word distributions in word contexts. Both kinds have significant limitations. By proposing a hybrid measure based on distributions of coarse-grained concepts (thesaurus categories) in word contexts, we avoid the limitations of purely corpus-based and WordNet-based measures. Its performance is competitive with WordNet-based measures (and better than corpus-based measures), it operates across parts of speech, and it offers the possibility of cross-lingual use for resource-poor languages. In addition we have shown how it can be used in a method for determining the degree of antonymy between words.

Acknowledgements. This paper is based on work that was first reported in the following papers: [21], [22], [23], and [24]. Parts of the work were carried out in collaboration with Bonnie Dorr and Philip Resnik, University of Maryland, and Iryna Gurevych and Torsten Zesch, Technische Universität Darmstadt. The research was supported by the Natural Sciences and Engineering Research Council of Canada (Hirst, Mohammad), the University of Toronto (Mohammad), the U.S. National Science Foundation (Mohammad, Dorr), the Human Language Technology Center of Excellence (Dorr), die Deutsche Forschungsgemeinschaft (Gurevych, Zesch), and the U.S. Office of Naval Research and the U.S. Department of Defense (Resnik). We are grateful to Alex Budanitsky, Diana McCarthy, Michael Demko, Suzanne Stevenson, Frank Rudzicz, Afsaneh Fazly, Afra Alishahi, Siddharth Patwardhan, Xinglong Wang, and Vivian Tsang for helpful discussions.

References

[1] Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pp. 805–810 (2003)

[2] Beigman Klebanov, B.: Semantic relatedness: Computational investigation of human data. In: Proceedings of the 3rd Midwest Computational Linguistics Colloquium, Urbana-Champaign, USA (2006)

[3] Bernard, J. (ed.): The Macquarie Thesaurus. Macquarie Library, Sydney, Australia (1986)

[4] Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of semantic distance. Computational Linguistics 32(1), 13–47 (2006)

[5] Charles, W.G., Miller, G.A.: Contexts of antonymous adjectives. Applied Psychology 10, 357–375 (1989)

[6] Dagan, I.: Contextual word similarity. In: Dale, R., Moisl, H., Somers, H. (eds.) Handbook of Natural Language Processing, pp. 459–475. Marcel Dekker Inc., New York (2000)

[7] Gurevych, I.: Using the structure of a conceptual network in computing semantic relatedness. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing, Jeju Island, Republic of Korea, pp. 767–778 (2005)

[8] Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering 11, 87–111 (2005)

[9] Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, ch. 13, pp. 305–332. The MIT Press, Cambridge (1998)

[10] Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and semantic similarity. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003), pp. 212–219 (2003)

[11] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference on Research in Computational Linguistics (ROCLING X), Taiwan, pp. 19–33 (1997)

[12] Justeson, J.S., Katz, S.M.: Cooccurrences of antonymous adjectives and their contexts. Computational Linguistics 17, 1–19 (1991)

[13] Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, ch. 11, pp. 265–283. The MIT Press, Cambridge (1998)

[14] Li, J., Hirst, G.: Semantic knowledge in a word completion task. In: Proceedings, 7th International ACM SIGACCESS Conference on Computers and Accessibility, Baltimore, MD (2005)

[15] Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 36th annual meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING- ACL 1998), pp. 768–774 (1998)

[16] Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304 (1998)

[17] Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and Cognitive Processes 6(1), 1–28 (1991)

[18] Mohammad, S.: Measuring semantic distance using distributional profiles of concepts. PhD thesis, Department of Computer Science, University of Toronto (2008)

[19] Mohammad, S., Hirst, G.: Distributional measures as proxies for semantic relatedness (2005), http://ftp.cs.toronto.edu/pub/gh/Mohammad+Hirst-2005.pdf

[20] Mohammad, S., Hirst, G.: Determining word sense dominance using a thesaurus. In: Proceedings of the 11th conference of the European chapter of the Association for Computational Linguistics (EACL 2006), Trento, Italy, pp. 121–128 (2006)

[21] Mohammad, S., Hirst, G.: Distributional measures of concept-distance: A task-oriented evaluation. In: Proceedings, 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Sydney, Australia (2006)

[22] Mohammad, S., Gurevych, I., Hirst, G., Zesch, T.: Cross-lingual distributional profiles of concepts for measuring semantic distance. In: 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), Prague (2007)

[23] Mohammad, S., Hirst, G., Resnik, P.: TOR, TORMD: Distributional profiles of concepts for unsupervised word sense disambiguation. In: SemEval-2007: 4th International Workshop on Semantic Evaluations, Prague (2007)

[24] Mohammad, S., Dorr, B., Hirst, G.: Computing word-pair antonymy. In: 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), Waikiki, Hawaii (2008)

[25] Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17(1), 21–48 (1991)

[26] Morris, J., Hirst, G.: Non-classical lexical semantic relations. In: Workshop on Computational Lexical Semantics, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Boston, MA (2004): reprinted in: Hanks, P.(editor), Lexicology: Critical Concepts in Linguistics, Routledge (2007)

[27] Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, pp. 241–257 (2003)

[28] Resnik, P.: Using information content to evaluate semantic similarity. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 448–453 (1995)

[29] Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Communications of the ACM 8(10), 627–633 (1965)

[30] Weeds, J.E.: Measures and applications of lexical distributional similarity. PhD thesis, University of Sussex (2003)

[31] Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, pp. 133–138 (1994)

Chapter 5
Collecting Semantic Similarity Ratings to Connect Concepts in Assistive Communication Tools

Sonya Nikolova, Jordan Boyd-Graber, and Christiane Fellbaum

Abstract. To compensate for the common inability of people with lexical production impairments to access and express intended concepts, we make use of models of human semantic memory that build on the notion of semantic similarity and relatedness. Such models, constructed on evidence gained from psycholinguistic experiments, form the basis of a large lexical database, WORDNET. We augment WORDNET with many additional links among words and concepts that are semantically related. Making this densely connected semantic network available to people with anomic aphasia through assistive technologies should enable them to navigate among related words and concepts and retrieve the words that they intend to express.

5.1 Background and Motivation

In this section, we briefly review a debilitating language disorder known as anomic aphasia: the inability to access, retrieve, and produce words. Technology can aid people suffering from the failure to generate the words they wish to express. Our work is motivated by the belief that the effectiveness of such tools can be enhanced by our knowledge of human semantic memory.

Sonya Nikolova · Christiane Fellbaum
Department of Computer Science, Princeton University,
35 Olden Street, Princeton NJ 08540, USA
e-mail: {nikolova,fellbaum}@princeton.edu

Jordan Boyd-Graber
School of Information Studies, South Hornbake, University of Maryland,
College Park MD 20742, USA
e-mail: jbg@umiacs.umd.edu

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 81–93.
springerlink.com

5.1.1 Aphasia

Estimated to affect 1 million people in the United States alone, aphasia is an acquired disorder that impacts an individual's language abilities [23]. It can affect speaking, language comprehension, and writing to varying degrees in any combination in an individual. Rehabilitation can reduce the impairment level, but a significant number of people with aphasia are left with a life-long chronic disability that impacts a wide range of activities and prevents full re-engagement in life. Aphasic individuals often employ different techniques in order to compensate for their inability to communicate; for example, they write notes, gesture, draw, or mimic.

There have been sustained efforts to use technology to help individuals with aphasia communicate. Designing technology that satisfies the needs and expectations of the intended user is a fundamental challenge in the field of human-computer interaction research. This is particularly challenging when designing technology for people with aphasia due to the variability of impairment. The failure of existing assistive communication tools to address the problems arising from the heterogeneity of the user population has stimulated additional research efforts that show it is essential to seek flexible and customizable solutions [5, 16, 19].

Despite efforts to design adaptive assistive tools for elderly and cognitively disabled people, none has proven to be usable by aphasic individuals. Such aids mainly include scheduling and prompting systems that aim to reduce the burden of caregivers [6, 11, 13, 18]. On the other hand, most assistive tools for people with aphasia focus on essential therapeutic efforts and the recovery of basic language function. Thus, they do little to leverage the skills of individuals with some residual communicative ability [3]. There have been relatively few systems for non-therapeutic purposes to be used by less severely affected individuals, such as systems that support daily activities like email or social interactions [8, 16].

In addition, the growing ubiquity of personal electronics, whose form factor could address the stigma attached to communicating with the help of a computer, has not benefited individuals with aphasia. In our experience [5], the weakest link is the ability for users to intuitively and quickly select words. Users, particularly those suffering from anomic aphasia, are confused by arbitrary organization of vocabulary terms and terms absent from the vocabulary. The real difficulty is in providing a flexible system in terms of adding new vocabulary items, adapting to users, and minimizing the complexity of navigating the vocabulary. While we are interested in addressing all of these issues, in this work we focus on vocabulary navigation.

5.1.2 ViVA: Visual Vocabulary for Aphasia

Although initial vocabulary sets can be formed from words frequently needed by the target population, no packaged system has the depth or breadth to meet the needs of every individual. In addition to expressiveness, vocabulary organization and retrieval in existing assistive technology are also problematic. Most of the existing visual vocabularies have a lexical organization scheme based on a simple list of words. The words are organized either in hierarchies which tend to be deep and non-intuitive or in a long list of arbitrary categories. Disorganization and inconsistency result in fruitless scrolling, backtracking, and ultimately frustration. It is important to build an easy to construct and maintain visual vocabulary that rests on a framework of a well-structured computerized vocabulary.

We have developed a multi-modal visual vocabulary that relies on a mixed-initiative design and enables the user to compose sentences and phrases efficiently. The visual vocabulary for aphasia (ViVA) implements a novel approach that organizes the words in the vocabulary in a context-aware network tailored to a user profile that makes finding words faster. ViVA is designed to reorganize and update the vocabulary structure depending on links created between words due to specific user input and system usage.

5.2 The Design of ViVA

In this section, we aim to describe the design of ViVA to show how semantic similarities can play a role in creating a better communication aid for people with aphasia. Our goal for ViVA is for it to be adaptable, able to be customized by the user, in addition to being adaptive, able to dynamically change to better suit the user's past actions and future needs.

The first component, adaptivity, allows the user to add and remove vocabulary items, group them in personalized categories (for example a "Favorites" folder or ideas related to "Family"), enhance words with images and sounds and associate existing phrases and sentences with a concept. In addition to practical concerns of having sufficient vocabulary terms to express the needed concepts, the ability to adapt a system invests in the user a sense of ownership and empowerment. This attachment to the system, brought about by a sense of accomplishment, is an important aspect of the rehabilitation process [1].

We explain the adaptive component with an example. If the user wishes to compose the phrase "I need an appointment with my doctor.", for example, and she searches for `[doctor]` first, the vocabulary network centered on `[doctor]` may look as the one shown in Figure 5.1. The links between the words may exist because the user has previously composed sentences using `[doctor]` and `[medication]` or using `[doctor]` and `[appointment]`. `[hospital]` and `[doctor]`, for example, may be linked because of a prediction based on known word association measures and usage. In addition, the user may be able to find the phrase "Need appointment with doctor" right away if she had already composed it in the past and had linked it deliberately to `[doctor]`.

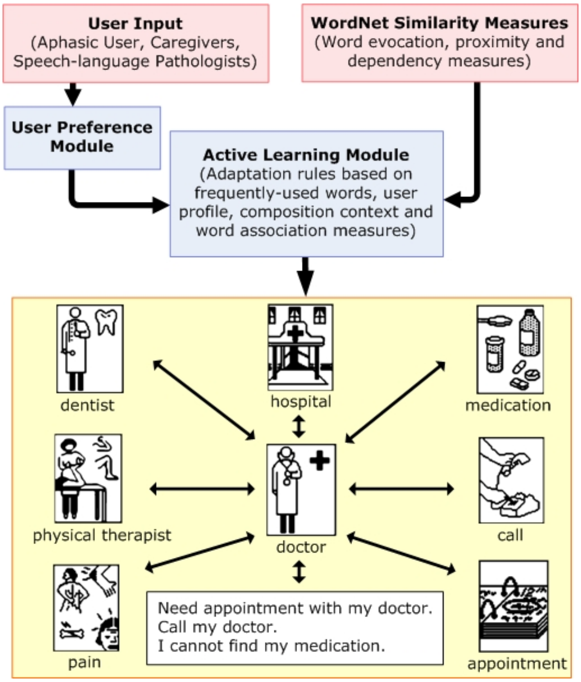

User Input
(Aphasic User, Caregivers, Speech-language Pathologists)
WordNet Similarity Measures
(Word evocation, proximity and dependency measures)
User Preference Module
Active Learning Module
(Adaptation rules based on frequently-used words, user profile, composition context and word association measures)
dentist
hospital
medication
physical therapist
doctor
call
pain
Need appointment with my doctor.
Call my doctor.
I cannot find my medication.
appointment

Fig. 5.1 Schematic of components of a system to assist individuals with aphasia.

This is in contrast to existing systems (e.g., [14], [9]) that have a dichotomy between user-created organization and content and the initial vocabulary. Our goal is to let the user seamlessly add new content and for the organizational structure to change to better suit usage needs. However, we still need an initial organization to allow the user to successfully use ViVA from day one. We derive this scaffold from the body of work investigating how the human brain organizes concepts.

5.3 The Organization of Words

To address the fundamental issues which prevent individuals with aphasia from effectively using communication aids, we appeal to the psychological literature on speakers' "mental lexicon," where words are stored and organized in ways that allow efficient access and retrieval. Our goal is to build a system that can help provide the missing semantic connections in the mental lexicon for sufferers of aphasia. Thus, any successful system must provide an ersatz mental lexicon that users can easily and naturally navigate and explore.

The tip-of-the-tongue (TOT) phenomenon is familiar to every speaker: the temporary inability to retrieve from our mental lexicon a specific word needed to express a given concept. This access failure may be due to a variety of factors, including fatigue and interference from a word that is morphologically or phonologically similar to the target word. People with anomic aphasia can be thought of as suffering from a chronic and severe case of TOT, as they have persistent difficulties accessing and retrieving words that express the concepts they wish to communicate.

Experimental evidence – including evidence from TOT states induced in the laboratory – suggests that words are organized in speakers' mental lexicons by various similarity relations, in particular phonological and semantic similarity. For example, subjects in word association experiments overwhelmingly respond with *husband* to the stimulus *wife* [17]. Semantic priming [22], a robust and powerful tool for the experimental investigation of cognitive processes, relies on the semantic relatedness of the prime and an experimental target: responses to the target are faster when it is related to the prime as in the classic case *doctor–nurse*. Spreading network activation models [7] assume that presenting a prime stimulus word activates the corresponding representation in lexical memory and that this activation spreads to other related nodes, thus facilitating the processing of related target words. The semantic network WORDNET [15, 10] is a large-scale lexical database inspired by network theories of semantic memory that accommodate the spreading activation paradigm among related words and concepts.

5.3.1 WORDNET *and Evocation*

WORDNET has a rich structure connecting its component synonym sets (synsets) to one another. Noun synsets are interlinked by means of hyponymy, the *super–subordinate* or *is-a relation*, as exemplified by the pair `[poodle]-[dog]`.[1] Meronymy, the *part–whole* or *has-a* relation, links noun synsets like `[tire]` and `[car]` [15]. Verb synsets are connected by a variety of lexical entailment pointers that express manner elaborations `[walk]-[limp]`, temporal relations `[compete]-[win]`, and causation `[show]-[see]` [10]. The links among the synsets structure the noun and verb lexicons into hierarchies, with noun hierarchies being considerably deeper than those for verbs.

We aim to exploit the structure of WORDNET to help "find" intended concepts and words by navigating along the paths connecting WORDNET's synsets. However, WORDNET's internal density is insufficient – there are too few connections among the synsets. Boyd-Graber et al. [4] represents an attempt to create thousands of new links that go beyond the relations specified in WordNet. This measure is called "evocation" as it attempts to measure how much one concept brings to mind another.

[1] Throughout this article we will follow the convention of using a single word enclosed in square brackets to denote a synset. Thus, `[dog]` refers not just to the word dog but to the set – when rendered in its entirety – consisting of {`dog, domestic dog, canis familaris`}.

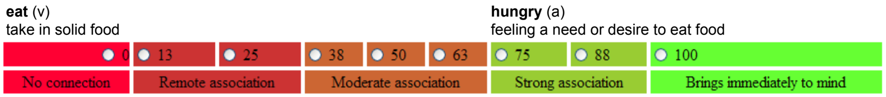
In total, evocation [4] aims to add cross-part-of-speech links, connecting nouns to verbs and adjectives. Such syntagmatic relations allow for connections among entities (expressed by nouns) and their attributes (encoded by adjectives); similarly, events (referred to by verbs) can be linked to the entities with which they are characteristically associated. For example, the intuitive connections among such concepts as `[traffic]`, `[congested]`, and `[stop]` should be encoded in WORDNET. This paper [4] also addressed another shortcoming of WORDNET, namely the absence of weights that indicate the semantic distance between the members of related pairs.

These human judgements of evocation were collected via a laborious, expensive method. Undergraduate students were put through a training and vetting process to consistently rate pairs of synsets through a specially designed interface. Because the pairs of synsets were randomly selected, many of the ratings, as expected, were zero. Although we originally hoped that these initial ratings, collected over the course of year and with a significant outlay of time and money, would allow us to automatically label the rest of WORDNET with directed, weighted links, machine learning techniques could not reliably replicate human ratings. In Section 5.4.1, we propose a method to collect the same valuable empirical similarity ratings using a far less expensive annotation strategy.

5.4 Building the Visual Vocabulary for Aphasia

We selected ViVA's initial vocabulary set such that it is a collection of commonly used words as well as ones relevant to our target population, people who have aphasia. ViVA's core vocabulary was mined from two sources: the "core" WORDNET consisting of frequent and salient words selected for the initial collection of evocation data and the visual vocabulary of an assistive device for people with aphasia created by Lingraphicare [14]. The core WORDNET consists of two sets of 1000 and 5000 salient words. The sets were constructed by collecting the most frequently used words from the British National Corpus [24]. Each word from the resulting list was then assigned its most salient meaning available in WORDNET [4]. We used all synsets from the core 1000 synsets used in our initial evocation study [4], all verbs in Lingraphicare's vocabulary, and all nouns and adjectives in both Lingraphica's vocabulary and the core 5000 synsets.

Lingraphica's multi-modal vocabulary consists of icons that combine text, a pictorial representation of the concept and speech output of the text. We used the pictorial representation to perform a form of coarse disambiguation. For each concept in Lingraphicare's vocabulary, we selected the corresponding concept from WORDNET to create a single, unified representation of the vocabulary.



eat (v)
take in solid food

hungry (a)
feeling a need or desire to eat food

○ 0 ○ 13 ○ 25 ○ 38 ○ 50 ○ 63 ○ 75 ○ 88 ○ 100

No connection | Remote association | Moderate association | Strong association | Brings immediately to mind

Fig. 5.2 Example stimulus for collecting evocation ratings. A user rates 50 pairs in a single sitting.

5.4.1 *Collecting Inexpensive Ratings from Untrained Annotators*

Many natural language processing tasks such as determining evocation require human annotation that is expensive and time-consuming on large scale. Snow et al. [21] demonstrated the potential of Amazon's Mechanical Turk [2] as a method for collecting a large number of inexpensive annotations quickly from a broad pool of human contributors. Their experiment illustrated that labels acquired through Amazon Mechanical Turk (AMT) from non-expert annotators are in high agreement with gold standard annotations from experts. The positive results of their work motivated us to collect evocation ratings to be used in the visual vocabulary for aphasia through a Mechanical Turk experiment described in this section.

5.4.2 *Method*

We used a machine learning algorithm to select the synset pairs to be rated via AMT annotators. We used many of the features found to be predictive of evocation including those based on WORDNET connectivity [12], pointwise mutual information based on words appearing in the same sentence, and context similarity [4]. We duplicated high evocation pairs (having a median rating of greater than 15) to create a high-recall training set, trained a classifier using AdaBoost [20], and then took the subset of all pairs of synsets in our vocabulary labeled as having a high predicted evocation by our learning algorithm. These pairs were the ones selected to be rated via AMT.

We created 200 tasks consisting of 50 pairs each. The design of the template we posted on AMT was closely modeled after the computer program used by Boyd-Graber et al. [4] to collect ratings from undergraduate annotators. Anchor points on a scale from 0 to 100 were available to rate evocation (Figure 5.2). Raters were first presented with the following set of instructions:

1. Rate how much the first word brings to mind the second word using the provided scale.
2. The relationship between the two words is not necessarily symmetrical. For example, "dollar" may evoke "green" more than the reverse.
3. Pay attention to the definition of the words given on the second line; words can have more than one meaning. For example "dog" (the animal) would not bring to mind "bun" (the piece of bread you serve with a hot dog).

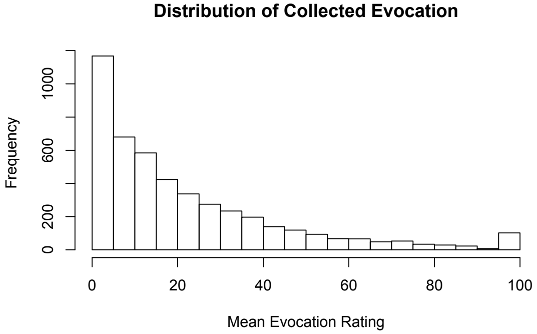


Fig. 5.3 The distribution of the mean of the evocation pairs collected. The extra bump at 100 is because same synset check was placed in each task to ensure annotator reliability.

4. The letter in parenthesis signifies whether the word is *a*: an adjective, *n*: a noun or *v*: a verb.
5. Don't use information from your personal life. For example, if you had a dog named "bog" you personally would associate "bog" and "dog," but the average person wouldn't.
6. Don't use the spelling of words to make your decisions. For example, even though "bog" and "dog" rhyme, they are not associated.
7. We cannot offer you a big reward for your time, but we greatly appreciate your sincere effort. There are a few pairs with known average ratings embedded in the task. If your ratings for those pairs do not fall in generously set acceptance bounds, we will have to reject your responses.

The last instruction was included to forewarn annotators that sloppy contributions such as clicking all zeros will not be rewarded. We embedded five checks, unknown to the annotators, in each task which were later used to determine the validity of the gathered results. Annotators were paid $0.07 to complete a task.

5.5 Results

We collected 2990 completed tasks in a period of ten days. The average time to complete a task was 4.5 minutes, resulting in an average pay of $0.92 per hour. To ensure the quality of the ratings and a consistency with previous results, we used embedded checks to decide which submitted tasks were valid. The ratings for four of those checks were collected from the dataset provided by Boyd-Graber et al. [4]. The fifth check required annotators to rank a pair consisting of the same synset, for example `[help]` and `[help]`.

Table 5.1 Correlation of the mean and median against evocation annotations collected by trained undergraduate annotators.

Filtering Method	Correlation with Mean	Correlation with Median	Number Ratings
All Checks	0.604	0.563	14850
Most Checks	0.529	0.484	23700
Some Checks	0.355	0.279	24750

Table 5.2 Examples of mean evocation ratings given three different methods to ensure rater reliability. For comparison, evocation ratings from trained undergraduates are also shown.

Filtering Method			Trained	Synset 1	Synset 2
All	Most	Some	Undergraduates		
50	10	61	88	trust.v.01	responsible.a.01
39	44	41	44	surgeon.n.01	responsible.a.01
25	18	22	42	deservingness.n.01	exceed.v.02
31	31	28	33	philosopher.n.01	convert.v.03
29	30	30	20	television_receiver.n.01	performance.n.02
46	57	62	19	log.n.01	leaf.n.01
12	12	14	18	subject.n.06	check.v.22
34	33	31	16	diligence.n.02	craft.n.04
25	20	27	16	abundant.a.01	harmony.n.02
21	10	14	1	category.n.02	beginning.n.03
23	19	18	0	eyelid.n.01	wrist.n.01
25	28	26	0	reason.n.02	reference_point.n.01
4	5	9	0	spread.n.05	pill.n.02

We ran three different reliability tests depending on the number of checks we wanted satisfied. If the annotator's rating for the fifth check was 100 and a number of the remaining checks were met within certain acceptance bounds, the annotations were considered valid. The acceptance bounds were defined as follows. As in the task, the scale of 0 to 100 was split into 5 intervals, [0–9], [10–29], [30–69], [70–89], [90–100]. If an annotator's rating fell within the same interval as the corresponding check or in the immediately lower or higher intervals, the rating was considered valid. The first reliability test required **all** checks to be met. For this set, 40.2% of the pairs were rated as having moderate or no association and 3.5% fell in the category immediately brings to mind (see Figure 5.3).

The second reliability test required **most**, three or more, checks to be met in addition to satisfying the complete-evocation check. The final and most relaxed reliability test required **some**, two or more, checks to be met in addition to the complete-evocation check. Table 5.1 shows the number of synset pair ratings for each of the reliability levels, and Table 5.2 has explicit examples of mean evocation ratings for the three levels.

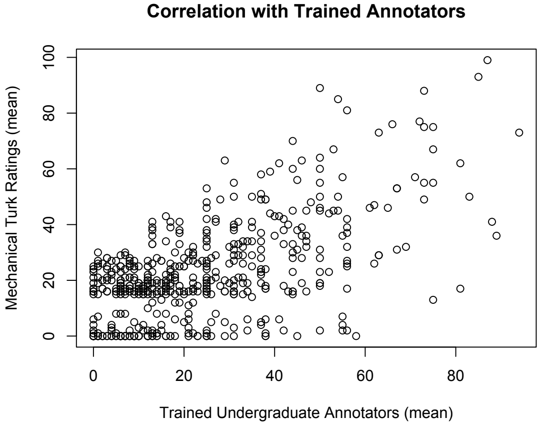


Fig. 5.4 Ratings from untrained annotators on the web correlated well (0.604) with those collected by trained undergraduate annotators.

Finally, Table 5.1 shows mean and median correlation of the three reliability sets against the ratings provided by undergraduate students in [4]. As expected, the results get noisier when less strict checks are applied. The set of synsets where all checks were met results in the highest correlation to the original evocation data. While it is not very high, it is sufficient to show that with good quality control, gathering ratings through AMT was a valid approach. While AMT annotators seem to rate on average evocation lower than the trained annotators, as seen from Figure 5.4, the inter-annotator agreement in the most reliable set and the original evocation set are comparable.

5.6 Discussion

While the results may appear less compelling than one might have expected, it is important to bear in mind the difficulty of the task. First, the nature of the task was such that we asked the participants to actively *produce* a rating, rather than to agree or disagree with a pre-set judgment or to select one from a few pre-defined options. Second, the ratings were to be expressed on a scale from 0 to 100, thus allowing for – and in fact, encouraging – very subtle judgments that permitted significant disagreement. Third, while we controlled for intra-rater reliability, we did not know who our raters were in terms of educational level, literacy, and familiarity with the words and concepts that were presented. Indeed, we had no way to ascertain that the raters were native or near-native speakers of English. Finally, the raters might have received insufficient training given the cognitive demands of the task.

The results must be compared to those obtained in the carefully controlled study reported by Boyd-Graber et al. [4]. At the outset of that experiment, it was unknown whether any reasonable reliability could be obtained at all, as we were well aware of the difficulty of the task, for which no precedent existed, and we considered the results encouraging. Our raters came from a small, homogeneous pool – Princeton undergraduate students – whose identity we knew and whom we trained carefully and with personal feedback. In light of the different methods of data collection, the results of the current study are comparable. The inherent noise in the evocation data reflects idiosyncrasies in world knowledge; only by accepting this reality and incorporating it into assistive technologies can we hope to build devices that can truly help a heterogenous target population.

Even though our strictest reliability test invalidated half of the collected data, using AMT to gather evocation is still more efficient and economical compared to using trained annotators. This collection of reliable evocation ratings adds on to the scaffolding of our assistive vocabulary by providing meaningful links between words. Such links will compensate for impaired access to the user's "mental lexicon" and assist her in communicating. A network of words whose organization reflects human semantic memory has the potential to help users with anomic aphasia navigate the vocabulary more naturally and thus find what they are trying to express faster.

Acknowledgments

This work was partially supported by a 2007 Intelligent Systems for Assisted Cognition Grant by Microsoft Research and by the Kimberly and Frank H. Moss 71 Research Innovation Fund of the Princeton School of Engineering and Applied Science. We thank Perry Cook for his feedback and Lingraphicare Inc. for providing us with access to their icon library. The results from our experiments are available for download at:
http://wordnet.cs.princeton.edu/downloads.html.

References

[1] Allen, M., McGrenere, J., Purves, B.: The design and field evaluation of phototalk: a digital image communication application for people. In: Assets 2007: Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 187–194. ACM, New York (2007), http://doi.acm.org/10.1145/1296843.1296876

[2] Amazoncom, Inc., Amazon mechanical turk (2010), https://www.mturk.com (last accessed December 8, 2010)

[3] Beukelman, D.R., Mirenda, P.: Augmentative and alternative Communication: Management of Severe Communication Disorders in Children and Adults. Brookes Publishing Company, Leiden (1998)

[4] Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R.: Adding dense, weighted, connections to WordNet. In: Sojka, P., Choi, K.S., Fellbaum, C., Vossen, P. (eds.) Proc. Global WordNet Conference 2006, Global WordNet Association, Masaryk University in Brno, Brno, Czech Republic, pp. 29–35 (2006)

[5] Boyd-Graber, J.L., Nikolova, S.S., Moffatt, K.A., Kin, K.C., Lee, J.Y., Mackey, L.W., Tremaine, M.M., Klawe, M.M.: Participatory design with proxies: Developing a desktop-PDA system to support people with aphasia. In: Proc. CHI 2006, pp. 151–160. ACM Press, New York (2006), doi: http://doi.acm.org/10.1145/1124772.1124797

[6] Carmien, S.: MAPS: PDA scaffolding for independence for persons with cognitive impairments. In: Human-Computer Interaction Consortium (2002)

[7] Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychological Review 82(6), 407–428 (1975), http://psycnet.apa.org/index.cfm?fa=search.displayRecord&#38;uid=1976-03421-001

[8] Daeman, E., Dadlani, P., Du, J., Li, Y., Erik-Paker, P., Martens, J., Ruyter, B.D.: Designing a free style, indirect, and interactive storytelling application for people with aphasia. In: INTERACT, pp. 221–234 (2007)

[9] Dynavox, M.-J.: Dynavox, http://www.dynavoxtech.com/ (last accessed Dec 08, 2010)

[10] Fellbaum, C.: A semantic network of English verbs. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)

[11] Haigh, K., Kiff, L., Ho, G.: The Independent LifeStyle AssistantTM (I.L.S.A.): Lessons Learned. Assistive Technology 18, 87–106 (2006)

[12] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings on International Conference on Research in Computational Linguistics, Taiwan (1997)

[13] Levinson, R.: PEAT: The planning and execution assistant and trainer. Journal of Head Trauma Rehabilitation 12(2), 769–775 (1997)

[14] Lingraphicare Inc. Lingraphica, http://www.aphasia.com/ (last accessed December 8, 2010)

[15] Miller, G.A.: Nouns in WordNet: A lexical inheritance system. International Journal of Lexicography 3(4), 245–264 (1990), http://ijl.oxfordjournals.org/cgi/reprint/3/4/245.pdf; doi: 10.1093/ijl/3.4.245

[16] Moffatt, K., McGrenere, J., Purves, B., Klawe, M.: The participatory design of a sound and image enhanced daily planner for people with aphasia. In: Proc. CHI 2004, pp. 407–414. ACM Press, New York (2004), doi: http://doi.acm.org/10.1145/985692.985744

[17] Moss, H., Older, L.: Birkbeck Word Association Norms. Psychology Press, San Diego (1996)

[18] Pollack, M.E., Brown, L., Colbry, D., McCarthy, C.E., Orosz, C., Peintner, B., Ramakrishnan, S., Tsamardinos, I.: Autominder: An intelligent cognitive orthotic system for people with memory impairment. Robotics and Autonomous Systems 44(3-4), 273–282 (2003)

[19] van de Sandt-Koenderman, M.M., Wiegers, J., Hardy, P.: A computerised communication aid for people with aphasia. Disability Rehabilitation 27(9), 529–533 (2005)

[20] Schapire, R.E.: The boosting approach to machine learning: An overview. In: Denison, D.D., Hansen, M.H., Holmes, C., Mallick, B., Yu, B. (eds.) Nonlinear Estimation and Classification. Springer, Heidelberg (2003)

[21] O'Connor, S.R., Jurafsky D, Ng A (2008), Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. Proceedings of EMNLP 2008, http://aclweb.org/anthology-new/D/D08/D08-1027.pdf

[22] Swinney, D.: Lexical access during sentence comprehension: (Re)consideration of context effects. Journal of Verbal Learning and Verbal Behavior 18, 645–659 (1979)

[23] The National Aphasia Association Aphasia: The facts, http://www.aphasia.org (last accessed December 8, 2010)

[24] University of Oxford, British National Corpus (2006), http://www.natcorp.ox.ac.uk/, http://www.natcorp.ox.ac.uk/

Part III
From Textual Data to Ontologies, from Ontologies to Textual Data

Chapter 6
An Introduction to Hybrid Semantics: The Role of Cognition in Semantic Resources

Alessandro Oltramari

Abstract. This paper focuses on the problem of opening semantic technologies to a comprehensive cognitive modeling of information: assuming an unilateral focus on symbolic ‘representations’, in fact, spoils human knowledge processes of their peculiar ‘embodied’ features. We claim, conversely, that the mutual interlinks between static and dynamic cognitive structures should be adopted in semantic technologies to improve existing knowledge resources. In particular, in the process of ontology creation, a preliminary analysis of knowledge formation and cognitive structuring of conceptualization becomes a necessary module of semantic technologies.

6.1 Introduction

In cognitive science, knowledge is conceived as the main outcome of the process of understanding (see for example [21] and [23]): by interacting with the environment, intelligent agents are able to interpret and represent world situations, suitably acting to preserve themselves and pursue specific goals. Representing knowledge is a necessary step for communication; but knowledge can be represented in so far that world phenomena are previously *presented* to humans, namely structured and perceived as unique experiences of the subject: one can hear the sound of a boiling teapot and express a sentence about that; trivially, the linguistic rendering is different from the personal hearing event, which is called ‘act of presentation’: in the former humans exploit natural language to transmit the conceptual content of experience; in the latter, a concrete sense-driven process supplies perceptual information to the subject. What is not trivial, instead, is the analysis of the relations holding between ‘knowledge representations’ and ‘perceptual presentations’, namely what we refer to here as ‘*cognitive matrix* of static and dynamic dimensions of human

Alessandro Oltramari
Department of Psychology, Carnegie Mellon University, Pittsburgh, USA
e-mail: aoltrama@andrew.cmu.edu

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 97–109.
springerlink.com

cognition'. *Knowledge contents* are neither simple products of abstraction nor bare perception-driven units: they genuinely correspond to an interwoven combination (so called *cognitive matrix*) of the two.

These arguments are particularly important for the field of human-computer interaction, with the substantial difference that here conceptual contents have to be 'realized' in a machine-readable format: to ignore the embodied nature of meaning when switching from natural to computational languages would be to discard an essential part of information that is needed to face the problem of semantic ambiguity in *knowledge technologies*. If static and dynamic dimensions of cognition constitute meaning, as we claim in the current work, then they need to be also encoded in those information systems dealing with world knowledge description. The cognitive adequacy of knowledge technologies allows humans and machines to improve the mutual access to information: in this work, our study of the process of understanding aims at integrating the proper vision underlying the field of knowledge representation, in particular concerning so-called 'ontological level'. In this perspective we outline an *hybrid approach to cognitive* science and technology to overcome the delineated problems and improve information systems accordingly. Quoting Negroponte , the 'quality of the (digital) world we live by depends on the quality of information we access to, that needs to be selected, filtered, and organized' [22].

6.2 Statics and Dynamics of Cognition

In this section we introduce the core aspects of a **hybrid perspective on cognition**, where both static and dynamic structures of the mind are involved. We define as **statics of cognition** the study of mental representations from a formal or symbolic viewpoint. The usefulness of the symbolic approach for computational analysis is well-recognized: nevertheless, assuming mental representations to be *genetically* symbolic could be dangerous and misleading. In this context, **dynamics of cognition** refers to a level of analysis where mental phenomena and their perceptual basis become the real objective of the inquiry ([1], p.2).

The core principle grounding statics and dynamics of cognition consists in the seminal distinction between **presentation** (*Vorstellung*) and **representation** (*Darstellung*): the concept of *Darstellung* appeared in Kant's *Kritik der Urteilskraft* [16] and, roughly speaking, designates the communicable part of a mental representation/object, its *exhibition*.

In order to eschew any terminological problem, we consider henceforth **mental presentation** (or just **presentation**) and **conceptual representation** (or **mental object**) as kinds of **mental entities**. Conceptual representations, as the adjective itself suggests, refer to concepts (e.g. 'composer'), namely organizations of individuals (e.g. 'Bach','Beethoven','Verdi') according to the similar properties they share ('writing music').

'Buying' the notion of conceptual representation can be advantageous only if we resist the temptation of reducing all mental phenomena to it. We are not denying here the use of formal languages to model the dynamics of cognition, that is of course an important issue related to the implementation of information systems and machine-understandable resources. Nevertheless, the dynamic forms of mental presentations have to be necessarily analysed at first by suitable instruments, in order to bring out their structural elements and relations, which cannot be merely interpreted as 'abstract symbols managed by formal rules'. Among those instruments we can find **cognitive semantics**, according to which language is not interpreted as an autonomous and self-referent domain but consists of features that are directly dependent on perception. Far from willing to deepen a topic that is intimately related to history of philosophy and psychology, we need at least to trace back the notion of 'presentation' to its peculiar origin. We recall, in particular, Franz Brentano's metaphysical conception [5]. When something from the environment becomes present to the subject, there is an *act of presentation* – a psychic phenomenon – that reveals the manifestation of that object to the subject. Even if the act of presentation depends on the interaction of the subject with the environment, and ultimately to the peculiar features of the environment itself as stated one century later in the 'ecological approach' [11], there is also a massive information 'imposed' by the subjective act to the presented object, a sort of 'filtering' through cognitive structures. Presentations correspond to psychological events directed to objects, namely the objects presented by the act: hearing the sound of a cascade points to the phenomenal aspect of an object i.e. – in Brentano's terms – the **intentional object** of the hearing that differs from the characteristic wavelength of the sound emission produced by the falling water.

Summarizing, the backbone of our hybrid perspective on cognition has two core keystones:

1. the notion of **mental presentation** studied by cognitive semantics, according to which language is conceived as a complex system directly dependent on perceptual structures;
2. the notion of **conceptual representation** analysed by means of formal semantics, which assumes language to be autonomous.

Two concepts of 'meaning' are clearly into play here: on one side, meaning is "the product of mental activity on the part of physically embodied, socio-culturally grounded human minds" (see [18], p. 26); on the other side, formal semantics characterizes meaning by pointing to the truth-conditions of a sentence in a language (e.g. Montague semantics – see [8]). This double aspect of cognition constitute the basis of the 'cognitive matrix'. Following the former approach, the connection between language and the world does not take on a direct shape but displays to be mediated by perceptive structures: these structures, suitably indicated as **conceptual schemas** or just **schemas**, capture the dynamics of the cognitive relations holding within the elements of a presentation, their mutual position, dependence, salience and ambiguity, projecting them into language ([3], p. 9). Although the notion of schema has not been deeply inspected by any of the proponents of cognitive semantics ([2], p. 16), its role in the analysis of the cognitive level remains unquestionable: Langacker, for example, refers to it as 'a mapping of a structural complex in a coarse grid', e.g. the concept of TOOL is a schema for notions like HAMMER, SCREWDRIVER, SAW, which specify peculiar details the schema abstracts from specific instances or content units – [19], p.103).

6.3 From Conceptualization to Specification: The Ontological Level

The overall framework of the *Cognitive Matrix* refers to the specific problem of defining what is a *representation*: in particular, here we are considering how the above-mentioned notions of **presentation** and **mental representation** affect the present topic, even helping to detect dangerous misunderstandings in the basic vocabulary widely in use. First of all, we need to make a step back to literature.

Many different definitions of 'conceptualization' and 'ontology' are available[1], even though a common agreement has been reached today on the 'usage' of both for capturing shared knowledge in software systems. In general, "ontologies are defined as a formal specification of a *shared* conceptualization"[2]. Guarino gave a finer characterization: a 'conceptualization' is a language-independent view of the world, a set of conceptual relations defined on a domain space[3]; besides, an ontology is a language-dependent cognitive artifact, committed to *a certain* conceptualization of the world by means of a given language (see [14] for formal details)[4]. As words in italics suggest, the main disagreement here consists in the fact that the former definition talk about shared conceptualizations, while latter doesn't ask for such a strong requirement. We endorse here Guarino's definition for the following reason: every conceptualization is bound to a single agent, namely it is a *mental product* which stands for the view of the world adopted by that agent; it is by means of ontologies, which are language-specifications of those mental products, that heterogeneous agents (humans, artificial or hybrid groups of them) can assess if *a certain* conceptualization is *shared* or not and choose if it is worthwhile to negotiate meaning or not. Borst and Gruber's definition, which is an example of the general position adopted by AI, database and information systems community, overturns the argument and misleads the inner sense of the problem: since we can't directly 'read' agent's minds, how could we know that different agents share the same conceptualization, without a linguistic projection of such conceptualization? The exclusive entryway to concepts is by language; if the layman normally uses *natural language*,

[1] See [12] for a short review.

[2] This definition by Borst [4] is a refinement of Gruber's one [13].

[3] Given a domain of entities, a domain space is a set of possible states of affairs of that domain (see also Kripke's notion of *possible worlds* [17]).

[4] Guarino and Giaretta, in their 1995 seminal paper [15] distinguish between 'Ontology' as a discipline (with the capital 'o') from 'ontologies' as *engineering artifacts*, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words.

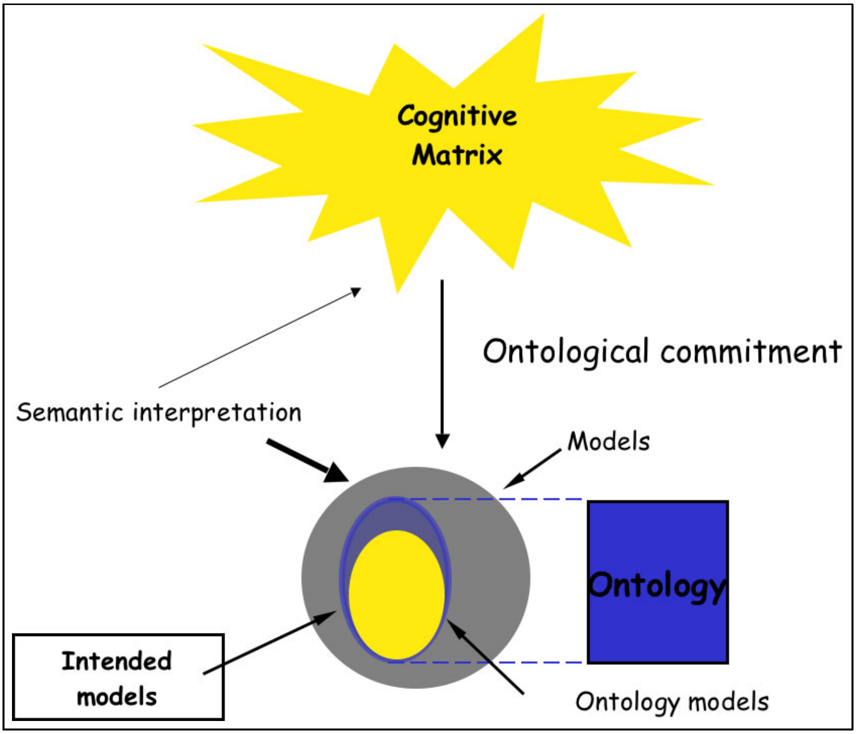


Fig. 6.1 Guarino's revisited.

societies of hybrid agents composed by computer, robot and humans, need a formal machine-understandable language.

Apart from the level of complexity and explicitness, what is crucial is that ontologies, as language-dependent specifications of conceptualizations, are the basis of communication, the bridge through which common understanding is possible.

Figure 6.1, based on Guarino's original one[5], depicts the above-mentioned general scenario: note that we have substituted here the notion of **conceptualization** with the notion of **Cognitive Matrix** to stress the fact that ontological models originate from the (semantic) interpretation of the network of static and dynamic structures underlying the cognitive level of human mind, even though they don't completely reflect it.

Definition. The intended models of a logical language reflect its commitment to a *Cognitive Matrix*; an ontology indirectly reflects this commitment (and the underlying *Cognitive Matrix*) by approximating this set of intended models'.

Let's resume the situation. Different agents (independently from their biological or artificial nature) have different cognitive matrices: they need ontologies to bargain meanings and communicate in a common environment. Ontologies enable the

[5] See, for example, [14].

organization of information contents and “provide agents with the shared knowledge that they can use to communicate and work together” ([26], p.343).

6.4 Building ‘Hybrid’ Semantic Technologies

If static and dynamic structures of cognition form the essential features of linguistic meanings – as we showed above – the comprehensiveness and efficiency of semantic technologies in terms of human-computer interaction actually depends on an adequate treatment of those aspects in the design and implementation phases of different computational resources. As testified by research and industrial projects, ‘knowledge technologies’ are bound to a life-cycle constituted by acquisition, retrieval, modeling, reuse, publishing and maintenance of knowledge. In this context, we introduce the notion of *knowledge matrix* as *the machine-understandable manifold structure* emerging from modeling knowledge according to the *cognitive matrix* perspective. This notion reflects the *cognitive matrix* meta-model from an implementational standpoint: interfacing ontologies and computational lexicons has to be seen as a new route of the *information space* across which hybrid agents negotiate **knowledge contents** and exchange linguistic and ontological aspects of knowledge[6].

The **ontological layer** has been recognized in recent years as a fundamental one in agent communication [10]: besides **the protocol layer**, where the syntax of the communication language is specified[7], the ontological layer formally defines the concepts needed to exchange messages[8]. Furthermore, when building human-computer interaction resources, accessing to ontological concepts through natural language becomes a necessary requirement: in this context, the complementary role played by computational lexicons is very important, also considering the function of natural language as common ‘vehicles’ conceptual schemas. Computational lexicons, whose aim is to make lexical-content machine-understandable, constitute a fundamental component of the *Knowledge Matrix*. As Lenci, Calzolari and Zampolli pointed out, “ontologies represent an important bridge between knowledge representation and computational lexical semantics, and actually form a *continuum* with semantic lexicons”[9][20]. The most relevant areas of interest in this context are Semantic Web and Human-Language Technologies (HLT): they converge in the task of providing the semantic description of content, although concerning two different

[6] In Computer Science, this particular notion of ‘knowledge’ would correspond to T-box (terminological) statements in a knowledge base (i.e. student is a subclass of person). The assertion component (A-box), namely factual knowledge associated to terminology (i.e. John is a person), is not central in the present context.

[7] KQML (Knowledge Query and Manipulation Language) is one of the most well known language protocols in the field [9].

[8] The granularity of these concepts may change from high level to domain-dependent: for instance, in e-learning systems an agent may ask about the general notion of ‘learner’ and also about the knowledge of a specific topic [6]: the ontology must provide both concepts.

[9] Here ‘semantic lexicon’ and ‘computational lexicon’ have been used as synonyms.

dimensions, the conceptual and lexical one. Implemented ontologies and computational lexicons aim at digging out the basic elements of a given semantic space (domain-dependent or general), characterizing the different relations holding among them. As we claim in [24], the interface between ontologies and lexical resources, provides the ground for approximating and implementing information according to the *cognitive matrix* components. Cognition has been depicted as a complex network of static and dynamic structures (the *cognitive matrix*) that filters information from the environment to the human mind. Dynamics of cognition concerns the acts of presentation and their intentional objects; statics of cognition, on the other side, deals with peculiar conceptual representations which emerge from singling out common properties from those intentional objects. Making machine-understandable this grounding manifold can foster the creation of information systems according to suitable cognitive principles and distinctions.

6.5 The Project of a Collaborative Hybrid Semantic Resource

Several automatic methodologies and techniques are employed to build ontologies from variable-scaled data sources such as annotated corpora, unstructured texts, terminologies, lexical databases, etc. Although, at a first stage, it is important to adopt automatic tools to "skim off" knowledge from heterogeneous data sources, the essential limit of ontology learning lays in the quality of the obtained resources: in order to make extracted ontologies robust and consistent with cognitive models of knowledge, human-based conceptual analysis and possible re-organization of the overall ontological structure are needed, together with suitable comparisons with pre-existing ontologies. Therefore, human intervention over the extracted conceptual and relational distinctions becomes a mandatory requirement for developing well-founded re-usable ontologies. By separating the linguistic layer from the ontology, users are allowed to manifest their knowledge in a free, incremental, natural, collaborative and potentially conflicting way. As Wikipedia demonstrates, collaborative projects produce huge amount of knowledge, which is continuously updated, amended and extended by wiki-editors. We think that by applying a crowdsourcing approach to the collection of human common-sense and linguistic knowledge can also fit the Semantic Web paradigm. Following the cognitive perspective presented in this paper, we introduce here the general features of TMEO, a tutoring methodology to support semi-automatic ontology learning by means of interactive enrichment of ontologies (both from the lexical and the ontological levels). Originally conceived as a Q/A system for guiding humans in the elicitation of ontological knowledge at the common sense level, TMEO's model is being also applied to the industrial and business scenarios of Italian FIRB 2006 project "TOCAI.it" (Knowledge-oriented technologies for enterprise aggregation in Internet).

TMEO is based on a simplified OWL-lite version of DOLCE (called DOLCE-spray, characterized only by very general ontological distinctions 'spread over' the original DOLCE[10]: it corresponds to a formal model where upper level categories (endurant, perdurant, quality, and abstract) and general relations (part-of, participation, dependence, etc.) are represented in a standard logic language. This ontology has been explicitly developed in order to meet some core cognitive and linguistic features of common sense knowledge. Although presenting DOLCE-spray in details is out of scope in this paper, we can say that the basic layer of DOLCE-spray originates from the distinction between ENDURANT and PERDURANT, the former class including those entities that are in time (like tables, cars, trees, persons, etc.), the latter corresponding to entities that happen in time (such as conversations, wars, dives, concerts, weddings, thoughts, et cetera). According to common sense, ENDURANT would roughly point to the everyday category of PHYSICAL OBJECT while PERDURANT class would refer to EVENT (see Figure 6.2).

TMEO is inspired by Plato's dialectic (Socrates' methodology to drive his disciples to true knowledge, posing questions and arguing on answers): it exploits some suitable ontological properties for posing questions to users in support of domain independent or dependent knowledge modeling. TMEO is an interactive Q/A system based on general distinctions embedded in DOLCE-spray.

Consider the case in which a given user is asked to classify the term (shoe), whose WordNet gloss is "footwear shaped to fit the foot (below the ankle) with a flexible upper of leather or plastic and a sole and heel of heavier material"[11]. After initializing TMEO wizard, the HCI interface will put the user through a series of intuitive conceptual questions – driven by the underlying DOLCE-spray ontological model – in order to make explicit the intended meaning of the term. The following sequence reflects an experimental trial made with multiple users.

- TMEO-Wizard: Can you touch or see or smell or taste or hear or feel **a shoe**?
  User: Yes
- TMEO-Wizard: Can you count or enumerate **shoe**s?
  User: Yes
- TMEO-Wizard: Can you say that "a **shoe** is happening or occurring"?
  User: No
- TMEO-Wizard: Can you say that a **shoe** is a product of human activity (no matter whether hand-made or with the help of suitable tools and machines)?
  User: Yes
- TMEO-Wizard: **shoe** in the sense of "footwear shaped to fit the foot (below the ankle) with a flexible upper of leather or plastic and a sole and heel of heavier material" has been classified as a kind of ARTIFACT.
- TMEO-Wizard: What are **shoe**s commonly used for?
  User: to walk

[10] Descriptive Ontology for Linguistic and Cognitive Engineering (see http://www.loa-cnr.it/DOLCE.html)
consists of about 40 concepts, 100 relations and 80 axioms

[11] See http://wordnetweb.princeton.edu/perl/webwn

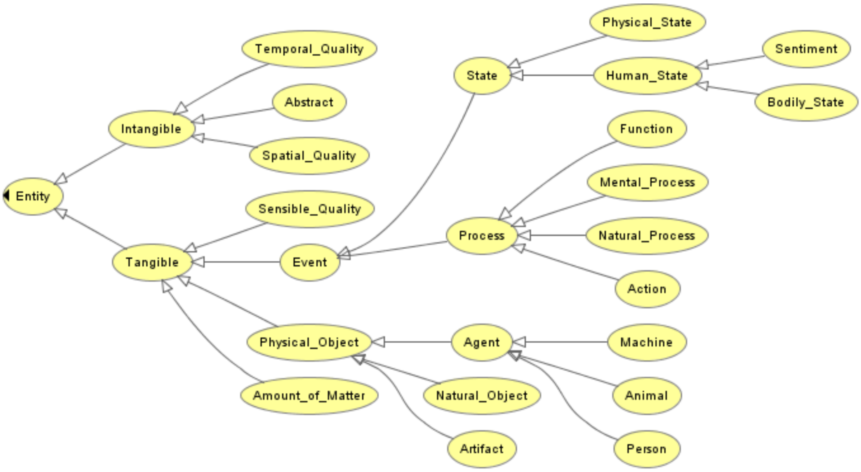


Fig. 6.2 The DOLCE-spray general schema.

- TMEO-Wizard: Select the correct sense of **to walk**: [s_1 - s_2 - s_3 - s_n][12].
  User: s_1
- TMEO-Wizard: **to walk** in the sense s_1 has been classified as the typical FUNCTION of **shoe**s.

Here the algorithm drives the user through tracing the following path of knowledge: *shoes as* ARTIFACT *have the common* FUNCTION *of being used in walking events*. As the above-mentioned scenario suggests, TMEO methodology may therefore be adopted not only in the unilateral classification of a given term ('shoe') but also in making related lexical items explicit. This kind of relatedness between terms actually unwraps the inter-categorial relation(s) holding between the corresponding ontological categories. Indeed, from the ontological viewpoint we can say that there is a relation of *Participation* holding between the category ARTIFACT (which is a kind of PHYSICAL OBJECT) and FUNCTION, which is conceptualized in DOLCE-spray as a kind of PROCESS[13].

TMEO has been implemented as a Java finite state machine (FSM): in general, the elaboration process of a FSM begins from one of the states (called a 'start state'), goes through transitions depending on input to different states and can end in any of those available (only the subset of so-called 'accept states' mark a successful flow of operation). In the architectural framework of TMEO, the 'start state' is equivalent to the top-most category ENTITY, the 'transitional states' correspond to disjunctions within ontological categories and 'accept states' are played by to the most specific

[12] For the sake of readability, we don't go through the basic senses of the verb 'to walk', also assuming that s_1 is adequately selected by the user.

[13] Note that we may wish to distinguish descriptions of functions from actual ones, namely those functions which are performed at a certain time by a given object. In the above example we simplify this distinction only focusing on the latter case.

categories of the model, i.e. 'leaves' of the relative taxonomical structure. In this context, queries represent the conceptual means to transition: this means that, when the user answers to questions like the ones presented in the above-mentioned example (e.g. "can you count or enumerate shoes?"), the FSM shifts from one state to another according to answers driven by boolean logic[14]). If no more questions are posited to the user, then this implies that the operations have reached one of the available final 'accept state', corresponding to the level where ontological categories don't have further specializations (no transitions are left). TMEO human language interface is very simple and comes in the form of a window where *yes/no* options are presented together with the step-by-step questions: Figure 6.3 shows an example in Italian for the word 'cane' (=dog), where the Wizard asks whether one can perceive *cane* with the five senses or not. At the end of any single process of enrichment, the system automatically stores the new concept as an OWL class in the knowledge base under the category of DOLCE-spray suitably selected by the user (e.g. in this sense,'shoe' and 'dog' become respectively a subclass of ARTEFACT and of 'animal'). Future work on TMEO aims at extending the coverage of the model, adding new 'transitional states' and 'accept states'. We discovered that users, in fact, have an high degree of confidence and precision in classifying the concept referring to the physical realm, while they face several problems in distinguishing abstract notions like 'number', 'thought', 'beauty', 'duration', etc. [7]: future releases of TMEO will have to be improved both conceptually and heuristically, in this direction.

As the reader should have realized so far, one of the major advantages of this approach consists in its flexibility: it is relatively easy in terms of human-effort to apply and customize TMEO to specific domains, given some preliminary conceptual analysis of the entities and relations at play, or to add some finer-grained distinctions to enhance the degree of precision of the enrichment (e.g., distinguishing 'artistic artefact' from 'mechanical artefacts', 'collective actions' from actions depending on a single agent, and so on and so forth). From the software engineering perspective, the scalability of the system is bound to the maintenance of the knowledge base: that is, any change to the conceptual structures of TMEO can be made at the level of the ontology, directly modifying the OWL model (for instance, in the Protégé frame-based platform[15]): in fact, the actual Q/A dialog system retrieves information from the ontology and eventually updates questions (stored in the form of RDF "annotation properties") and categories accordingly.

In conclusion we should also mention that TMEO has been deployed in the TasLab Project[16], concerning the realization of a semantic portal for fostering territorial ICT innovation, including the use of domain ontologies and thesauri (e.g., Eurovoc[17]), indexing and semantic search techniques[18]. We are also planning to

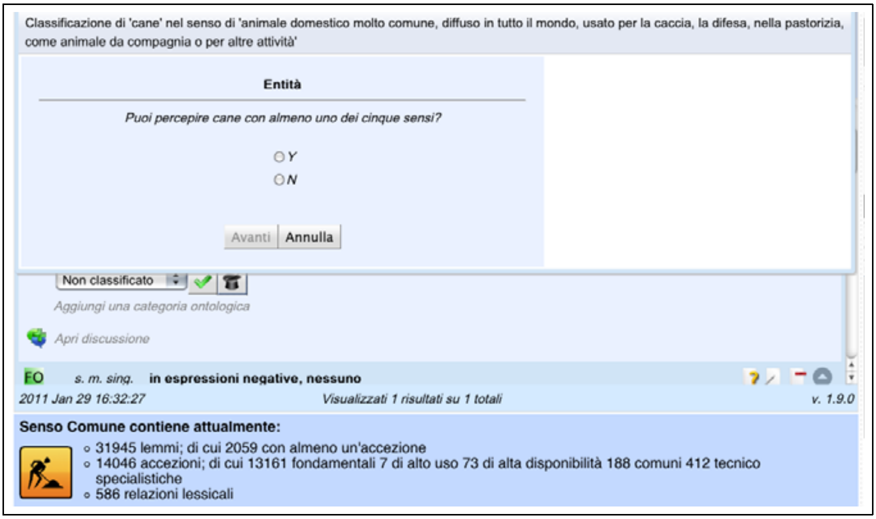
[14] Uncertainty will be included only in future releases of the TMEO system.

[15] http://protege.stanford.edu/

[16] http://www.taslab.eu/

[17] http://europa.eu/eurovoc/

[18] See [25].



Classificazione di 'cane' nel senso di 'animale domestico molto comune, diffuso in tutto il mondo, usato per la caccia, la difesa, nella pastorizia, come animale da compagnia o per altre attività'

Entità

Puoi percepire cane con almeno uno dei cinque sensi?

○ *Y*
○ *N*

FO *s. m. sing.* **in espressioni negative, nessuno**

2011 Jan 29 16:32:27 *Visualizzati 1 risultati su 1 totali* *v. 1.9.0*

Senso Comune contiene attualmente:

- 31945 lemmi; di cui 2059 con almeno un'accezione
- 14046 accezioni; di cui 13161 fondamentali 7 di alto uso 73 di alta disponibilità 188 comuni 412 tecnico specialistiche
- 586 relazioni lessicali

Fig. 6.3 The TMEO Wizard.

adopt TMEO in the DARPA project 'Mind's Eye", which deals with recognition and classification of basic actions[19].

6.6 Conclusion

In a world where artificial agents are built to recognize, adapt and mimic human features in interactive scenarios, it is a major task for cognitive scientists, and in particular for knowledge engineers, to acknowledge the cognitive dimensions of information in their systems: knowledge is externally accessible, but internally shaped. To hold the multifariousness of the knowledge world in building information systems, leaving aside those static and dynamic cognitive structures that shape contents is thus to be avoided. In this paper we tried to sketch the general problems, foundational issues and implementational direction towards the cognitive adequacy of knowledge technologies.

Acknowledgements

The current work contributes to the Italian FIRB 2006 project "TOCAI.it" (Knowledge-oriented technologies for enterprise aggregation in Internet). Alessandro Oltramari's work was also supported by the Italian FIRB 2003 project "Knowledge management in the Extended Enterprise: new organizational models in the digital age". The final stage of this research was also sponsored by the Army Research Laboratory and was accomplished under

[19] http://www.darpa.mil/i2o/programs/me/me.asp

Cooperative Agreement Number W911NF-10-2-0061. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Finally, the author wants to thank dr. Guido Vetere (IBM Center of Advanced Studies) for the support to the implementation of TMEO in the project "Senso Comune".[20]

References

[1] Albertazzi, L.: Form Aestethics: Introduction. In: Albertazzi, L. (ed.) Shapes of Form, pp. 1–17. Kluwer, Dordrecht (1998)

[2] Albertazzi, L.: Which semantics? In: Albertazzi, L. (ed.) Meaning and cognition. A multidisciplinary approach, pp. 1–24. Benjamins Publishing Company, Amsterdam (2000)

[3] Albertazzi, L.: Deformazioni secondo regole. La Grammatica del vedere. Paradigmi 64 (2004)

[4] Borst, W.: Construction of Engineering Ontologies. Centre for Telematica and Information Technology. University of Tweente, Eschede (1997)

[5] Brentano, F.: Psychologie vom Empirischen Standpunkte. Duncker & Humblot (1874)

[6] Chen, W., Mizoguchi, R.: Communication content ontology for learner model agent in multi-agent architecture. In: AIED 1999 Workshop on Ontologies for Educational Systems (1999)

[7] Chiari AeGV, I., Oltramari: Di cosa parliamo quando parliamo fondamentale? In: Atti del Convegno della Societ'a di linguistica Italiana, Viterbo, Italy (2010)

[8] Dowty, D.R., Wall, R.E., Peters, S.: Introduction to Montague Semantics. D. Reidel, Dordrecht (1981)

[9] Finin, T., Fritzson, R., McKay, D., McIntire, R.: KQML as an Agent Communication Language. In: 3rd International Conference on Information and Knowledge Management (CIKM 1994). ACM Press, New York (1994)

[10] Genesereth, M., Ketchpel, S.: Software agents. Communications of ACM 7(37), 48–53 (1994)

[11] Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Mifflin, Boston (1979)

[12] Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M.: Ontological Engineering (with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web). Springer, London (2004)

[13] Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220 (1993)

[14] Guarino, N.: Formal ontology in information systems. In: Guarino, N. (ed.) Formal Ontology in Information Systems. Proceedings of FOIS 1998, Trento, Italy, June 6-8, pp. 3–15. IOS Press, Amsterdam (1998)

[15] Guarino, N., Giaretta, P.: Ontologies and knowledge bases: towards a terminological clarification. In: Mars, N. (ed.) Towards very large knowledge bases: knowledge building and knowledge sharing. Proceedings of KBKS 1995, Enschede, pp. 25–32. IOS Press, Amsterdam (1995)

[20] http://sensocomune.it/

[16] Kant, I.: Critica della Capacita' di Giudizio. Italian edn., Rizzoli (1995)

[17] Kripke, S.: Naming and Necessity. Basil Blackwell, Oxford (1980)

[18] Langacker, R.: Why a mind is necessary? conceptualizion, grammar and linguistic semantics. In: Albertazzi, L. (ed.) Meaning and cognition. A multidiscipliary approach, pp. 25–38. Benjamins Publishing Company, Amsterdam (2000)

[19] Langacker, R.: Concept, Image, and Symbol. The Cognitive Basis of Grammar, 2nd edn. Mouton de Gruyter, Berlin-New York (2002)

[20] Lenci, A., Calzolari, N., Zampolli, A.: From text to content: Computational lexicons and the semantic web. In: Eighteenth National Conference on Artificial Intelligence; AAAI Workshop, "Semantic Web Meets Language Resources", Edmonton, Alberta, Canada (2002)

[21] Marconi, D.: Filosofia e scienza cognitiva. Laterza (2001)

[22] Negroponte, N.: Being Digital. Alfred A. Knopf Inc. edn., New York (1995)

[23] Neisser, U.: From direct perception to conceptual structure. In: Concepts and Conceptual development, pp. 11–24. Cambridge University Press, Cambridge (1987)

[24] Prevot, L., Borgo, S., Oltramari, A.: Interfacing ontologies and lexical resources. In: OntoLex (Ontologies and Lexical Resources), Jeju Island, Soth Korea (2005)

[25] Shvaiko, P., Oltramari, A., Cuel, R., Pozza, D., Angelini, G.: Generating innovation with semantically enabled tasLab portal. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 348–363. Springer, Heidelberg (2010)

[26] Sycara, K., Paolucci, M.: Handbook on ontologies in information systems. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 343–364. Springer, Heidelberg (2004)

Chapter 7
Modal Logic Foundations of Markup Structures in Annotation Systems

Marcus Kracht

Abstract. In this paper I explain how modal logic is used to talk about structured documents and how this relates to markup languages, in particular XML. It will be seen that there is a tight connection between XPath and dynamic logic over ordered trees. This connection allows to get a good insight into the semantics and complexity of XPath.

7.1 Introduction

Markup structures have established themselves as a quasi universal tool for storing and sharing data. Deriving ultimately from attribute value systems, they have become very powerful through the use of recursive embedding. The most known format is perhaps XML, but similar formats have been used before that. What is more, typed feature structures—known mainly from computational linguistics—are quite similar (see [4] and [6], for an introduction to XML see [18]).

Formal work on markup languages has focused on the computational behaviour as well as the expressive power. On the one hand, one wants to allow for a rich query language, on the other one would like to guarantee reasonably fast algorithms considering the magnitude of the data that is being searched. The relationship between expressiveness of a language and its computational complexity is precisely the field of finite model theory [8]. Consequently, the quickest way to establish results is to measure the strength of queries into XML documents against languages studied in finite model theory. The queries are formulated in a special language of the XML family called XPath. XPath is a language that allows to define and use relations in a document. XPath is most directly connected with modal logic (see [2]). Research has therefore proceeded by comparing XPath with various variants of modal logic.

Markus Kracht
Bielefeld University, Faculty of Linguistics and Literary Studies,
Universitätsstraße 25, D-33615 Bielefeld, Germany
e-mail: marcus.kracht@uni-bielefeld.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 111–127.
springerlink.com

There is a very close connection between markup and linguistic structures. For the idea of using modal logic for analysing semistructured data derives from the research on the model theory of syntactic and phonological structures (see [12]). This was part of a research agenda now known as model theoretic syntax (MTS). Similar to finite model theory, MTS uses logical languages to describe the model structures of linguistic theories. There are two plurals here: "logical languages" means that there are alternatives. Indeed, [20] has used (weak) monadic second order logic to do this. [17] has used predicate logic with an added transitive closure operator. Similarly, "linguistic theories" means that there are several theories, not just one. Indeed, not only is there a plethora of linguistic theories, it should also be said that there are numerous other languages, each defining a different set of constraints on structures.

The research into model theoretic syntax was mainly an exercise in formalising linguistic theories. It has emerged, though, that there is a mutual benefit for markup languages. There is now quite an active research area of query languages, where techniques of finite model theory are being used to determine the strength and tractability of query languages. The results can be applied almost directly also to implementations of MTS.

The present paper serves as an introduction into the particular perspective of modal logic on markup structures, or semistructured data. Its main purpose is to explain the background and not to give a comprehensive overview over the literature and the results of the research.

Many thanks to Richard Schröder for helping me piecing the files together after some data crash and reading earlier drafts. I also thank two anonymous referees for carefully reading this paper and making many helpful suggestions. Remaining errors and omissions are solely my own responsibility.

7.2 Some Elements of Modal Logic

Propositional modal logic is defined as follows. The set of symbols of the language consists of

1. a set Var of propositional variables (typically $\mathrm{Var} = \{p_i : i \in \mathbb{N}\}$);
2. a set Con of propositional constants;
3. a set MOp of modalities;
4. and the boolean connectives $\top$, $\bot$, $\neg$, $\wedge$, $\vee$, $\rightarrow$.

So, we basically distinguish modalities from modal operators. In standard terminology, when μ is a modality, the construct $[\mu]$ is a *modal operator*. The terminology here makes the notation more user friendly and aligns modal logic with dynamic logic (see the end of this section). Propositions are formed as follows.

1. Variables and constants are propositions.
2. $\top$ and $\bot$ are propositions.
3. If φ is a proposition and μ a modality then $([\mu]\varphi)$ is a proposition as well.
4. If φ and χ are propositions, so are $(\neg\varphi)$, $(\varphi \wedge \chi)$, $(\varphi \vee \chi)$ and $(\varphi \rightarrow \chi)$.

There is another modal operator associated with the modality μ, namely $\langle\mu\rangle$. It can be defined as follows.

$$\langle\mu\rangle\varphi := (\neg([\mu](\neg\varphi))) \tag{7.1}$$

Brackets are dropped when no confusion arises. Conjunction binds stronger than disjunction and implication. Sequences of unary operators need not be disrupted by brackets. There is another notational tool that I shall borrow from dynamic logic, namely sequencing.

$$\begin{aligned} [\mu_1;\mu_2;\cdots;\mu_n]\varphi &:= [\mu_1][\mu_2]\cdots[\mu_n]\varphi \\ \langle\mu_1;\mu_2;\cdots;\mu_n\rangle\varphi &:= \langle\mu_1\rangle\langle\mu_2\rangle\cdots\langle\mu_n\rangle\varphi \end{aligned} \tag{7.2}$$

Propositions are evaluated in so-called *pointed frames*. A *frame* is a triple $\langle W,R,U\rangle$, where W is a set (members of which are called *worlds*), R is a function assigning each modality a binary relation on W, and U is a function assigning to every constant a subset of W. A *pointed frame* is a quadruple $\langle W,R,U,w\rangle$ where $\langle W,R,U\rangle$ is a frame and $w \in W$. A *valuation* is a function $\beta : \mathrm{Var} \to \wp(W)$. A *model* is a quintuple $\langle W,R,U,\beta,w\rangle$ such that $\langle W,R,U,w\rangle$ is a pointed frame and β a valuation into it.

In the following definition p ranges over variables, c over constants and μ over modalities.

$$\begin{array}{lll} \langle W,R,U,\beta,w\rangle \vDash p & :\Leftrightarrow & w \in \beta(p) \\ \langle W,R,U,\beta,w\rangle \vDash c & :\Leftrightarrow & w \in U(c) \\ \langle W,R,U,\beta,w\rangle \vDash \top & :\Leftrightarrow & \text{true} \\ \langle W,R,U,\beta,w\rangle \vDash \bot & :\Leftrightarrow & \text{false} \\ \langle W,R,U,\beta,w\rangle \vDash (\neg\varphi) & :\Leftrightarrow & \langle W,R,U,\beta,w\rangle \nvDash \varphi \\ \langle W,R,U,\beta,w\rangle \vDash (\varphi\wedge\chi) & :\Leftrightarrow & \langle W,R,U,\beta,w\rangle \vDash \varphi \\ & & \quad \text{and } \langle W,R,U,\beta,w\rangle \vDash \chi \\ \langle W,R,U,\beta,w\rangle \vDash (\varphi\vee\chi) & :\Leftrightarrow & \langle W,R,U,\beta,w\rangle \vDash \varphi \\ & & \quad \text{or } \langle W,R,U,\beta,w\rangle \vDash \chi \\ \langle W,R,U,\beta,w\rangle \vDash (\varphi\to\chi) & :\Leftrightarrow & \langle W,R,U,\beta,w\rangle \nvDash \varphi \\ & & \quad \text{or } \langle W,R,U,\beta,w\rangle \vDash \chi \\ \langle W,R,U,\beta,w\rangle \vDash ([\mu]\varphi) & :\Leftrightarrow & \text{for all } w'\text{: if } w\,R(\mu)\,w' \text{ then} \\ & & \langle W,R,U,\beta,w'\rangle \vDash \varphi \end{array} \tag{7.3}$$

It is an easy exercise left to the reader to check that

$$\langle W,R,U,\beta,w\rangle \vDash (\langle\mu\rangle\varphi) \Leftrightarrow \text{there is } w'\text{: } w\,R(\mu)\,w' \text{ and } \langle W,R,U,\beta,w'\rangle \vDash \varphi \tag{7.4}$$

(Since $\langle\mu\rangle\varphi$ is an abbreviation, this is not a definition. Rather, it now follows from the above definitions.) Note that only in the case of modalities does the world at which we evaluate get changed. In XPath terminology, we change the *focus* (see [10]).

Example 1. Linear Structures.

Let the language L_ℓ be defined by $\mathrm{Var} := \{p_i : i \in \mathbb{N}\}$, $\mathrm{Con} := \varnothing$, and $\mathrm{MOp} := \{\rightarrow, \leftarrow, \rightarrow^*, \leftarrow^*\}$. Now let $\langle W, R, U\rangle$ be such that

1. W is finite.
2. $R(\rightarrow^*)$ is the reflexive and transitive closure of $R(\rightarrow)$.
3. $R(\leftarrow^*)$ is the reflexive and transitive closure of $R(\leftarrow)$.
4. $R(\leftarrow)$ is the converse of $R(\rightarrow)$.
5. For all $v, w \in W$: either $w\, R(\rightarrow^*)\, v$ or $v\, R(\rightarrow^*)\, w$.
6. If $w\, R(\rightarrow^*)\, v$ and $v\, R(\rightarrow^*)\, w$ then $w = v$.

Recall that H is the transitive closure of $K \subseteq W \times W$ if and only if it is the smallest set that is transitive and contains K. The reflexive and transitive closure of K, denoted by K^*, is the transitive closure of $K \cup \{\langle w, w\rangle : w \in W\}$. Also, $K^{\smile} := \{\langle y, x\rangle : \langle x, y\rangle \in K\}$ is called the *converse* of K. It is easy to see that $(K^{\smile})^* = (K^*)^{\smile}$.

The above conditions on the frames say that $\langle W, R(\rightarrow^*)\rangle$ is a finite linear order. We may think, for example, of a file as an instance of such a structure. (The frame here supplies only the positions; I shall discuss below where the actual symbols of the file come in.)

Example 2. Ordered Trees.

Let the language L_t be defined by $\mathrm{Var} := \{p_i : i \in \mathbb{N}\}$, $\mathrm{Con} := \varnothing$, and $\mathrm{MOp} := \{\uparrow, \downarrow, \rightarrow, \leftarrow, \uparrow^*, \downarrow^*, \rightarrow^*, \leftarrow^*\}$. Now let $\langle W, R, U\rangle$ be such that

1. W is finite.
2. $R(\uparrow^*)$ $(R(\downarrow^*), R(\rightarrow^*), R(\leftarrow^*))$ is the reflexive and transitive closure of $R(\uparrow)$ $(R(\downarrow), R(\rightarrow), R(\leftarrow))$.
3. $R(\uparrow)$ is the converse of $R(\downarrow)$; $R(\leftarrow)$ is the converse of $R(\rightarrow)$.
4. There is exactly one w such that for all v: $w\, R(\downarrow^*)\, v$. (This is called the *root*.)
5. For all v, the set of w such that $w\, R(\downarrow^*)\, v$ is linearly ordered by $R(\downarrow^*)$.
6. For all v, the set of w such that $w\, R(\rightarrow^*)\, v$ or $w\, R(\leftarrow^*)\, v$ is linearly ordered by $R(\rightarrow^*)$.

This structure is therefore defined by only two of the eight relations, namely the 'vertical' $R(\downarrow)$ and the 'horizontal' $R(\rightarrow)$. For $R(\uparrow)$ is the converse of $R(\downarrow)$, and $R(\leftarrow)$ the converse of $R(\rightarrow)$; and the other relations are reflexive and transitive closures of these four.

Such a structure is easily recognised as an *ordered tree*. It will play a fundamental role in what is to follow. It is a tree because of the vertical relation $R(\downarrow)$ satisfies the typical properties of immediate dominance; it is ordered since the daughters of each node (in the tree sense) are linearly ordered with respect to each other.

We conclude this section with some remarks on PDL. Propositional Dynamic Logic or PDL, presents a substantial strengthening of modal logic. In PDL, modalities are called *programs*. As before there is a fixed set MOp of programs; however, programs can now be combined.

1. If α and β are programs, so are α^*, $\alpha;\beta$ and $\alpha \cup \beta$.
2. If φ is a proposition, $\varphi?$ is a program.

We extend the function R as follows.

$$\begin{array}{ll} R(\alpha;\beta) & := R(\alpha) \circ R(\beta) \\ R(\alpha \cup \beta) & := R(\alpha) \cup R(\beta) \\ R(\alpha^*) & := R(\alpha)^* \\ R(\varphi?) & := \{\langle x,x\rangle : x \vDash \varphi\} \end{array} \tag{7.5}$$

In PDL with converse we also have an operator $\smallsmile$ on programs. Furthermore, $R(\alpha^{\smallsmile}) := R(\alpha)^{\smallsmile}$. I note here the following properties of the converse, which I state as identities between programs.

$$\begin{array}{ll} (\alpha \cup \beta)^{\smallsmile} & = \alpha^{\smallsmile} \cup \beta^{\smallsmile} \\ (\alpha;\beta)^{\smallsmile} & = \beta^{\smallsmile};\alpha^{\smallsmile} \\ (\alpha^*)^{\smallsmile} & = (\alpha^{\smallsmile})^* \\ (\varphi?)^{\smallsmile} & = \varphi? \end{array} \tag{7.6}$$

Using PDL we can reduce the number of basic modalities as follows. $\uparrow^*$ can be defined; in fact, the notation has now become transparent in the right way. Now we have only four basic programs, $\uparrow$, $\downarrow$, $\rightarrow$, and $\leftarrow$. With converse, just two of them suffice.

7.3 Classes of Models

Given a proposition and a model we can evaluate the proposition and see whether it is true; this is known as *model checking*, since we are checking the proposition in the model. Given a proposition, we can also ask which frames allow no countermodel for it. This is used in axiomatising classes of structures. First a few definitions. We write $\langle W,R,U,w\rangle \vDash \varphi$ if for all valuations β: $\langle W,R,U,\beta,w\rangle \vDash \varphi$. In this way, a pointed frame satisfies a formula if it satisfies the formula for all valuations; with propositional quantifiers (which we do not use) we may write this as $\langle W,R,U,w\rangle \vDash (\forall \overline{p})\varphi$, where $\overline{p}$ collects all variables occurring in φ. We write $\langle W,R,U\rangle \vDash \varphi$ if for all $w \in W$: $\langle W,R,U,w\rangle \vDash \varphi$.

Definition 1 (Axiomatisable Classes). *Let $\mathcal{K}$ be a class of pointed frames. $\mathcal{K}$ is called **axiomatisable** if there is a set Δ of formulae such that $\langle W,R,U,w\rangle \in \mathcal{K}$ iff $\langle W,R,U,w\rangle \vDash \delta$ for all $\delta \in \Delta$.*

Similarly for classes of frames. I shall present here a few positive and negative results. First, let us note the following. Given w, let $\mathrm{Tr}(w)$ denote the set of worlds that can be reached from w using any of the relations. Formally, let S be the reflexive and transitive closure of the union of all the $R(\mu)$. Then

$$\mathrm{Tr}(w) := \{v : w\,S\,v\} \tag{7.7}$$

Furthermore, let $R'(\mu) := R(\mu) \cap S$ and $U'(c) := U(c) \cap \mathrm{Tr}(w)$. Then $\langle \mathrm{Tr}(w), R', U', w\rangle$ is called the *subframe generated by w*. Given a valuation β into W, we put $\beta'(p) := \beta(p) \cap \mathrm{Tr}(w)$. Then

$$\langle W, R, U, \beta, w\rangle \vDash \varphi \text{ iff } \langle \mathrm{Tr}(w), R', U', \beta', w\rangle \vDash \varphi \tag{7.8}$$

If β' is a valuation on the generated subframe then we may put $\beta(p) := \beta'(p)$. In that case, (7.8) also holds.

Definition 2. *$\langle W, R, U, w\rangle$ is said to be* ***generated*** *if $W = \mathrm{Tr}(w)$. $\langle W, R, U\rangle$ is said to be* ***1-generated*** *if there is a w such that $W = \mathrm{Tr}(w)$.*

It then follows that if $\mathscr{K}$ and $\mathscr{L}$ are axiomatisable classes of frames whose 1-generated members are identical then the two classes are identical. Similarly, two axiomatisable classes of pointed frames are identical if only their generated members are the same. In view of this, it is perhaps better to axiomatise just classes of generated pointed frames (1-generated frames).

It turns out that the classes of finite linear frames is axiomatisable (as a class of 1-generated frames).

Theorem 3. *The following holds.*

1. $\langle W, R, U\rangle \vDash \langle\mu\rangle p_0 \to \langle\nu\rangle p_0$ *iff* $R(\mu) \subseteq R(\nu)$.
2. $\langle W, R, U\rangle \vDash \langle\mu;\mu\rangle p_0 \to \langle\mu\rangle p_0$ *iff* $R(\mu)$ *is transitive.*
3. $\langle W, R, U\rangle \vDash p_0 \to [\mu]\langle\nu\rangle p_0$ *iff* $R(\mu) \subseteq R(\nu)^{\smile}$.
4. $\langle W, R, U\rangle \vDash [\nu](p_0 \to [\mu]p_0) \wedge p_0 \to [\nu]p_0$ *iff* $R(\mu)^* \supseteq R(\nu)$.
5. $\langle W, R, U\rangle \vDash \langle\mu\rangle p_0 \to [\mu]p_0$ *iff* $R(\mu)$ *is a partial function.*
6. $\langle W, R, U\rangle \vDash [\mu]([\mu]p_0 \to p_0) \to [\mu]p_0$ *iff* $R(\mu)$ *is transitive and conversely well-founded.*

I show the second claim. Suppose that $\langle W, R, U\rangle \vDash \langle\mu;\mu\rangle p_0 \to \langle\mu\rangle p_0$. Now let $x\, R(\mu)\, y\, R(\mu)\, z$. Pick β such that $\beta(p_0) := \{z\}$. Then $\langle W, R, U, \beta, x\rangle \vDash \langle\mu;\mu\rangle p_0$. Hence, by assumption, $\langle W, R, U, \beta, x\rangle \vDash \langle\mu\rangle p_0$. So there is a u such that $x\, R(\mu)\, u$ and $\langle W, R, U, \beta, u\rangle \vDash p_0$. By choice of β this means $u = z$, and so $x\, R(\mu)\, z$. Conversely, suppose that $R(\mu)$ is transitive. Pick x and β such that $\langle W, R, U, \beta, x\rangle \vDash \langle\mu;\mu\rangle p_0$. Then there are y and z such that $x\, R(\mu)\, y\, R(\mu)\, z$ and $\langle W, R, U, \beta, z\rangle \vDash p_0$. $R(\mu)$ is transitive, and therefore $x\, R(\mu)\, z$, which means that $\langle W, R, U, \beta, x\rangle \vDash \langle\mu\rangle p_0$.

For a somewhat more difficult case, I turn to (6). Rather than proving the entire claim, let me show that the formula is valid on a transitive, conversely well-founded frame $\langle W, R, U\rangle$. To that end, let P_0 be all the points that have no R-successor. Inductively, define P_α to be the set of all points w such that all successors are in P_β for some $\beta < \alpha$ and for every $\beta < \alpha$ there is some $u \in P_\beta$ which is a successor of w. (This definition is over all ordinals, it does not require the frame to be finite.) By ordinal induction it is shown that $\langle W, R, U, \beta, w\rangle \vDash [\mu]([\mu]p_0 \to p_0) \to [\mu]p_0$ for every $w \in P_\alpha$. To that end, assume that the claim has been shown for all $\beta < \alpha$. Pick $w \in P_\alpha$. Assume that $\langle W, R, U, \beta, w\rangle \vDash [\mu]([\mu]p_0 \to p_0)$. We need to show that $\langle W, R, U, \beta, w\rangle \vDash [\mu]p_0$. To that end we pick a successor u. It is in P_β for some $\beta < \alpha$. By assumption on w, we have $\langle W, R, U, \beta, u\rangle \vDash [\mu]p_0 \to p_0$. By transitivity, we also have $\langle W,R,U,\beta,u\rangle \vDash [\mu]([\mu]p_0 \to p_0)$. Finally, by inductive hypothesis we have $\langle W,R,U,\beta,u\rangle \vDash [\mu]([\mu]p_0 \to p_0) \to [\mu]p_0$. This gives $\langle W,R,U,\beta,u\rangle \vDash [\mu]p_0$, and finally $\langle W,R,U,\beta,u\rangle \vDash p_0$. u has been arbitrary. This shows the claim. (Notice that we do not need to prove the case $\alpha = 0$ separately. It is however easy to see directly that the claim holds in that case.)

Corollary 4. *The following holds.*

① *The class of linear orders is axiomatisable in L_ℓ (as a class of 1-generated frames).*

② *The class of (ordered) trees is axiomatisable in L_t (as a class of 1-generated frames).*

Let me also say a few words about the relationship between frames and pointed frames. We say that ξ is a *master modality* in $\langle W,R,U\rangle$ if $R(\xi)$ is reflexive, transitive, and for all $\mu \in \mathrm{MOp}$, $R(\mu) \subseteq R(\xi)$.

Proposition 5. *Let $\langle W,R,U,w\rangle$ be generated and assume that ξ is a master modality. Then $\langle W,R,U,\beta,w\rangle \vDash [\xi]\varphi$ iff $\langle W,R,U,\beta\rangle \vDash \varphi$. Also, $\langle W,R,U,w\rangle \vDash [\xi]\varphi$ iff $\langle W,R,U\rangle \vDash \varphi$.*

It is possible to axiomatise the class of frames where a given modality is guaranteed to be a master modality. In linear frames, there is no such master modality. However, it turns out that we have something that is effectively the same. Namely, if both $R(\to^*)$ and $R(\leftarrow^*)$ are linear and each others converse, then for every x,y there is a z such that: $x\,R(\to^*)\,z\,R(\leftarrow^*)\,y$. Thus, in place of the above we have

$$\langle W,R,U,\beta,w\rangle \vDash [\to^*;\leftarrow^*]\varphi \;\Leftrightarrow\; \langle W,R,U,\beta\rangle \vDash \varphi \tag{7.9}$$

7.4 Modal Logic and DOMs

Now we turn to more realistic models. First of all, we like to define a realistic theory of a file. To this end, all we need to do is to take the language of our first example and supplement it with constants. There are many ways to go. Given our alphabet A of letters, we can define a constant $\underline{a}$ for each letter of A. (The alphabet is usually assumed to be finite.) Then we add the axioms $\underline{a} \to \neg\underline{b}$ for all $a,b \in A$ such that $a \neq b$; moreover we shall add $\bigvee\langle \underline{a} : a \in A\rangle$. In raw (that is, binary) format, we can take A to be just $\{0,1\}$. The worlds are then the positions of the individual bits. But different structures can be used (say, with a constant for each alphabetical symbol of Unicode).

A given file is therefore a 1-generated frame for this logic. It has the form $\langle W,R,U\rangle$, where W is a finite set, and $R(\to)$ is a partial function with inverse $R(\leftarrow)$. Since the frame is 1-generated, every point is connected with every other, which means that we have a finite linear order in the standard sense. The worlds can thus be identified with an initial segment of the natural numbers. The function U assigns to each constant $\underline{a}$, with $a \in A$, a set $U(\underline{a})$. If $i \in U(\underline{a})$ the file is said to carry the letter a at position i. The axiom $\underline{a} \to \neg\underline{b}$ guarantees that every position carries exactly one letter.

Consider a generated pointed frame $\langle W, R, U, i\rangle$. Here, the members of W are numbers, and i is a particular number. This is equivalent to a file plus cursor position. The commands that move the cursor to the right and left can be interpreted as changing the world i to $i+1$ and $i-1$, respectively. More complex motions of the cursor can be defined, though the present language may be too limited for that. Without the modalities, we can only determine what symbol is present at the current position; the modalities present something of a lookahead. For example, $\langle\rightarrow\rangle\underline{a}$ is true at i if $i+1$ carries the letter a, equivalently, if $\underline{a}$ is true at $i+1$.

Let me now turn to XML. For readers unfamiliar with this format, I advise to get hold of a book on XML, for example [18]. An XML-document is a file, that is, in first instance a string. The data in the file combines both the primary data itself and the metadata in the form of annotation tags. The tags may carry any additional information about this data. But first and foremost they serve to structure it. They turn it into a *tree*.

```
<library>
   <book>
      <author>Hugo</author>
      <title>Les Misérables</title>
   </book>
   <book>
      <author>Flaubert</author>
      <title>Madame Bovary</title>
   </book>
</library>
```

(7.10)

For each opening tag, say `<author>` there must be a corresponding closing tag, `</author>`. In principle, tags can be inserted anywhere as long as they are properly nested. However, from a theoretical point of view it is best to require that text can only be inserted between deepest embedded tags. (This is the default in XSL Schema, by the way.) A slight modification of the structure is enough to get this form. Say we have the following line.

```
<p>This was an <i>inspiring</i> discussion.</p>
```

(7.11)

This line is then transformed as follows, where `<text>` is a tag reserved for text input.

```
<p><text>This was an</text>
   <i><text>inspiring</text></i>
   <text>discussion.</text></p>
```

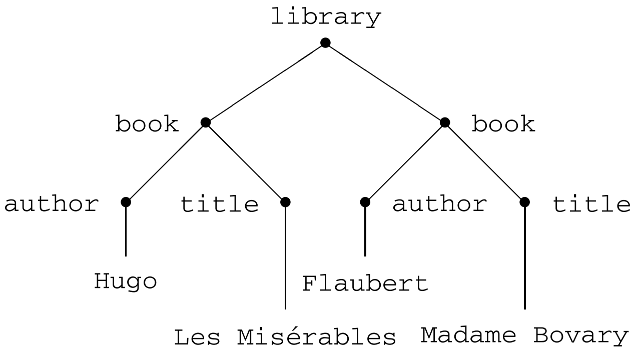
(7.12)

This is the form we shall assume here. In XML talk, we disallow the mixed type.

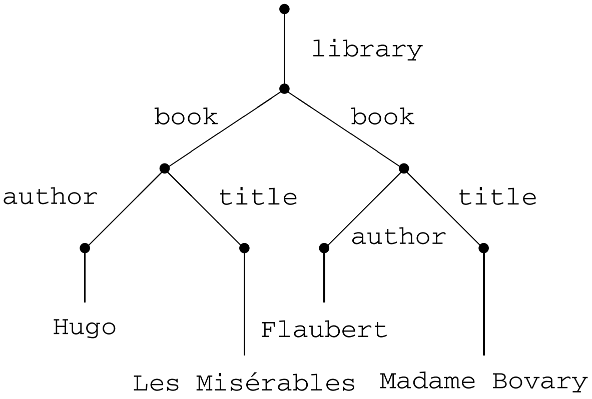
The linear structure (7.10) is converted into a tree structure in the following way. A tag <τ> together with the next corresponding closing tag </τ> define a constituent of type τ. In this way, a linguistic representation of the above structure might look like this:

$$\begin{array}{l}[_{\texttt{library}}[_{\texttt{book}}[_{\texttt{author}}\texttt{Hugo}][_{\texttt{title}}\texttt{Les Misérables}]] \\ [_{\texttt{book}}[_{\texttt{author}}\texttt{Flaubert}][_{\texttt{title}}\texttt{Madame Bovary}]]]\end{array} \tag{7.13}$$

However, this is not entirely adequate. For disregarding order the structure is a node labelled directed graph. These are triples $\langle N, E, \ell \rangle$, where $E \subseteq N \times N$ and $\ell : N \to L$, where L is a labelling domain.



However, in semistructured data we like to think of these structures as edge labelled graphs [1]. These are triples $\langle N, E, \ell \rangle$, where $E \subseteq N \times N$ and $\ell : E \to L$. These graphs are called the **DOM**s (document object models).



This necessitates the introduction of a root node since we need an edge with label `library`. Now, in this structure the tags correspond to edge labels. This does not apply to the attributes, though. Furthermore, types and content are properties of the nodes, not the edges. Thus the following tag represents a mixture of edge and node labels:

$$< \underbrace{\texttt{author}}_{\substack{\uparrow\\ \text{edge}\\ \text{label}}}\ \underbrace{\texttt{ID="VH07"}}_{\substack{\uparrow\\ \text{node}\\ \text{label}}}\ \underbrace{\texttt{nationality="French"}}_{\substack{\uparrow\\ \text{node}\\ \text{label}}} > \tag{7.14}$$

In fact, as noted in [1], the notation is not fully transparent with respect to the structure that it represents. First, ID and IDREF should be treated separately; they deal with so-called oids (object identifiers). Attributes on the other hand should also be seen as edge labels, so that they are basically equivalent to tags. However, one difference remains: attributes are not recursive; and they are not ordered. I remark briefly that in a tree there is a simple correspondence between edge labels and labels on nodes other than the root. Observe that every node has a unique parent, so if we make the edge label a label of the endpoint of the edge, the edge labelling can be recovered except of the root node. The document root however is added on top of the root node of the tree; and this makes the correspondence exact. Also note that in practice, XPath seems to treat DOMs rather as node labeled trees.

I shall first present a formalism for fixed tag sets (for example, HTML). In the next section I shall return to XML. The tags are at present undecorated (no attributes, no identifiers). Hence, for each tag τ we take a separate modal operator with the same name. So, we have, in the case of HTML, modal operators h1, p, and so on. The following is assumed: the union over all $R(\tau)$, where τ is a tag, corresponds to the relation $R(\downarrow)$. Second, for different tags τ and τ', $R(\tau) \cap R(\tau') = \varnothing$ (though, surprisingly, this is not modally axiomatisable unless all daughters of a node are linearly ordered with respect to each other). Third, $x\, R(\rightarrow)\, y$ only if there is z and a tag τ such that $z\, R(\tau)\, x, y$. This is a known situation: only the siblings are ordered with respect to each other. (Thus $R(\rightarrow)$ is the sibling ordering, not the linear precedence in the file defining the model.) A different proposal is to assume that $x\, R(\rightarrow)\, y$ if x and y have ancestors that are neighbouring siblings. (So, using the previous definition, we take the ordering to be $R(\uparrow^*) \circ R(\rightarrow) \circ R(\downarrow^*)$.) The sibling order is derived from the linear precedence in the XML-document (the file), which specifies a linear order on the entire set of nodes.

7.5 XPath

In XML the tagset is not fixed. Yet we still can encode XML structures with a finite set of modalities. The trick is as follows. We return to the language of ordered trees. The edge label no longer defines a modality in its own right. Rather, we make the edge label a property of the node to which the edge points. This is exactly how it is done in XML. (And it is the reason why in the DOM we need an extra root node.) We introduce a new modality, τ, which relates a node with its tag. XPath has a function called name() to return the name of a node. However, notice that there is no limit on the number of tags, so we need to convert tags into structures as well. This we can do by representing the tag in the model as a string. Thus we arrive at a new sort of language and structures. They extend the ordered trees. The additional postulates are as follows. Say that z is a *tag node* if there are x and y such that $x\,R(\tau)\,y$, and $z \in \mathrm{Tr}(y)$. Then we add the condition that if z is a tag node, there is no y such that $z\,R(\tau)\,y$ or $z\,R(\uparrow)\,y$ or $z\,R(\downarrow)\,y$. This makes $\mathrm{Tr}(y)$ in effect a linear structure, for any tag node y. It is important to note that if z is not a tag node and $z\,R(\mu)\,y$ for $\mu \in \{\uparrow, \downarrow, \rightarrow, \leftarrow\}$, then y also is not a tag node. Thus, the only way to enter tag nodes is via the relation τ. Finally, the tag nodes are treated as coding a string, so the letters are basically introduced via constants. It must then be specified that the constants are false at ordinary tree nodes.

With this in mind we shall now turn to the analysis of XPath. XPath is a language for selecting nodes from a tree, based on various properties. These properties need not be local to the node (like the tag), mostly they involve ways in which the node is embedded in a structure. We can define the set of nodes that have a particular parent node, for example. We can also find nodes based on their linear position in the DOM. The expressive power of XPath is therefore quite rich; too rich to receive a comprehensive logical treatment. (In fact, [5] show that adding numeric comparisons quickly leads to undecidability.) [9] have therefore proposed to define a subset, called Core XPath, where only the relational properties are studied. It is this language that we shall look at below. XPath itself nowadays has two versions: there is XPath 1.0 and XPath 2.0. Consequently, there is Core XPath 1.0, and Core XPath 2.0. The discussion below treats Core XPath 1.0, and takes only a brief look at Core XPath 2.0.

XPath contains so-called *axes*. These are relations between nodes in a tree. Here are the main axis relations:

XPath	PDL
`parent`	$\uparrow$
`ancestor-or-self`	$\uparrow^*$
`ancestor`	$\uparrow;\uparrow^*$
`child`	$\downarrow$
`descendant-or-self`	$\downarrow^*$
`descendant`	$\downarrow^*;\downarrow$
`following-sibling`	$\rightarrow^*;\rightarrow$
`following`	$\uparrow^*;\rightarrow;\rightarrow^*;\downarrow^*$
`preceding-sibling`	$\leftarrow^*;\leftarrow$
`preceding`	$\uparrow^*;\leftarrow;\leftarrow^*;\downarrow^*$

(7.15)

There are three more: `self`, `namespace` and `attribute`. The actual surface syntax of XPath is somewhat different. There are first of all two types of path expressions: *relative* and *absolute*. We deal with the relative pattern first. The relative pattern is composed from so-called **step patterns** by means of `/` and `//`. A step pattern in turn is a sequence consisting of (a) an axis specifier, (b) a node test, and (c) a sequence of predicates. The axis specifier says whether the node test specifies the value of an attribute or the label. The predicates are properties of nodes. They are enclosed in square brackets. These properties can be even numeric, but it is customary to restrict them to what is expressible to the relational language described here. The absolute paths are obtained from the relative paths by prefixing them with `/` or `//` (as is customary in Unix). This makes `/` and `//` both unary and binary symbols. Finally, it is important to realise that path expressions can have a short form and a long form. What appears in short as `author` is in long form `self::node()/child::author`. In the long form the symbol `/` is interpreted as relational composition while in the short form it effectively takes the meaning "compose with child-of", as it does in Unix. (More on the precise syntax and the relationship to modal logic and relational algebras can be found in [5].)

One big problem area in the theory of markup language is fast algorithms for the path containment problem. This is the problem to determine, given two path descriptions p and q, whether or not in every tree all the p-paths are included in the q-paths. One of the reasons to be interested in this problem is in reformulating queries either to speed them up or to discover if they are consistent [7, 16].

Since the full language is quite difficult to tackle one is therefore interested in fragments of it. A popular fragment is one where the horizontal axes are eliminated. One can then in effect only talk about hierarchy, not about linear order. An XP-expression is an expression generated by the following grammar, taken from [16]. It is based on the short forms and does not use upward directed axes.

$$\begin{array}{rll} p := & p_1 \mid p_2 & \text{(disjunction)} \\ & \mid \mathtt{/}p & \text{(root)} \\ & \mid \mathtt{//}p & \text{(descendant)} \\ & \mid p_1 \mathtt{/} p_2 & \text{(child)} \\ & \mid p_1 \mathtt{//} p_2 & \text{(descendant)} \\ & \mid p_1 \mathtt{[} p_2 \mathtt{]} & \text{(filter)} \\ & \mid \sigma & \text{(element test)} \\ & \mid \star & \text{(wildcard)} \end{array} \tag{7.16}$$

Here, σ is a basic boolean constant. The queries are evaluated as relations in Kripke-models. Let $\mathfrak{T}$ be a tree with root r and $\langle W, R, U, \beta, w\rangle$ a Kripke-model.

$$\begin{array}{ll} [\![p_1 \mid p_2]\!]_{\mathfrak{T}} & = [\![p_1]\!]_{\mathfrak{T}} \cup [\![p_2]\!]_{\mathfrak{T}} \\ [\![\mathtt{/}p]\!]_{\mathfrak{T}} & = [\![p]\!]_{\mathfrak{T}} \cap \{r\} \times W \\ [\![\mathtt{//}p]\!]_{\mathfrak{T}} & = \{\langle r, w\rangle : \exists u : \langle u, w\rangle \in [\![p]\!]_{\mathfrak{T}}\} \\ [\![p_1 \mathtt{/} p_2]\!]_{\mathfrak{T}} & = [\![p_1]\!]_{\mathfrak{T}} \circ R(\downarrow) \circ [\![p_2]\!]_{\mathfrak{T}} \\ [\![p_1 \mathtt{//} p_2]\!]_{\mathfrak{T}} & = [\![p_1]\!]_{\mathfrak{T}} \circ R(\downarrow^+) \circ [\![p_2]\!]_{\mathfrak{T}} \\ [\![p_1 \mathtt{[} p_2 \mathtt{]}]\!]_{\mathfrak{T}} & = \{\langle v, w\rangle \in [\![p_1]\!]_{\mathfrak{T}} : \exists u : \langle w, u\rangle \in [\![p_2]\!]_{\mathfrak{T}}\} \\ [\![\sigma]\!]_{\mathfrak{T}} & = \{\langle w, w\rangle : w \in \beta(\sigma)\} \\ [\![*]\!]_{\mathfrak{T}} & = \{\langle w, w\rangle : w \in W\} \end{array} \tag{7.17}$$

Based on work by [15], [16] establish the following. The first, standard, case is when the set L of labels is infinite:

Theorem 6. *The following holds.*

1. *Containment of XP(/, //, [], $*$, $|$) is in* CONP.
2. *Containment of XP(/, $|$) is* CONP*-hard.*
3. *Containment of XP(//, $|$) is* CONP*-hard.*

A problem P is in CONP if it can be verified in nondeterministic polynomial time (= NP) whether a given structure is a counterexample to P.

Theorem 7. *Let L be finite.*

1. *Containment of XP(/, //, [], $*$, $|$) is in* PSPACE.
2. *Containment of XP(/, //, $|$) is* PSPACE*-hard.*

[16] contains many more results. Additional complexity comes from the addition of DTDs, which describe the structure of documents. The problem becomes the following: given a DTD d and two path expressions p, q say whether $[\![p]\!]_{\mathfrak{T}} \subseteq [\![q]\!]_{\mathfrak{T}}$ for all $\mathfrak{T}$ satisfying d. Effectively, DTDs are some kind of axioms. Thus adding a DTD may in fact increase the complexity of the problem.

Theorem 8. *The following holds.*

1. *Containment of XP(DTD, /, []) is in* CONP.
2. *Containment of XP(DTD, /, []) is* CONP*-hard.*
3. *Containment of XP(DTD, //, []) is* CONP*-hard.*
4. *Containment of XP(/, //, [], $*$) is* EXPTIME*-complete.*
5. *Containment of XP(/, //, $|$) is* EXPTIME*-complete.*

7.6 Paths in Dynamic Logic

Based on the results of Sections 7.2 and 7.3 we can conclude.

Theorem 9. *The class of ordered forests is axiomatisable in* PDL *over* $\uparrow$, $\downarrow$, $\rightarrow$ *and* $\leftarrow$. *The logic is denoted by* $\mathsf{PDL}_{\mathsf{t}}$.

It is possible to extend this to edge labelled forests. Just add one more operator, node, and the axioms

$$[\texttt{node};\texttt{node}]\bot, \qquad \langle\texttt{node}\rangle p \rightarrow [\texttt{node}]p \tag{7.18}$$

This makes the interpretation of node a relation R such that if $x\ R\ y$ then y has no R-successor, and if $x\ R\ z$ then $y = z$. The value at this node is any text (the edge label). If there are only finitely many tags available, then we can mimic the edge labels by a fixed set of boolean constants. Otherwise, we need to encode the strings over an alphabet.

We can offer an analysis of path expressions by translating them into PDL expressions. Recall that in XPath expressions are evaluated into node sets. The set contains the nodes satisfying the expression. Since the syntax of path expressions is somewhat different from PDL, we must first translate them. This must be done with care. For in the previous section we have just translated them as relations. Here we must reduce them to formulae. Given an expression `book/author` we must decide whether we look at the set of nodes where the path originates or whether we look at the set of nodes where the path ends. In a template, for example, we look at the nodes where the path ends. So, for each path we have two translations, $\{\cdot\}^o$ (looking at nodes where the paths originate) and $\{\cdot\}^g$ (looking at nodes where the paths end). These can be derived, though, from a direct translation $\{\cdot\}^\delta$ into relations, which runs as follows (based on the long form).

$$\begin{aligned}
&\{p_1 \,|\, p_2\}^\delta &&:= \{p_1\}^\delta \cup \{p_2\}^\delta \\
&\{p_1 / p_2\}^\delta &&:= \{p_1\}^\delta \circ \{p_2\}^\delta \\
&\{p_1\,[\,p_2\,]\,\}^\delta &&:= \{p_1\}^\delta \circ \{\langle x,x\rangle : (\exists y)(\langle x,y\rangle \in \{p_2\}^\delta)\} \\
&\{\rho :: \tau\}^\delta &&:= \rho \circ \tau?
\end{aligned} \tag{7.19}$$

Now $\{\cdot\}^o$ can be obtained, translating $\circ$ by ; and $\cup$ by $\cup$:

$$\begin{aligned}
&\{p_1 \,|\, p_2\}^o &&:= \{p_1\}^o \cup \{p_2\}^o \\
&\{p_1 / p_2\}^o &&:= \{p_1\}^o ; \{p_2\}^o \\
&\{p_1\,[\,p_2\,]\,\}^o &&:= \{p_1\}^o ; (\langle\{p_2\}^o\rangle\top)? \\
&\{\rho :: \tau\}^o &&:= \rho ; \tau?
\end{aligned} \tag{7.20}$$

To see the rationale behind this translation note that $\langle\alpha\rangle\top$ is true at a point iff there is an α-path starting at that point. Also, $\langle\alpha;\varphi?\rangle\top$ is true iff there is an α-path ending in a node satisfying φ. With the converse operator we can now write as follows. If $x \vDash \langle\alpha\rangle\top$ selects the nodes at which α can successfully start, $\{x : x \vDash \langle\alpha^{\smile}\rangle\top\}$ selects the nodes at which α ends. Thus we have

$$\{\pi\}^g = (\{\pi\}^o)^{\smile} \tag{7.21}$$

Applying this to path expressions gives the following.

$$\begin{aligned}
&\{p_1 \,|\, p_2\}^g &&:= \{p_1\}^g \cup \{p_2\}^g \\
&\{p_1 / p_2\}^g &&:= \{p_2\}^g ; \{p_1\}^g \\
&\{p_1\,[\,p_2\,]\,\}^g &&:= \langle\{p_2\}^o\rangle\top? ; \{p_1\}^g \\
&\{\rho :: \tau\}^g &&:= \tau? ; \rho^{\smile}
\end{aligned} \tag{7.22}$$

The third line is noteworthy (the derivation makes use of the equations (7.6)).

$$\begin{aligned}
\{p_1\,[\,p_2\,]\,\}^g &= (\{p_1\,[\,p_2\,]\,\}^o)^{\smile} \\
&= (\{p_1\}^o ; \langle\{p_2\}^o\rangle\top?)^{\smile} \\
&= (\langle\{p_2\}^o\rangle\top?)^{\smile} ; (\{p_1\}^o)^{\smile} \\
&= \langle\{p_2\}^o\rangle\top? ; \{p_1\}^g
\end{aligned} \tag{7.23}$$

Absolute paths can be defined as follows.

$$\{/p\}^o := \neg\langle\uparrow\rangle\top? ; \{p\}^o \tag{7.24}$$

The program $\varphi?;\alpha$ effectively restricts the set of nodes to those where φ is true.

Here is a classical result.

Theorem 10 ([22]). *The satisfiability problem for* PDL *is in* EXPTIME.

By reducing it to the original result, [2] show that the same complexity holds for the logic of trees.

Theorem 11 ([2]). *The satisfiability problem of* $\mathsf{PDL_t}$ *is in* EXPTIME.

Path inclusion can be reduced to the negation of a satisfiability problem. Namely, $\pi \nsubseteq \rho$ in *some* structure iff $[\{\pi\}^o]p \wedge \langle\{\rho\}^o\rangle\neg p$ is satisfiable in $\mathsf{PDL_t}$. Formulated differently, $\pi \subseteq \rho$ in *every* structure iff $[\{\rho\}^o]p \to [\{\pi\}^o]p \in \mathsf{PDL_t}$. According to Theorem 11 the problem is decidable in EXPTIME. Notice that the actual queries that can be issued in XPath are a proper subset of the queries that are definable in $\mathsf{PDL_t}$ [13].

An interesting property of $\mathsf{PDL_t}$ is that we can define nominals. Nominals are special kinds of propositional variables that may be instantiated to a single point [3]. That is, if i is a nominal and $\langle W, R, U, \beta, w\rangle$ a model then $\beta(i) = \{v\}$ for some $v \in W$. Given a formula $\varphi(i)$ which contains such a nominal, we can replace the nominal by a standard variable p as follows. We put

$$\neq := \downarrow^+ \cup (\uparrow^*; (\rightarrow^+ \cup \leftarrow^+); \downarrow^*) \tag{7.25}$$

It is not hard to see that $R(\neq) = \{\langle v, w\rangle : v \neq w\}$. Then suppose that the following is true at w:

$$\langle\uparrow^*; \downarrow^*\rangle(p \wedge [\neq]\neg p) \wedge \varphi(p) \tag{7.26}$$

Then at some point v, $v \vDash p \wedge [\neq]\neg p$, which is to say that $\beta(p) = \{v\}$, as required. Since an added nominal can be recoded at constant expense, the complexity does not rise in $\mathsf{PDL_t}$ if we add nominals.

Formally, the argument runs as follows. Let $\mathsf{NPDL_t}$ be the extension of $\mathsf{PDL_t}$ by nominals. Then one can show that for every formula φ there is a formula $\varphi^\sharp$ such that $\varphi \in \mathsf{NPDL_t}$ iff $\varphi^\sharp \in \mathsf{PDL_t}$. The complexity of $\mathsf{NPDL_t}$ can be bounded from that of $\mathsf{PDL_t}$ using the properties of the map $\varphi \mapsto \varphi^\sharp$. Since in this case $|\varphi^\sharp| \leq c|\varphi|$ for some constant c, if $\mathsf{PDL_t}$ is decidable in $f(x)$ time, $\mathsf{NPDL_t}$ is decidable in $f(cx)$ time. The same argument can be used to show that the logic with an added constant axiom ξ has the same complexity. Here we take $\varphi \to \varphi \wedge [\uparrow^*; \downarrow^*]\xi$. The function is $f(d+x)$, where d is a constant (1 plus the length of $[\uparrow^*; \downarrow^*]\xi$. Likewise, making a program deterministic or adding the converse typically have no effect on the complexity (see [11]). Let us close this section with a quick look at Core XPath 2.0. This language extends Core XPath 1.0 by operators on paths. There are new constructs `union`, `intersect`, and `except` to form the union, intersection and difference of relations or path sets. Paths can be combined using these expressions. This exceeds the syntactic means of PDL, so [5] turn to relation algebras instead. This language is decidable, and expressively complete for first-order logic. This means that if a property of nodes is first-order definable it is also definable in Core XPath 2.0 [14]. Since the paths definable in Core XPath 1.0 are not closed under negation [19], this language is therefore stronger. [5] give a complete axiomatisation of path equivalence.

7.7 Conclusion

A logical analysis of computer languages is important in many respects: it gives us a clear idea of what is expressible and what is not; it also gives us a clear notion of the sort of structures we are using; and third, it allows to prove precise results about the complexity of algorithms. The analysis of XML, in particular Core XPath, in terms of modal logic is a good example of this. The relational character of the models for modal logic make it very useful in studying DOMs from an abstract point of view. Also, expressive and computational properties of Core XPath can be addressed succinctly. Although quite different in detail, the two languages share a large enough core to allow for useful results. Since the model theory of PDL is well understood, results can be transferred almost immediately.

References

[1] Abiteboul, S., Bunemann, P., Suciu, D.: Data on the Web. In: From Relations to Semistructured Data and XML. Morgan Kaufmann, San Francisco (2000)

[2] Afanasiev, L., Blackburn, P., Dimitriou, I., Gaiffe, B., Goris, E., Marx, M., de Rijke, M.: PDL for ordered trees. Journal of Applied Non-Classical Logics 15, 115–135 (2005)

[3] Blackburn, P.: Nominal tense logic. Notre Dame Journal of Formal Logic 39, 56–83 (1993)

[4] Carpenter, B.: The Logic of Typed Feature Structures. In: Cambridge Tracts in Theoretical Computer Science 32, Cambridge University Press, Cambridge (1992)

[5] ten Cate, B.D., Marx, M.: Axiomatizing the logical core of xPath 2.0. In: Schwentick, T., Suciu, D. (eds.) ICDT 2007. LNCS, vol. 4353, pp. 134–148. Springer, Heidelberg (2006)

[6] Copestake, A.: Implementing Typed Feature Structure Grammars. CSLI (2000)

[7] Deutsch, A., Tannen, V.: Containment and integrity constraints for XPath. In: Lenzerini, M., Nardi, D., Nutt, W., Suciu, D. (eds.) Proceedings of the 8th International Workshop on Knowledge Representation Meets Databases, KRDB 2001 (2001)

[8] Ebbinghaus, H.D., Flum, J.: Finite Model Theory. Perspectives in Mathematical Logic. Springer, Heidelberg (1995)

[9] Gottlob, G., Koch, C., Pichler, R.: Efficient algorithms for processing XPath queries. In: VLDB 2002, pp. 95–102 (2002)

[10] Kay, M.: XPath 2.0. Programmer's Reference. Wiley Publishing, Indianapolis (2004)

[11] Kracht, M.: Tools and Techniques in Modal Logic. No. 142 in Studies in Logic. Elsevier, Amsterdam (1999)

[12] Kracht, M.: Mathematics of Language. Mouton de Gruyter, Berlin (2003)

[13] Marx, M.: XPath and Modal Logic of DAGs. In: Cialdea Mayer, M., Pirri, F. (eds.) TABLEAUX 2003. LNCS, vol. 2796, pp. 150–164. Springer, Heidelberg (2003)

[14] Marx, M.: Conditional XPath. ACM Transactions on Database Systems 30, 929–959 (2005)

[15] Miklau, G., Suciu, D.: Containment and equivalence for an XPath fragment. In: Proceedings of the 21st Symposium on Database Systems, pp. 65–76 (2002)

[16] Neven, F., Schwentick, T.: XPath containment in the presence of disjunction, dTDs, and variables. In: Calvanese, D., Lenzerini, M., Motwani, R. (eds.) ICDT 2003. LNCS, vol. 2572, pp. 312–326. Springer, Heidelberg (2002)

[17] Palm, A.: Tranforming Tree Constraints into Formal Grammars. The Expressivity of Tree Languages. PhD thesis, Universität Passau (1997)

[18] Ray, E.T.: Learning XML. O'Reilly, Sebastopol (2003)

[19] de Rijke, M., Marx, M.: Semantic characterisation of navigational XPath. Transactions of the ACM 34, 41–46 (2005)

[20] Rogers, J.: Studies in the Logic of Trees with Applications to Grammar Formalisms. PhD thesis, University of Delaware, Department of Computer & Information Sciences (1994)

[21] Stead, W.W., Hammond, W.E., Straube, M.J.: A Chartless Record–Is It Adequate? In: Proceedings of the Annual Symposium on Computer Application in Medical Care, pp. 89–94 (1982)

[22] Vardi, M., Wolper, P.: Automata theoretic techniques for modal logics of programs. Journal of Computer and Systems Sciences 32, 183–221 (1986)

Chapter 8
Adaptation of Ontological Knowledge from Structured Textual Data

Tonio Wandmacher, Ekaterina Ovchinnikova, Uwe Mönnich, Jens Michaelis, and Kai-Uwe Kühnberger

Abstract. This paper provides a general framework for the extraction and adaptation of ontological knowledge from new structured information. The cycle of this process is described starting with the extraction of semantic knowledge from syntactically given information, the transformation of this information into an appropriate format of description logic, and the dynamic update of a given ontology with this new information where certain types of potentially occurring inconsistencies are automatically resolved. The framework uses crucially certain tools for this incremental update. In addition to WordNet, the usage of FrameNet plays an important role, in order to provide a consistent basis for reasoning applications. The cycle of rewriting textual definitions into description logic axioms is prototypically implemented as well as the resolution of certain types of inconsistencies in the dynamic update of ontologies.

8.1 Introduction

Ontologies have been playing an important role for many applications in artificial intelligence, computational linguistics, and text technology. The tremendous success of the world wide web, large repositories of textual data used by the public sector and industry, and knowledge-based expert systems boosted the endeavor to

Uwe Mönnich
Seminar für Sprachwissenschaft, Universität Tübingen
e-mail: um@sfs.uni-tuebingen.de

Jens Michaelis
Fakultät für Linguistik und Literaturwissenschaft, Universität Bielefeld
e-mail: jens.michaelis@uni-bielefeld.de

Kai-Uwe Kühnberger · Tonio Wandmacher · Ekaterina Ovchinnikova
Institut für Kognitionswissenschaft, Universität Osnabrück
e-mail: firstname.lastname@uni-osnabrueck.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 129–153.
springerlink.com

develop standardized representations of background knowledge. The state-of-the-art for such a representation formalism are ontologies, commonly understood as a hierarchy (partial order) of concepts enriched by certain additional features, like relations holding between concepts.

There are several obvious challenges in ontology design and maintenance related to textual applications. In particular, with respect to distributed, highly interactive, and dynamic applications, e.g. in the WWW context, the need for a fast and reliable possibility to generate and to adapt ontologies is an absolutely necessary prerequisite for successful applications. Because hand-coded ontologies, developed by knowledge engineers, lack the possibility of rapid updates and changes and the pure generation is expensive and time-consuming in the first place, tools for the automatic acquisition and adaptation of ontologies need to be developed. In this context, different aspects can be distinguished:

- *Development of top-level ontologies:* usually top-level ontologies are hand-coded, represented in full first-order logic, and static, i.e. they describe the most general conceptual principles that are used to govern knowledge of a particular domain [44].
- *Knowledge acquisition for domain ontologies:* Domain ontologies describe conceptual hierarchies and relations between concepts in particular domains: As a de facto standard for domain ontologies, description logics (DLs) are commonly used [2]. A (semi-)automatic procedure for acquiring ontological knowledge is in the focus of this type of challenge [31].
- *Development and representation of lexical ontologies:* Lexical ontologies tend to be logically inconsistent and are therefore inappropriate for reasoning tasks. Consequently, interfaces are necessary to map lexical ontologies to domain and top-level ontologies [35].
- *Dynamic updates of domain ontologies:* Domain ontologies are subject to changes, updates, and conflict resolution. However, not all updates can be consistently added to the existing ontology. Therefore, conflict resolution strategies need to be developed to implement dynamic changes in an automatic way [37].

This article attempts to give a general framework for the whole cycle of incrementally updating existing ontologies with new information (cf. Figure 8.1 for a schematic diagram of this process). More precisely, given structured textual information, e.g. coded in annotation graphs (cf. Subsection 8.2.1), the first step is the extraction of semantic knowledge from these structures that can be used in an appropriate format for an update of an existing ontology (cf. Subsection 8.2.2 and Section 8.3). Some specific features of the presented system (including well-known tools that are used) are documented in Section 8.4. Our procedure for extracting ontological axioms from natural language definitions is built upon the approach presented in [45] which is discussed in section 8.3.1.

It is possible that the update makes the ontology inconsistent resulting in a knowledge base that is inappropriate for deducing inferences (a necessary prerequisite for many important applications in text technology). This problem can be addressed by the development of an integration engine that resolves occurring inconsistencies

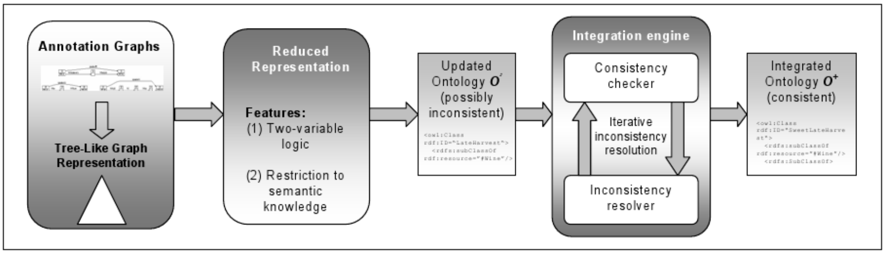

Annotation Graphs
Tree-Like Graph Representation
Reduced Representation
Features:
(1) Two-variable logic
(2) Restriction to semantic knowledge
Updated Ontology O' (possibly inconsistent)
<owl:Class rdf:ID="LateHarvest"> <rdfs:subClassOf rdf:resource="#Wine"/>
Integration engine
Consistency checker
Iterative inconsistency resolution
Inconsistency resolver
Integrated Ontology O^+ (consistent)
<owl:Class rdf:ID="SweetLateHarvest"> <rdfs:subClassOf rdf:resource="#Wine"/> <rdfs:SubClassOf>

Fig. 8.1 Architecture depicting the cycle of incrementally updating an existing ontology with new information.

(cf. Sections 8.5 and 8.6). The output is an integrated and consistent ontology in a standard format (like OWL). In this article, we focus primarily on the following challenge of ontology design: given an update of an ontology, tools and resolution strategies need to be used for its consistent extension and the possibility to deduce inferences. The task to extract ontological knowledge from given information resources is a preliminary step for addressing this problem.

8.1.1 Project Context

The results of this paper have emerged from work done in sub-project C2 “Adaptive Ontologies on Extreme Markup Structures” of the research unit FOR 437 “Text Technological Information Modelling”, a larger research endeavor funded by the German Research Foundation. The collaborative research has been carried out by the Universities of Bielefeld, Dortmund, Gießen, Tübingen, and Osnabrück and has been funded for six years. The overall goal of FOR 437 is the development of theoretical foundations and practical tools for text technological applications. With respect to the C2 sub-project the overall goal is to develop a formal framework that allows the representation of the process of extracting semantic knowledge from structured textual data, the incremental process of expanding existing conceptual background knowledge by this new semantic knowledge, and the adaptation of this expansion in case the added information is inconsistent with the existing ontology.

8.2 Theoretical Background

8.2.1 Annotation Graphs and Their Logical Representation

Recently, artificial intelligence and computational linguistics have tried to develop tools to extract semantic knowledge from syntactic information. In particular, from a text technological point of view, the general research perspective is to extract (semantic) information from annotated documents. Regarding this aim, some of the relevant annotation models used in this context are multilayer annotations, hyperlinks, discourse structure, or (classical) linguistic description levels.

An important aspect concerning annotation models is the development of a suitable logic system that is rich enough to represent linguistic information on multiple levels. The challenges of such architectures are twofold: first, the dynamic interaction between syntactic and semantic information must be represented and second, the efficiency of algorithms must be guaranteed.

Fortunately, in the case of annotation graphs (AGs) in the sense of [5], techniques from parameterized complexity theory can be exploited. This theory provides powerful tools for a detailed investigation of algorithmic problems. As it turns out, the concept of *treewidth* in the sense of [40], indicating the similarity of a graph or a relational structure with a tree, is a parameter which helps to show that many otherwise intractable problems become computable in linear time when restricted to tree-like inputs (cf. [32]).

The key feature of AGs is their abstraction from the diversity of concrete formats used for the transcription of text and speech. This feature makes them an ideal candidate for the comparison of different annotation systems, e.g., those currently developed by several linguistic collaborative research centers in Germany.

Translating these different representation schemes into the framework of AGs is a necessary prerequisite for the transfer of the pleasant computational properties of AGs to the original systems. This is particularly important in those circumstances where the natural data structure of the concrete markup system cannot be readily understood as describing trees.

A classical domain for the model of multilayered annotations is linguistics. Utterances of speakers can be considered from different perspectives: examples are syntactic, semantic, discourse and intonation aspects, just to mention some of them. Representing these types of data in one representation format yields overlapping hierarchies. Moreover, the resulting structures are naturally considered as graphs rather than trees.

These different types of data can be accommodated by representing AGs in logical form resulting in a structure of low descriptive complexity. Furthermore, logical representations are amenable to techniques from logical graph theory, especially in terms of Monadic Second-Order Logic (MSO) based on the work of Courcelle starting from [15]. While the logical approach towards annotation models provides a unified format for the syntactic level, it still has to be complemented with a component that serves to integrate syntactic with semantic structures, as, e.g., those given within the FrameNet account [41].

Clearly, an MSO representation of AGs is not yet in accordance with a format generally accepted for representing semantic knowledge. This format usually requires a reduced representation of the available knowledge in a form similar to a two-variable logic, namely, description logics (DLs), thus, in accordance with a reasonable logical representation of the semantic information provided by FrameNet.

A procedure of transforming the rich MSO representations into a two-variable representation where semantically irrelevant information is discarded and complex representations are simplified yields a core of conceptual information that can be used for a learning procedure on ontologies. Transformations of this sort by which labeled or tagged XML trees, e. g., are converted into patterns provided by a different language like HTML are critical nowadays to the core business in applications such as data exchange. A case in point is the use of the transformation language XPath in the approach [45], to be discussed below, in which annotated natural language definitions are translated into DL axioms.

A multitude of languages for transforming tree structures coded along the XML format was created with the particular needs of specific applications in mind. The language XPath, e. g., comes in a bewildering variety of dialects and this is also true of the more powerful Turing complete transformation language XSLT. It is thus highly desirable to characterize these transformation languages by suitable logics in order to understand their expressive power and their computational behavior. The way modal logic can be exploited for this purpose is documented by the contribution of Marcus Kracht to this volume [28]. He shows convincingly how complexity properties of XPath can be successfully addressed by an analysis of this XML language in terms of an adequate modal logic.

XPath serves to define expressions for absolute and relative paths in an *XML* document. The most general language for specifying such relations on trees which still has pleasant computational properties is MSO. MSO formulas with two free variables built on the appropriate signature have as models regular path languages on a (family of) tree(s). A path p from a node x to a node y satisfies such a formula if the sequence of tags or labels from node x to node y is a word in the regular language specified by this formula. Note that expressions in this logical language are not restricted to the domain of XML trees. By switching to this logical expression format an additional level of independence from the concrete details of XML syntax is achieved. Furthermore, the expressive power of MSO on trees is equivalent with the level of regular tree grammars and its expressive power on words/strings matches exactly the family of regular string grammars. The two steps of the logic-based approach described in this subsection consisting of the restriction to annotation structures of bounded treewidth and their subsequent transformation by means of logic-based path definitions, thus rely on two regular tools and retain in this way the desirable properties of the component formalisms.

It goes without saying that the path expressions on which the approach in [45] is based on are easily translated into MSO expressions. This can be done either directly or via the translation scheme provided in Kracht's contribution. The step from the modal logic in latter paper to MSO is then a routine matter.

For concreteness we have abstained in the ensuing discussion from rewriting the approach followed in [45] along the method sketched in this subsection. The informal presentation is intended to outline the theoretical background of such a transformation procedure. We hope it will be of help for potential readers who are not familiar with the techniques of model theoretic syntax.

8.2.2 Ontologies and Description Logics

Although there is no generally accepted definition of what an ontology is, from an abstract point of view, an ontology contains as a core terminological knowledge in form of hierarchically structured concepts. These concepts can be enriched by relations specifying constraints on them.

Certain standards allow to represent ontological knowledge in well-defined formal languages. In recent years, the fast development of the WWW has brought about a wide variety of standards for knowledge representation. Probably the most important existing markup language for ontology design is the *Web Ontology Language* (OWL) in its three different versions: OWL *Lite*, OWL *DL*, and OWL *Full*[1]. The mentioned OWL versions are hierarchically ordered, such that OWL *Full* includes OWL *DL*, and OWL *DL* includes OWL *Lite*. Consequently, they differ in their expressive power with respect to possible concept formations.

All versions of OWL are based on the logical formalism called *description logics* (DL) [2]. This family of logical representation formalisms was originally designed for the representation of terminological knowledge and reasoning processes. They can be characterized, roughly speaking, as subsystems of first-order predicate logic using at most two variables. In comparison to full first-order logic, description logics are – due to their restrictions concerning quantification – rather weak logics with respect to their expressive power. Therefore, DLs allow rather efficient reasoning.

A classical distinction in description logic is to separate knowledge about concepts and facts in two different data structures. Terminological knowledge about concepts is coded in the so-called *terminological box* (T-Box) whereas knowledge about facts is coded in the *assertion box* (A-Box).

A DL terminology contains terminological axioms that define concepts occurring in the domain of interest. In present paper, we consider $\mathscr{ALCN}$ terminologies. Let N_C and N_R be sets of *concept names* and *role names* respectively. A DL-terminology consists of *terminological axioms* (*TBox*) of the form $A \sqsubseteq C$ or $A \equiv D$ where $A \in N_C$ and C is a concept description specified in $\mathscr{ALCN}$-DL ($R \in N_R, n \in \mathbb{N}$) as follows:

$$C \rightarrow \top \mid \bot \mid A \mid \neg C \mid \forall R.C \mid \exists R.C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \leq nR \mid \geq nR$$

An assertion box is a finite set of facts $C(a)$ or $R(b,c)$ where C is a concept name, R is a relation name, and a,b, and c are individuals. $C(a)$ means that an individual a belongs to a concept C and $R(b,c)$ means that individuals b and c stand in relation R to each other.

The semantics of concepts and axioms is defined in the usual model-theoretic way (see [2]) in terms of an *interpretation function* $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$, where $\Delta^{\mathscr{I}}$ is a non-empty set of individuals and the function $\cdot^{\mathscr{I}}$ maps every concept name A to $A^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}}$ and every role name R to $R^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}} \times \Delta^{\mathscr{I}}$. $\mathscr{I}$ is called a *model* of a TBox $\mathscr{T}$ iff for every $C \sqsubseteq D \in \mathscr{T}$ we have $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$ and for every $C \equiv D \in \mathscr{T}$:

[1] see http://www.w3.org/TR/owl-features/

$C^{\mathscr{I}} = D^{\mathscr{I}}$. A concept description D *subsumes* C towards $\mathscr{T}$ ($\mathscr{T} \models C \sqsubseteq D$) iff $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$ for every model $\mathscr{I}$ of $\mathscr{T}$. A concept description C is called *satisfiable* in $\mathscr{T}$ iff there is a model M of $\mathscr{T}$ such that $C^M \neq \emptyset$. A terminology is *unsatisfiable* if it contains an unsatisfiable atomic concept.

8.3 Automatic Extraction of Ontological Knowledge from Texts

8.3.1 Existing Approaches in Ontology Learning

In the past few years, a variety of approaches has been presented that aim at extracting conceptual knowledge from unstructured and semi-structured data. These approaches are of growing importance in the ontology building process, since for many semantic web as for text technological applications the amount of available knowledge is crucial. Since these methods are unsupervised, their output is usually rather noisy.

So far, most of the approaches are light-weight from a logical point of view; they return structurally simple constructions such as concepts, instances, taxonomic relations and other general relations (e.g. `part-of` or `author-of`). Current methods basically make use of three strategies (or combinations of these):

1. *Distributional information:* The co-occurrence of terms within a given context or document is an important hint for their conceptual relatedness. Moreover, two terms will be similar in meaning if they tend to occur with the same neighbors (2nd order co-occurrence). Different distributional methods (e.g. collocation analysis or *Latent Semantic Analysis*, [16]) give a distance measure between two terms that can be used to represent semantic relatedness. For example, the term *dog* might appear to be more similar to *cat* than to *table*. Even though this cannot help labeling the type of relation, it gives a reliable clue that can be further used. Clustering techniques, for example, use this information to form sets of related terms. In hierarchical clustering procedures, these sets of terms are arranged in a hierarchical fashion. The hereby generated cluster hierarchy can be the base for a taxonomic structure, i.e. a hierarchy of concepts. Approaches that use this kind of strategy are for example described in [10] or [12].
2. *Lexico-syntactic patterns:* The second strategy basically relies on lexico-syntactic patterns, the so-called *Hearst* patterns [24]. Here, a text corpus is scanned for characteristic recurring word combinations, typically containing a semantic relation between two terms (e.g. [w_2, `such as` w_1] $\Rightarrow w_1 \sqsubseteq w_2$). These approaches however usually suffer from data sparsity, since many word combinations cannot be found in even large corpora. To cope with this fact, efforts have been made to harvest these patterns on the web, cf. [11].
3. *Syntactic and morphosyntactic information:* Finally linguistic structures like verb frames and modifier constructions can help extract conceptual relations. For example, it is easy to infer a subsumption relation between *car ferry* and *ferry*,

since *car* is here a modifier of *ferry* (cf. [7]) Moreover, from the analysis of dependency paths in syntactic derivations, reliable relations can be learned [26], other methods make use of predicate-argument relations [20]. For the extraction of nontaxonomic relations the analysis of selectional preferences of verbs can be very helpful, cf. [47].

Techniques based on these strategies can be found in many ontology learning systems, such as *Snowball* [1], *OntoLearn* [34], *OntoLT* [8], and *Text2Onto* [13]. Most of these systems are concerned with the extraction of the relevant terminology (from which they deduce the respective classes), with the derivation of subsumption relations and with some basic non-taxonomic relations.

Only very few approaches are aimed at extracting more expressive information in a (semi-)automatic way. One of the most elaborated works is detailed in [45].[2] Their methodology is based upon a syntactic analysis of definitional text segments, acquired from encyclopedia entries, and it works as follows.

Using a very efficient and robust dependency parser (Minipar, cf. [30]), a dependency tree is obtained from definitional text segments. By means of a set of manually defined transformation rules this tree is transformed into one or several axioms in OWL DL, as shown in the following example[3]:

$$\text{Disjunction: } NP_0 \texttt{ or } NP_1 \Rightarrow X \equiv (NP_0 \sqcup NP_1)$$
$$\text{Negation: } \texttt{not } V_0 NP_0 \Rightarrow X \equiv (\neg\exists V_0.NP_0)$$

The implemented rules comprise rather complex logical constructions such as negation, disjunction and number restriction, which implies that much of the expressivity of OWL DL is indeed exploited. By applying the methodology, axioms of considerable complexity can be created, as shown by the following example:

Example 1 (a).
Definition: *"A number is an abstract entity that represents a count or a measurement."*

The output of the Völker's approach yields the following:

Example 1 (b).
Axioms: $Number \equiv ((Entity \sqcap Abstract) \sqcap \exists represents.(Count \sqcup Measurement))$

However, this example also illustrates some of the major problems of the approach. While the axiom is syntactically quite complex, the symbols out of which it is composed are not grounded, i.e. their semantic interpretation is not provided. The grounding problem is probably one of the most difficult problems of artificial intelligence, and, as long as a complete theory of meaning has not been developed, it is obvious that some part of the semantic interpretation of ontological symbols

[2] Hereafter, the approach proposed in [45] is referred to as "Völker's approach".

[3] Alternatively, the transformation could be achieved by following the logic-based method described in the previous section

remains ungrounded, i.e. it can only be explained by means of natural language. However, in order to operate on output of the form presented in the example, it has to be mapped onto existing ontological resources, which is by itself a highly complex task (cf. [19]). The two main problems that have to be addressed are synonymy and ambiguity.

The **synonymy** problem is given by the fact that one ontological entity can usually be realized by several natural language expressions; as a consequence such expressions bearing the same (or very similar) meaning have to be identified and mapped onto each other. Let us look at the following example:

Example 2 (a).
Def.: *"A desert is a region receiving little precipitation." "Sahara is a desert."*
Query: *"Find all areas that have little rain."*

By applying the methodology described in [45] the definitory sentence and the query result in the following DL axioms:

Example 2 (b).
Axioms:
$Desert \equiv Region \sqcap \exists receives.(Little \sqcap Precipitation)$
$Sahara \sqsubseteq Desert$
$?Area \sqcap \exists have.(Little \sqcap Rain)$

It is easy to see that, with no further information given, it is impossible to answer the query. It is not known that *Rain* is a sort of *Precipitation* ($Rain \sqsubseteq Precipitation$); likewise the synonymous character of *receive* and *have* in this expression is not given. To address such problems the extracted axioms have to be connected with a large-scale ontology of general language such as the Princeton *WordNet* [21]. It comprises taxonomic information as well as standard semantic relations (e.g. meronymy), and it groups words as sets of synonyms (synsets). If each of the given words in an axiom would be mapped onto the respective synset, a part of the synonym problem could be solved.

However, this mapping process involves the second problem mentioned above, the problem of **ambiguity**: a given natural language expression can have more than one meaning, i.e. it can be mapped onto several ontological entities. This means that without a proper disambiguation procedure the intended meaning of a symbol cannot be specified. Disambiguation is a very active field of research in NLP, however up to now a robust and reliable disambiguation procedure has not been developed.

Moreover, ambiguity is not limited to atomic symbols, but can be extended to larger units. The definition of *number* above is a good example: Syntactically the given semantic interpretation is not the only possible solution; the following interpretation would be valid as well:

Example 3.
$Number \equiv ((Entity \sqcap Abstract) \sqcap \exists represents.Count) \sqcup Measurement$

Treating ambiguous constructions is a very difficult aspect for parsers, since the disambiguation on the syntactical level involves information from previous sentences as well as an important amount of world knowledge.

The Völker's approach has another problem concerning the treatment of different arguments of the same predicate. Consider Example 4 below. The relation names in this example are constructed from the predicates and the corresponding propositions attached. Thus, the relation names are strongly connected to a particular syntactic realization. Therefore the phrase *passes or rivers separate a mountain range from other mountain ranges* will have a completely different OWL representation ($\exists separate.(Mountain \sqcap Range) \sqcap \exists separate.(Other \sqcap Mountain \sqcap Range)$) than the definition in Example 4, while they are semantically equivalent. The existence of different superficial representations of a concept (in this case of the relation *separate*) can hinder or even prevent reasoning.

Example 4.
Definition: *A mountain range is a group of mountains separated from other mountain ranges by passes or rivers.*
Axiom: $Mountain \sqcap Range \equiv \exists group\_of.Mountains \sqcap$
$\exists separated\_from.(Other \sqcap Mountain \sqcap Range) \sqcap \exists separated\_by.(Pass \sqcup River)$

In addition to the problematic issues mentioned above, a rule-based approach like the one described here is usually unable to deal with non-compositional semantics, occurring for example in idiomatic expressions or certain types of adjective constructions. In such cases, manual intervention by an ontology engineer remains an essential part of the ontology development process.

8.4 Our Proposal

In order a) to deal with the natural language ambiguity and b) to be able to represent complex propositions with their argument structure we propose to "standardize" axiomatic representation of ontological knowledge extracted from textual resources.

A natural way for coping with the lexical ambiguity is mapping synonymous lexemes (e.g. *beverage* and *drink*) onto a unique representation. An ideal reference resource for such a mapping is WordNet [21], an electronic database containing lexical-semantic knowledge presented in a network-like structure. Nouns, verbs, adjectives, and adverbs are grouped in WordNet into sets of cognitive synonyms (synsets). In fact, synsets do not contain words but word senses. For example, the lexeme *board* participates in several synsets ($\{board1, plank\}$, $\{board2, committee\}$ etc.) which refer to its different senses. Every synset in WordNet is identified by a unique "id".

As mentioned in the previous section, disambiguating lexemes is not enough. One has to be able to bind arguments to a corresponding predicate and abstract from a concrete syntactic realization. We propose to use FrameNet (FN) frames as patterns for the representation of predicates. The FrameNet lexical resource is based on frame semantics [41][4]. The lexical meaning of predicates in FN is expressed in terms of frames which are supposed to describe prototypical situations in natural language. Every frame contains a set of roles (or frame elements, FEs) corresponding to the participants of the described situation. Predicates with similar semantics are assigned to the same frame, e.g. *to give* and *to hand over* refer to the Giving frame. Consider a FN annotation for Example 5(a) below. In this annotation, donor, recipient, and theme are roles in the frame Giving and *John*, *Mary*, and *a book* are fillers of these roles. The FN annotation generalizes across near meaning-preserving transformations, cf. Example 5(b).

Example 5.
(a) [John]$_{\text{DONOR}}$ [*gave*]$_{\text{GIVING}}$ [*Mary*]$_{\text{RECIPIENT}}$ [*a book*]$_{\text{THEME}}$.
(b) [*John*]$_{\text{DONOR}}$ [*gave*]$_{\text{GIVING}}$ [*a book*]$_{\text{THEME}}$ [*to Mary*]$_{\text{RECIPIENT}}$.

Additionally, semantic relations, such as inheritance, causation, precedence, are defined on frames, e.g. the Killing and Death frames are connected to each other with the causation relation. Frame elements of the connected frames are also related, e.g. victim in Killing is related to protagonist in Death. For some of the roles semantic types of the possible fillers are defined. For example, the role victim can be filled by an entity of the type *Sentient*. Semantic types are organized in a small hierarchy in FN.

The presence of the frame relations gives an essential advantage in using FN frames for the representation of predicates, because these relations serve as an additional knowledge source which can be exploited in reasoning (c.f. [42]). Let us consider Example 2 again. The predicates *receives* and *has* in the definition and in the query will be mapped on different frames in FN: the frame Receiving with roles recipient (*desert*) and theme (*little precipitation*) and the frame Possession with roles owner (*desert*) and possession (*little rain*) respectively. The frames Receiving and Possession are connected in FN trough the following path of relations: Receiving *inherits_from* Getting *precedes* Post_getting *inherits_from* Possession. Therefore it is possible to infer that the two frames under consideration refer to the same situation and map corresponding roles on each other.

In order to make inferences with FrameNet available one has to represent the resource in a logical form. At present there exists an OWL DL representation of FN in which frame names and frame element names are formalized as OWL concepts, see [42] for more details. If a role belongs to a frame in FN then the corresponding frame name is connected to the corresponding role name with the *usesFE* relation in OWL representation. We aimed at staying in line with the AI tradition of representing semantic roles as relations rather than as concepts [6]. Therefore we have

[4] cf. http://framenet.icsi.berkeley.edu

created an alternative OWL DL representation of FN such that role names are formalized as OWL relations connecting frame names and semantic types. Relations between frames are also represented as OWL relations. The following three axioms represent the KILLING and DEATH frames as well as the causation relation between them.

Example 6.

$Killing \sqsubseteq \exists Killing\_killer.\top \sqcap \exists Killing\_victim.Sentient \sqcap \exists causes.Death$
$Death \sqsubseteq \exists Death\_protagonist.Sentient$
$Killing\_victim \sqsubseteq Death\_protagonist$

To sum up, our approach consists in replacing lexically and syntactically ambiguous natural language expressions in axioms extracted from text on the basis of parsing and transformation rules (cf. section 8.3) with uniform representations using FrameNet frames and WordNet synsets as reference sources. Let us illustrate with Example 7 how the axioms resulting from the process described above look like.

Example 7 (a).

Definition: *"An alcoholic beverage is a drink containing ethanol."*
Query: *"Find all beverages in which alcohol is contained."*

Suppose we have axiom (1) extracted from the definition by the method presented in [46]. Additionally, we have a FN annotation of the original definition (2) and an OWL representation of the frame evoked by this definition (3).

Example 7 (b).

(1) $Alcoholic \sqcap Beverage \equiv Drink \sqcap \exists containing.Ethanol$
(2) *Alcoholic beverage is* $[a\ drink]_{\text{TOTAL}}\ [containing]_{\text{INCLUSION}}\ [ethanol]_{\text{PART}}$
(3) $Inclusion \equiv \exists Inclusion\_total.\top \sqcap \exists Inclusion\_part.\top$

The FN annotation (2) can be formalized in OWL as shown in axiom (4) below. In this axiom, words (*drink*, *ethanol*) have been already replaced with the corresponding WordNet synsets.

(4) $Alcoholic\_Beverage\_Inclusion \equiv Inclusion \sqcap Inclusion\_total.\{Drink\} \sqcap \exists Inclusion\_part.\{Ethanol\}$

The amalgamation of axioms (1) and (4) results in (5), where words are replaced with synsets and an additional relation *evokes* is introduced in order to express the connection between the defined concept and the evoked frame.

(5) $\{Alcoholic\} \sqcap \{Drink\} \equiv \{Drink\} \sqcap \exists evokes.Alcoholic\_Beverage\_Inclusion$

Let us now demonstrate how reasoning will work with axiom (5). Suppose we have such additional facts as (6) and (7) in our ontology, claiming that beer is an alcoholic beverage and ethanol is a kind of alcohol.

(6) $\{Beer\} \sqsubseteq \{Alcoholic\} \sqcap \{Drink\}$
(7) $\{Ethanol\} \sqsubseteq \{Alcohol\}$

The query *Find all beverages in which alcohol is contained* can be represented as follows.

(8) $X \sqsubseteq \{Drink\} \sqcap \exists evoke\_frame.X\_Inclusion$

(9) $X\_Inclusion \equiv Inclusion \sqcap \exists Inclusion\_total.\{Drink\} \sqcap$
$\exists Inclusion\_part.\{Alcohol\}$

For answering the query one has to find concepts in the ontology which are subsumed by the right part of (8). It is easy to see that in our example the target atomic concept is *Beer*.

8.5 Axiom Rewriting Procedure

8.5.1 Transforming Textual Input into OWL Axioms

Figure 8.2 shows the axiom rewriting procedure which is informally described in the previous section. A textual concept definition from a thesaurus is treated in two parallel processing lines. On the one hand, transformation rules are, in general, applied to annotated analysis trees and, in our special case, to the output of the Minipar parser as described in section 8.3. This module outputs OWL axioms preserving lexemes and syntactic structure of the original definition. Then, lexemes in the definition are disambiguated and mapped onto WordNet synsets. This task can be performed by the *SenseLearner* toolkit [33], a freely available package offering to disambiguate text with minimal supervision. The output of this processing line then consists of OWL axioms with WordNet synsets expressing concepts.

The second processing line concerns the annotation of the original definition with FrameNet frames. In order to achieve this step automatically we use the *Shalmaneser* tool [18], developed at *Saarland University*. *Shalmaneser* is a robust and shallow semantic parser assigning semantic roles to words in a text. In principle, it can be run with any set of senses and semantic roles, however we made use of the inbuilt FrameNet-based classifier. The output of *Shalmaneser* is a multi-level (i.e. syntactic and semantic) annotation of a given text. As annotation format SALSA/TIGER XML is used, which is expressive enough to describe various linguistic layers and also offers the possibility of inter-layer references [17].

Given a concept name *C*, the FN annotation for the definition of *C* is converted into the OWL format by using rule 1 as described below. Syntactic transformation rules as described in section 8.3 are again applied to the role fillers. Propositions are removed from the corresponding constituents.[5].

[5] In Example 4, fillers are expressed by prepositional phrases:
[a group of mountains]$_{\text{PART\_1}}$ [separated]$_{\text{SEPARATION}}$ [from other mountain ranges]$_{\text{PART\_2}}$ [by passes or rivers]$_{\text{AGENT}}$. We remove prepositions keeping the following roles: $\exists Separation\_part\_2.(Other \sqcap Mountain \sqcap Range)$ and $\exists Separation\_agent.(Pass \sqcup River))$

Rule 1. Translating FN annotation to OWL
Input: $\{[\mathit{filler\_1}]_{\mathrm{FE\_NAME\_1}}, \ldots, [\mathit{filler\_n}]_{\mathrm{FE\_NAME\_N}}$,
$[\mathit{frame\_evoking\_predicate}]_{\mathrm{FRAME\_NAME}}\}$
Output: $C\_Frame\_name \equiv Frame\_name \sqcap \exists fe\_name\_1.Trans\_filler\_1 \sqcap \ldots$
$\sqcap \exists fe\_name\_n.Trans\_filler\_n$

An OWL axiom obtained in the first processing line has the following form:

$$C \sqsubseteq D \sqcap \exists frame\_evoking\_predicate\_Y.filler\_1 \sqcap \ldots \exists X\_n.filler\_n,$$

where D is an arbitrary concept, Y is empty or has the form $prep$ ($prep$ stands for a preposition) and X_n has the form $frame\_evoking\_predicate\_Y$ or $prep$. Then, the amalgamation rule looks like the following:

Rule 2. Amalgamation rule
Input: $C \sqsubseteq D \sqcap \exists frame\_evoking\_predicate\_Y.filler\_1 \sqcap \ldots \exists X\_n.filler\_n$
Output: $C \sqsubseteq D \sqcap \exists evokes.C\_Frame\_name$

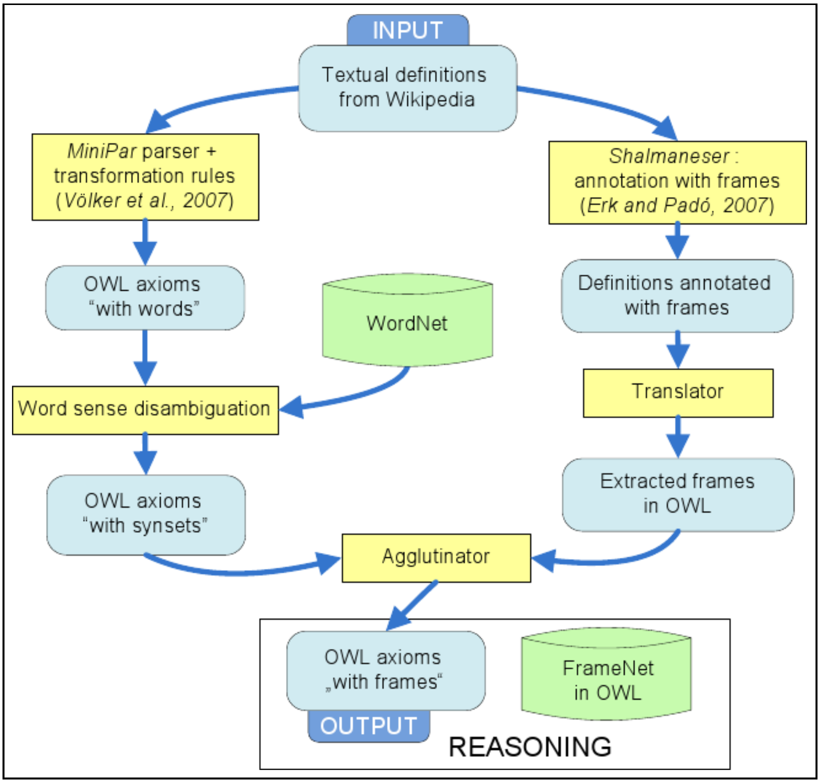

INPUT
Textual definitions from Wikipedia
MiniPar parser + transformation rules (Völker et al., 2007)
Shalmaneser : annotation with frames (Erk and Padó, 2007)
OWL axioms "with words"
WordNet
Definitions annotated with frames
Word sense disambiguation
Translator
OWL axioms "with synsets"
Extracted frames in OWL
Agglutinator
OWL axioms „with frames"
OUTPUT
FrameNet in OWL
REASONING

Fig. 8.2 Axiom rewriting procedure.

We have experimented with a set of 120 axioms created using the method described in [45].[6] Some of the axioms have been automatically created from definitions which have been selected from a fishery glossary of the Food and Agriculture Organization and extracted from *Wikipedia* in order to extend definitions for some classes of the *Proton* ontology, see [45] for more details. In the following we discuss several examples of the rewritten axioms and compare them to the original axioms created by the Völker's approach.

Example 8.
(a) *Ministry*
Def.: *A ministry is a department of a government, led by a minister.*
Orig.ax.: $Ministry \equiv Department \sqcap \exists of.Government \sqcap \exists led\_by.Minister$
Rewr.ax.: $Ministry \equiv \{Department\} \sqcap \exists of.\{Government\} \sqcap$
$\exists evokes.Ministry\_Leadership$
$Ministry\_Leadership \equiv Leadership \sqcap \exists Leadership\_Leader.\{Minister\}$
$\sqcap \exists Leadership\_Jurisdiction.\{Ministry\}$
(b) *Desert*
Def.: *In geography, a desert is a landscape form or region that receives little precipitation.*
Orig.ax.: $Desert \equiv ((Landscape \sqcap Form) \sqcup Region) \sqcap$
$\exists receives.(Little \sqcap Precipitation)$
Rewr.ax.: $Desert \equiv ((\{Landscape\} \sqcap \{Form\}) \sqcup \{Region\}) \sqcap$
$\exists evokes.Desert\_Receiving \sqcap \exists evokes.Desert\_Biological\_Area$
$Desert\_Receiving \equiv Receiving \sqcap \exists Receiving\_Theme.(\{Little\} \sqcap$
$\{Precipitation\}) \sqcap \exists Receiving\_Place.\{Geography\}$
$Desert\_Biological\_Area \equiv Biological\_Area \sqcap$
$\exists Biological\_Area\_Locale.\{Desert\} \sqcap$
$\exists Biological\_Area\_Relative\_location.\{Geography\}$
(c) *Spring*
Def.: *A spring is a point where groundwater flows out of the ground, and is thus where the aquifer surface meets the ground surface.*
Orig.ax.: $Spring \equiv Point \sqcap \exists groundwater\_flows\_out\_of.Ground \sqcap$
$\exists aquifer\_surface\_meets.(Ground \sqcap Surface)$
Rewr.ax.: $Spring \equiv \{Point\} \sqcap \exists evokes.Spring\_Fluidic\_motion \sqcap$
$\exists evokes.Spring\_Congregating$
$Spring\_Fluidic\_motion \equiv Fluidic\_motion \sqcap$
$\exists Fluidic\_motion\_Source.\{Ground\} \sqcap$
$\exists Fluidic\_motion\_Fluid.\{Groundwater\}$
$Spring\_Congregating \equiv Congregating \sqcap$
$\exists Congregating\_Configuration.(\{Ground\} \sqcap \{Surface\})$

While in simple cases like Example 7 (*alcoholic beverage*) the proposed procedure works rather well, for more complicated definitions it can produce mistakes. For definition (a) the *Leadership* frame was assigned correctly. However, no frame

[6] Special thanks to Johanna Völker for providing us with these data!

was assigned to the phrase *a department of a government*. In (b) the *Receiving* frame was assigned correctly, but the *Receiver* role which should be filled by *Desert* remained unfilled. Moreover, the phrase *in geography* was falsely defined as describing a location. In (c) the *Fluidic_motion* frame is in the right position, but the *Congregating* frame was incorrectly assigned to the predicate *meet* and the adverb *where* was not determined as pointing to a location. In spite of the mistakes, the given axioms support paraphrasing and allow answering queries which can not be treated on the basis of the original axioms. Some examples of the supported types of paraphrasing and queries which cannot be answered by reasoning with the original axioms are listed below:

Example 9.

Inclusion frame: *X contains Y, Y is contained in X, X includes Y, X has Y in it*
Query for Example 7: *Find all beverages in which alcohol is contained.*

Leadership frame: *X is a head of Y, Y is lead by X*
Query for Example 8 (a): *Who is the head of a ministry?*

Receiving frame: *X receives Y, X gets Y*
Query for Example 8 (b): *Find all areas which get little rain.*

Fluidic_motion frame: *X flows out from Y, X streams out of Y, X comes from Y*
Query for 8 (c): *How is a point called where the groundwater does come from?*

8.5.2 Discussion

In the previous sections, we have shown that the presented approach has some potential in integrating and standardizing axiomatic knowledge extracted from textual resources. It grounds automatically extracted concept and relation names by providing a reference to such standard lexical-semantic resources as WordNet and FrameNet which generalize over synonyms and allow additional inferences by using lexical-semantic relations. However, this approach obviously has several limitations which are connected both to technical issues and to some general problems. In the following, we briefly discuss some of these limitations.

8.5.2.1 Lexical Semantics and Conceptual Knowledge

As mentioned in section 2.2, description logics are based on model-theoretic semantics which in particular enables reasoning over DL knowledge bases. Natural language definitions in thesauri are usually formulated without taking into account (possibly unintended) inferences and logical contradictions. Therefore a straightforward mapping between lexical and model-theoretic semantics may cause unexpected logical consequences making the resulting ontology inconsistent, cf.

Section 6. Moreover, natural language definitions are not always ontologically sound. There are ontology engineering principles which are based on philosophical insights and practical experience, see for example [22]. For example, stating something like $Student \sqsubseteq Person$ (a part of a definition which can be easily found in a thesaurus) can be problematic from a formal ontological point of view, because *Student* is a role while *Person* is a natural kind.

8.5.2.2 FrameNet Frames as Predicate Patterns

FrameNet, as any other lexical semantic resource, is work in progress. It cannot be considered as a gold standard. In previous studies it was found that low coverage of the current version of FN makes its successful application to the real textual data difficult, cf. [9]. In addition, FN suffers from conceptual inconsistency and a lack of axiomatization which can prevent appropriate inferences, cf. [4]. For example, fundamental conceptual relations such as *part-of* or *causation* have no logical axiomatization in FN. However, the resource has been constantly improved and extended, see for example [9, 39].

8.5.2.3 Technical Problems

The described procedure relies on several NLP modules performing linguistically rather complex tasks, which are usually error-prone. However, mistakes arising on every processing step percolate to the next level and therefore multiply in the final output. For example, the *Minipar*[7] dependency parser applied has problems in treating such complex linguistic phenomena as ambiguity, anaphora, quantification, compounds, cf. [45]. The *Shalmaneser* tool that we have used for annotating definitions with FrameNet frames is at present also far from being perfect. Concerning 120 definitions selected for our experiment, approximately 320 frames were assigned correctly, 200 predicates were not annotated, 60 frames were assigned incorrectly, 40 frame elements were missing or were assigned incorrectly.

8.6 Adaptivity

As mentioned in section 8.1, it is one of the main features of a formal ontology to allow the inference of new information which is implicitly coded in a set of explicitly given axioms, i.e. automatic reasoning is supported. In order to support logical reasoning, an ontology has to be logically consistent.

Adding new axioms to an already existing knowledge base may cause logical contradictions and therefore disable reasoning. This concerns especially axioms automatically extracted from text, because, as mentioned in section 8.5.2, definitions in thesauri are usually formulated without taking into account resulting inferences. Since the purpose of our system is to expand ontologies through extracting axioms from textual resources we focus in particular on supporting consistency. The

[7] *http://www.cs.ualberta.ca/≫lindek/minipar.htm*

described system mostly extracts terminological axioms. Thus, in the following we concentrate on a procedure that resolves inconsistencies in terminological knowledge bases.

8.6.1 Terminological Inconsistency

The notion of terminological inconsistency has several meanings. In [23], for example, three types of inconsistency are distinguished:

- *Structural inconsistency* is defined with respect to the underlying representation language. A knowledge base is structurally inconsistent, if it contains axioms violating the syntactical rules of the representation language (for example, OWL DL).
- *Logical inconsistency* is defined on the basis of formal semantics of the knowledge base. An ontology is logically inconsistent, if the ontology has no model.
- *User-defined* inconsistency is related to application context constraints defined by the user.

In this paper, we consider logical inconsistency only. In particular, the main focus lies on contradicting unsatisfiable terminologies.

Definition 1. A terminology T is unsatisfiable if there exists a concept C that is defined in T and is unsatisfiable.

Informally, Definition 1 implies that an inconsistent ontology necessarily contains logical contradictions. An ontology can be inconsistent only if its underlying logic allows negation. Ontologies share this property with every logical system (like, for example, first-order logic). In practice, logical inconsistency can be caused by several reasons. For example, errors in the automatic ontology learning procedure or mistakes of the ontology engineer can generate unintended contradictions.

Another type of logical inconsistency is related to polysemy. If an ontology is learned automatically, then it is hardly possible to distinguish between senses of words that represent different concepts in texts. Suppose, the concept *tree* is declared to be a subconcept both of *plant* and of *data structure* (whereas *plant* and *data structure* are disjoint concepts). To cope with this, both the *structure* and the *plant* sense of *tree* will have to be described in the ontology by using distinct identifiers (e.g. *TreePlant*, *TreeStructure*).

Finally, there is a set of problems related to generalization mistakes, i.e. ontological definitions which are too general to take into account exceptions or too narrow, so that concepts are not reasonably distinguished from one another. Let us consider Example 10. Suppose that the ontology contains the following facts:

Example 10.

Terminology

(1) $Bird \sqsubseteq CanFly$	(*Birds are creatures that can fly.*)
(2) $CanFly \sqsubseteq CanMove$	(*If a creature can fly then it can move.*)
(3) $Canary \sqsubseteq Bird$	(*Canary is a bird.*)

New axiom

(4) $Penguin \sqsubseteq Bird \sqcap \neg CanFly$ (*Penguin is a bird and cannot fly.*)

The statement *all birds can fly* (1) in Example 10 is too general. If an exception *penguin* (a bird that cannot fly) is added, the terminology becomes unsatisfiable.

In the past few years, a number of new approaches to automatic ontology debugging have been suggested. A technique to find a minimal set of axioms that is responsible for inconsistencies in an ontology was first proposed in [3] and further developed, for example, in [43]. Several other debugging methods are concerned with explanation services that are integrated into ontology developing tools [23, 48]. However, these approaches either do not provide solutions of how to fix the discovered contradictions or just propose to remove a problematic part of an axiom, although removed parts of axioms can result in a loss of information. Considering Example 10 again, if the concept *CanFly* is removed from axiom 1, then the entailments $Bird \sqsubseteq CanMove$ and $Canary \sqsubseteq CanFly$ are lost.

The second type of solutions contains approaches that use several well-known techniques from non-monotonic reasoning, like default sets [25] or epistemic operators [27]. Unfortunately, these approaches go beyond the expressive power of description logics. The disadvantage is that standard DL-reasoners cannot be used easily for these extensions.

Finally, different techniques for rewriting problematic axioms were proposed [29, 36]: Besides the detection of conflicting parts of axioms, a concept is constructed that replaces the problematic part of the chosen axiom. [29] extend the tableau-based algorithm in order to find sets of axioms causing inconsistency and the set of "helpful" changes that can be performed to debug the ontology. This approach keeps the entailment $Bird \sqsubseteq CanMove$, but not $Canary \sqsubseteq CanFly$ in Example 10. An approach to resolve overgeneralized concepts conflicting with exceptions as well as to treat polysemous concept names is presented in [36]. Besides rewriting problematic axioms, a split of an overgeneralized concept C into a more general concept (not conflicting with exceptions) and a more specific one (capturing the original semantics of C) is proposed. The entailment $Canary \sqsubseteq CanFly$ is also preserved.

8.6.2 Adaptation Procedure

In this section, we informally describe an approach to resolve inconsistent ontologies that is based on the ideas technically introduced in [36] and developed in [37]. Given an inconsistently extended ontology we want to change it automatically in order to obtain a consistent one, according to the following criteria:

- The performed changes have to be relevant and intuitive.
- The changed ontology is formalized in a description logic language.
- As few pieces of information as possible are removed from the ontology.

In general, accidental mistakes cannot be fixed automatically. But the polysemy problem can be resolved by renaming concepts which have polysemous names. Furthermore, overgeneralized concepts can be redefined so that problematic pieces of information will be deleted from their definitions.

8.6.2.1 Adaptation Algorithm

The proposed approach treats inconsistent ontologies that are extended with additional axioms conflicting with the original knowledge base. Given a consistent ontology O (possibly empty) the procedure adds a new axiom A to O. If $O^+ = O \cup \{A\}$ is inconsistent then the procedure tries to find a polysemy or an overgeneralization and repairs O^+. Suppose that the new axiom A represents a definition of a concept C. Regarding the terminological component, O^+ is inconsistent if a subconcept C' of the newly introduced or newly defined concept C is unsatisfiable. Unfortunately, it is impossible to distinguish between accidental mistakes, polysemy problem and overgeneralization in a strict logical sense. Our algorithm inspects the definitions of the unsatisfiable concept C', tries to identify overgeneralized concepts subsuming C' and regeneralize these concepts. If no overgeneralized concepts have been found, then the algorithm defines which concepts are suspected to be polysemous and renames these concepts (by default or given the consent of the user).

8.6.2.2 Regeneralization of Overgeneralized Concepts

We will illustrate the regeneralization of the overgeneralized concepts with the ontology given in Example 10. Since the definition of the concept *Bird* is overgeneralized, it needs to be rewritten. We wish to retain as much information as possible in the ontology. The following solution is proposed:

Example 11. Adapted ontology from Ex. 10

$(1)\ Bird \sqsubseteq CanMove$ (*Birds are creatures that can move.*)
$(2)\ CanFly \sqsubseteq CanMove$ (*If a creature can fly then it can move.*)
$(3)\ Canary \sqsubseteq FlyingBird$ (*Canary is a flying bird.*)
$(4)\ Penguin \sqsubseteq Bird \sqcap \neg CanFly$ (*Penguin is a bird and cannot fly.*)
$(5)\ FlyingBird \sqsubseteq Bird \sqcap CanFly$ (*Flying birds are birds that can fly.*)

We want to keep in the definition of the concept *Bird* (subsuming the unsatisfiable concept *Penguin*) a maximum of information that does not conflict with the definition of *Penguin*. The conflicting information is moved to the definition of the new concept *FlyingBird*, which is declared to subsume all former subconcepts of *Bird* (such as *Canary* for example).

The example below represents a case where two overgeneralized definitions of the same concept conflict with each other.

Example 12.

Terminology

(1) $Child \sqsubseteq \forall likes.Icecream$ (*Children only like ice cream.*)
(2) $IceCream \sqsubseteq Sweets$ (*Ice cream is a sweet.*)
(3) $Cholocate \sqsubseteq Sweets$ (*Chocolate is a sweet.*)
(4) $IceCream \sqsubseteq \neg Chocolate$ (*Ice cream and chocolate are disjoint concepts.*)

New axiom

(5) $Child \sqsubseteq \forall likes.Chocolate$ (*Children only like chocolate.*)

In Example 12, the definitions of *Child* (*Children only like ice cream* and *Children only like chocolate*) are overgeneralized. *IceCream* and *Chocolate* being disjoint concepts produce a conflict. It seems to be an intuitive solution to replace these concepts by their least common subsumer (see [14]) *Sweets*. Furthermore, it is plausible to claim that children only like sweets without specifying it precisely, as described below:

Example 13. Adapted ontology from Ex. 12

(1) $Child \sqsubseteq \forall likes.Sweets$ (*Children only like sweets.*)
(2) $IceCream \sqsubseteq Sweets$ (*Ice cream is a sweet.*)
(3) $Cholocate \sqsubseteq Sweets$ (*Chocolate is a sweet.*)
(4) $IceCream \sqsubseteq \neg Chocolate$ (*Ice cream and chocolate are disjoint concepts.*)

Let us now describe informally the regeneralization procedure (see [37] for more details). Suppose that a) X is an unsatisfiable concept in the terminology T, b) X is defined in T by the definitions A and B, c) A and B which are logically conflicting (their conjunction is unsatisfiable in T). Then the following options can be distinguished:

1. A and B are disjoint concepts having common subsumers (Example 12):

   The solution in this case is to replace the definitions A and B of X by their least common subsumer.

2. A is defined in T and some definition D_A of A conflicts with B (Example 10):

   This case is considered as the overgeneralization of A. The definition D_A has to be revised as follows: (a) D_A is replaced by its minimal specific superdescription that does not conflict with B; (b) a new concept A' is added to T as a subconcept of A and D_A; (c) A is replaced by A' in the definitions of all its subconcepts except in the definition of X.

3. A and B are defined in T, a definition D_A of A conflicts with B, and a definition D_B of B conflicts with A:

In this case there is no unique solution. On the one hand the concept X is suspected to be polysemous. Here, the preferred solution is to split the definition of X and rename X as, for example, X_1 and X_2. On the other hand we may face two overgeneralized concepts, one or both definitions of which can be adapted in the way described in the previous option

4. Otherwise:

   The concept X is suspected to be polysemous as in the previous option.

In [38], a prototypical implementation of the idea of splitting overgeneralized concepts in $\mathscr{ALE}$-DL was discussed. This implementation was tested on the famous wine-ontology[8] that was automatically extended with new classes extracted from text corpora with the help of the *Text2Onto*[9] tool. Several cases of overgeneralization were detected and correctly resolved[10]. In the near future we plan to experiment with the adaptation of the *Proton* ontology provided by Völker et al. (cf. [45]) to the axioms extracted automatically from the thesauri, see section 5.

8.7 Conclusions and Future Work

This paper provides a general framework for the extraction and adaptation of ontological knowledge from new structured information. The cycle of this process is described starting with the extraction of semantic knowledge from syntactically given information, the transformation of this information into an appropriate format of description logic, and the dynamic update of a given ontology with this new information where certain types of potentially occurring inconsistencies are automatically resolved. The framework uses crucially certain tools for this incremental update. In addition to WordNet, the usage of FrameNet plays an important role, in order to provide a consistent basis for reasoning applications. The cycle of rewriting textual definitions into DL axioms is prototypically implemented as well as the resolution of certain types of inconsistencies in the dynamic update of ontologies. Future work will be devoted to refinements of the framework, in particular, with respect to the axiom generation and the resolution of inconsistencies. Furthermore, the framework needs to be evaluated whether it is practically applicable to large knowledge bases and to updates with bigger numbers of new axioms.

Acknowledgements. This research was partially funded by the grant MO 386/3-4 of the German Research Foundation, a sub-project of research unit FOR 437 “Text Technological Information Processing”.

[8] http://www.w3.org/TR/owl-guide/wine.owl

[9] http://ontoware.org/projects/text2onto/

[10] For example, the class *LateHarvest* originally defined to be a sweet wine was claimed to be overgeneralized after an exception *RieslingSpaetlese* which was defined to be a late harvest wine and a dry wine appeared.

References

[1] Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proc. of the 5th ACM International Conference on Digital Libraries (ACM DL), pp. 85–94 (2000)

[2] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York (2003)

[3] Baader, F., Lutz, C., Milicic, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms: First results. In: Proceedings of the 2005 International Workshop on Description Logics (DL2005). CEUR-WS (2005)

[4] Bejan, C., Harabagiu, S.: A Linguistic Resource for Discovering Event Structures and Resolving Event Coreference. In: Proc. of LREC 2008 (2008)

[5] Bird, S., Liberman, M.: A formal framework for linguistic annotation. Speech Communication 33(1), 23–60 (2001)

[6] Brachman, R.J., Schmolze, J.: An overview of the KL-ONE knowledge representation system. Cognitive Science 9(2), 171–216 (1985)

[7] Buitelaar, P., Olejnik, D., Hutanu, M., Schutz, A., Declerck, T., Sintek, M.: Towards ontology engineering based on linguistic analysis. In: Proc. of the Lexical Resources and Evaluation Conference, LREC (2004)

[8] Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé plugin for ontology extraction from text based on linguistic analysis. In: Proc. of the 1st European Semantic Web Symposium, ESWS (2004)

[9] Cao, D.D., Croce, D., Pennacchiotti, M., Basili, R.: Combining word sense and usage for modeling frame semantics. In: Proc. of STEP 2008 (2008)

[10] Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proc. of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 120–126 (1999)

[11] Cimiano, P., Staab, S.: Learning by googling. SIGKDD Explorations 6(2), 24–33 (2004)

[12] Cimiano, P., Staab, S.: Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In: Proc. of the ICML Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany (2005)

[13] Cimiano, P., Völker, J.: Text2Onto - a framework for ontology learning and data-driven change discovery. In: Proc. of the 10th International Conference on Applications of Natural Language to Information Systems, NLDB (2005)

[14] Cohen, W.W., Borgida, A., Hirsh, H.: Computing least common subsumers in description logics. In: Rosenbloom, P., Szolovits, P. (eds.) Proc. of the 10th Nat.Conf. on Artificial Intelligence (AAAI), pp. 754–761. AAAI Press, Menlo Park (1993)

[15] Courcelle, B.: The monadic second-order logic of graphs I: Recognizable sets of finite graphs. Information and Computation 85(1), 12–75 (1990)

[16] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by Latent Semantic Analysis. JASIS 41(6), 391–407 (1990)

[17] Erk, K., Pado, S.: A powerful and versatile xml format for representing role-semantic annotation. In: Proceedings of LREC 2004, Lisbon, Portugal (2004)

[18] Erk, K., Pado, S.: Shalmaneser - a toolchain for shallow semantic parsing. In: Proc. of the Lexical Resources and Evaluation Conference, LREC (2006)

[19] Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)

[20] Faure, D., Nédellec, C.: ASIUM: Learning subcategorization frames and restrictions of selection. In: Proc. of the 10th Conference on Machine Learning, ECML (1998)

[21] Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)

[22] Guarino, N.: Formal ontology and information systems, pp. 3–15. IOS Press, Amsterdam (1998)

[23] Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 353–367. Springer, Heidelberg (2005)

[24] Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of the 14th Int. Conf. on Computational Linguistics, Nantes, France (1992)

[25] Heymans, S., Vermeir, D.: A defeasible ontology language. In: Meersman, R., Tari, Z. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 1033–1046. Springer, Heidelberg (2002)

[26] Katrenko, S., Adriaans, P.: Learning patterns from dependency paths. In: Proceedings of the international workshop Ontologies in Text Technology, OTT, Osnabrück (2006)

[27] Katz, Y., Parsia, B.: Towards a nonmonotonic extension to OWL. In: Proceedings of the Workshop OWL: Experiences and Directions (2005)

[28] Kracht, M.: Modal logic foundations of markup structures in annotation systems. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling ofInformation and Markup Languages. Contributions to Language Technology, Springer, Heidelberg (2009)

[29] Lam, S.C., Pan, J.Z., Sleeman, D.H., Vasconcelos, W.W.: A fine-grained approach to resolving unsatisfiable ontologies. In: Web Intelligence, pp. 428–434 (2006)

[30] Lin, D.: Dependency-based evaluation of MINIPAR. In: Proceedings of the Workshop on the Evaluation of Parsing Systems (1998)

[31] Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems 16(2), 72–79 (2001)

[32] Michaelis, J., Mönnich, U.: Towards a logical description of trees in annotation graphs. LDV Forum 22(2), 68–83 (2007)

[33] Mihalcea, R., Csomai, A.: Senselearner: Word sense disambiguation for all words in unrestricted text. In: Proceedings of the 43nd ACL, Ann Arbor, MI (2005)

[34] Navigli, R., Velardi, P.: Learning domain ontologies from document warehouses and dedicated websites. Computational Linguistics 30(2), 151–179 (2004)

[35] Oltramari, O., Prevot, L., Borgo, S.: Theoretical and practical aspects of interfacing ontologies and lexical resources. In: Proc. of OntoLex 2005 (2005)

[36] Ovchinnikova, E., Kühnberger, K.U.: Adaptive ALE-TBox for extending terminological knowledge. In: 19th Australian Joint Conference on Artificial Intelligence, pp. 1111–1115 (2006)

[37] Ovchinnikova, E., Kühnberger, K.U.: Automatic ontology extension: Resolving inconsistencies. GLDV-Journal for Computational Linguistics and Language Technology 22(2), 19–33 (2007)

[38] Ovchinnikova, E., Wandmacher, T., Kühnberger, K.U.: Solving terminological inconsistency problems in ontology design. International Journal of Interoperability in Business Information Systems (IBIS) 2(1), 65–79 (2007)

[39] Ovchinnikova, E., Vieu, L., Oltramari, A., Borgo, S., Alexandrov, T.: Data-driven and ontological analysis of framenet for natural language reasoning. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association, ELRA (2010)

[40] Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. Journal of Algorithms 7(1), 309–322 (1986)

[41] Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., Scheffczyk, J.: FrameNet II: Extended Theory and Practice. International Computer Science Institute (2006)

[42] Scheffczyk, J., Baker, C.F., Narayanan, S.: Ontology-based reasoning about lexical resources. In: Proc. of OntoLex 2006, Genoa, Italy (2006)

[43] Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, pp. 355–362 (2003)

[44] Sowa, J.: Knowledge Representation. Logical, Philosophical, and Computational Foundations. Brooks/Cole, CA (2000)

[45] Völker, J., Hitzler, P., Cimiano, P.: Acquisition of OWL DL axioms from lexical resources. In: Franconi, E., Kifer, M., May, W. (eds.) Proceedings of the European Semantic Web Conference, Innsbruck, Austria, pp. 670–685 (2007)

[46] Völker, J., Haase, P., Hitzler, P.: Learning Expressive Ontologies. In: Ontology Learning and Population: Bridging the Gap between Text and Knowledge, vol. 167, pp. 45–69. IOS Press, Amsterdam (2008)

[47] Wagner, A.: Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In: Proc. of the ECAI Workshop on Ontology Learning, Berlin, Germany (2000)

[48] Wang, H., Horridge, M., Rector, A.L., Drummond, N., Seidenberg, J.: Debugging OWL-DL ontologies: A heuristic approach. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 745–757. Springer, Heidelberg (2005)

Part IV
Multidimensional Representations: Solutions for Complex Markup

Chapter 9
Ten Problems in the Interpretation of XML Documents

C.M. Sperberg-McQueen and Claus Huitfeldt

Abstract. We seek to develop methods for making the meaning of markup explicit using notations like first-order predicate calculus. In the context of that work, a number of problems have arisen; this paper discusses ten of them, describing how we propose to solve them. Topics include: Logical predicates of variable arity; the form to be taken by inference rules; deictic reference to other parts of the document; property inheritance and how to override inherited values; distinguishing between unions of properties and overriding of properties; treatment of milestones; definition of the universe of discourse; definite descriptions; and the recording of uncertainty and responsibility.

9.1 Background

In many ways, the use of markup in electronic documents to convey meaning seems, and is, unproblematic. Like many unproblematic topics, it may benefit from closer examination. An attempt to describe more formally and explicitly the use of markup to convey information may or may not lead to interesting or surprising results. In 1973, Wilhelm Ott wrote [9, p. V] that:

> Ihr [d.i. der EDV] Einsatz ist überall dort möglich, wo Daten irgendwelcher Art – also auch Texte – nach eindeutig formulierbaren und vollständig formalisierbaren Regeln verarbeitet werden müssen.[1]

C.M. Sperberg-McQueen
Black Mesa Technologies LLC, 259 State Road 399, Española,
New Mexico 87532-3170, USA
e-mail: cmsmcq@blackmesatech.com

Claus Huitfeldt
University of Bergen, Department of Philosophy, P.O. Box 7805, N-5020 Bergen, Norway
e-mail: Claus.Huitfeldt@fof.uib.no

[1] "The application of data processing is possible anywhere data of any kind – including texts – must be processed according to unambiguous, fully formalizable rules."

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 157–174.
springerlink.com

The goal of the work described here[2] is to formulate unambiguous formal rules for assigning meaning to the markup in documents. Our working definition of *meaning* is exhibited in this quotation from a work on the semantics of computer programs [11, p. 4]:

> . . . we shall accept that *the meaning of A is the set of sentences S true because of A*. The set S may also be called the set of consequences of A. Calling sentences of S consequences of A underscores the fact that there is an underlying logic which allows one to deduce that a sentence is a consequence of A.

When A is taken as denoting some markup construct in a document marked up using XML or some similar markup language, the description just given applies directly to the situation we wish to elucidate. (Without loss of generality, the remainder of this document will speak only of XML documents.) Following [10], S may also be referred to as the set of inferences licensed by A.

9.1.1 Derivation of Inferences

The rules for finding the set S for a given document or markup construct are not constant for all XML documents; they vary with the markup vocabulary in use. It is natural, then, to seek a method of documenting the meaning of a vocabulary more formally, in what may be called a ‘formal tag-set description’ (FTSD), in such a way that a process reading a document instance marked up using a vocabulary can consult an FTSD for that vocabulary and use it to generate the set S of inferences licensed by the markup in the document. Attention is limited to the meaning of the markup, rather than extending to the meaning of the document itself taken as a whole, because in the cases of interest here markup forms an utterance in a formal, artificial language and thus lends itself to systematic treatment by machines. The document content, in contrast, is usually in a natural language and in the current state of linguistics, natural-language processing, and artificial intelligence, its meaning is not readily identifiable by automatic means.

If S is the set of all inferences licensed by the markup in a document, then by its definition, S is closed under inference: any sentence inferable from S is already a member of S. In most systems of symbolic logic, the set of inferences licensed by any proposition p is infinite: from p one may infer $\neg\neg p$, $p \wedge p$, $p \leftrightarrow p$, $p \vee p$, $\neg\neg\neg\neg p$, etc. It should not be expected, therefore, that the sentences in S should be exhaustively enumerated; instead the goal is to enumerate some finite subset of S, which we may call S', whose closure under inference is S itself. That is, S' should be a set of sentences from which all of the sentences in S, and no others, may be inferred. In some cases only a subset of the inferences licensed by the markup is of interest and it is unnecessary to enumerate even S' exhaustively.

[2] This paper summarizes some work done in the context of the ‘Bechamel’ project. It owes a great deal to our collaborators in that project, Allen Renear and David Dubin of the University of Illinois in Urbana/Champaign, who are however not responsible for any of its shortcomings.

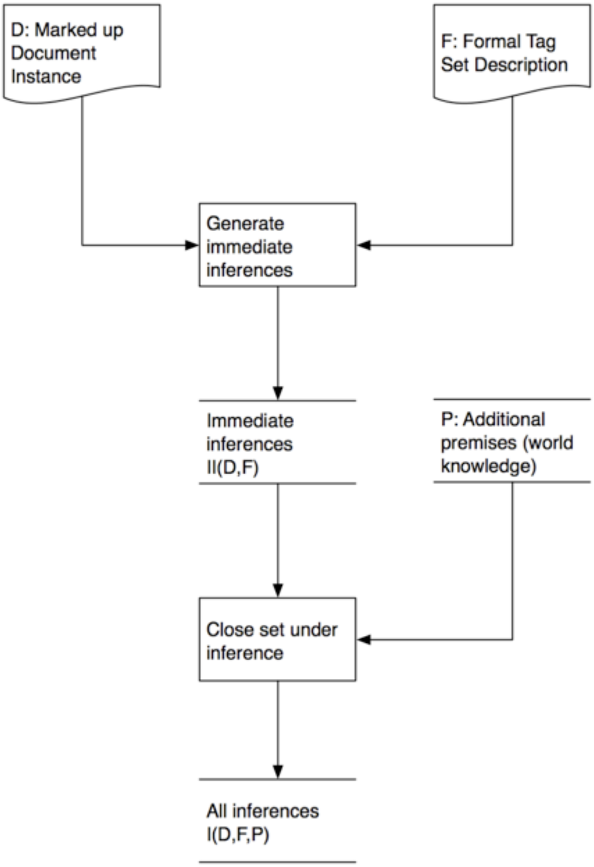

D: Marked up Document Instance
F: Formal Tag Set Description
Generate immediate inferences
Immediate inferences II(D,F)
P: Additional premises (world knowledge)
Close set under inference
All inferences I(D,F,P)

Fig. 9.1 Data flow in inferencing. As described in the text, the figure is a flowchart showing a flow of data through various processes. The flow begins at two documents labeled *D*: *marked up Document Instance* and *F*: *Formal Tag Set Description*. A process labeled *Generate immediate inferences* takes these as inputs and produces a data stream labeled *Immediate Inferences* $II(D,F)$. This in turn flows into a second process labeled *Close set under inference*. This second process also takes a second input, from a data stream labeled *P*: *Additional premises (world knowledge)*. The product of the second process is labeled *All inferences* $I(D,F,P)$.

The flowchart in Figure 9.1 illustrates the structure of the system:

The immediate inferences $II(D,F)$ to be drawn from the markup in a document D depend partly on D and partly on the formal tag-set description F which describes the markup vocabulary used in D. When combined with additional premises P (representing, for example, facts about a particular domain, or more generally ‘world knowledge’), and closed under inference, the immediate inferences give rise to the complete set $I(D,F,P)$ of inferences derivable from document D given the formal tag-set description F and the additional premises P.

In this flowchart, the set S is represented by the data stream labeled "*All inferences I(D,F,P)*"; the set S' is represented by the data stream labeled "*Immediate inferences II(D,F)*".

9.1.2 An Example

A short example may help to illustrate the goal. The following example shows part of a transcription using TEI markup [3] of a letter written in 1775 by the South Carolina political leader Henry Laurens to the royal governor of South Carolina, Lord William Campbell.

```
<p><del>It was be</del> <del>For</del> When we
applied to Your Excellency for leave to adjourn
it was because we foresaw that we
<del>were</del> <add>should continue</add>
wasting our own time ... </p>
```

The documentation of the TEI *Guidelines* enables the reader to make a number of inferences from the markup in this document. Among them:

- There is a paragraph in the document whose final text begins with the words "When we applied to Your Excellency for leave to adjourn".
- At the beginning of this paragraph, the words and letters "It was be" have been deleted.
- At the beginning of this paragraph, the word "For" has also been deleted (separately).
- Later in the paragraph, the word "were" has been deleted; at the same location, the words "should continue" have been inserted.

The inferences licensed by the markup in an XML document may be expressed as English sentences[3] or in some formal notation such as first-order predicate calculus (in any of its myriad forms) or Prolog.

If expressions of the form of a lower-case Latin letter and a subscript number (e.g. d_{10305}, p_1, s_1, s_2, etc.) denote individuals in the universe of discourse, and the predicates *is_document*, *is_paragraph*, and *contains* are defined in the obvious way, then S' might contain, among other things, the sentences:

- $\mathit{is\_document}(d_{10305})$
- $\mathit{is\_paragraph}(p_1)$
- $\mathit{contains}(d_{10305}, p_1)$

One of the tasks of the formal tag-set description is to provide the information necessary to generate appropriate sentences in the style just shown, for the markup actually encountered in the document instance.

[3] This is the approach taken by intertextual semantics, see [8, p. V].

XML provides no primitive semantics and imposes no restriction on the form of inferences licensed by markup; it may be impossible for any single notation for FTSDs to handle all imaginable XML-based markup languages. Many markup languages of fairly conventional design (what are sometimes called 'colloquial XML'), however, use similar idioms. One aim of our project is to formulate a notation for FTSDs which is relatively convenient for colloquial XML vocabularies.

In the discussion that follows, the inferences discussed are for the most part expressed in a conventional notation for first-order predicate calculus; in some cases, however, natural-language prose has seemed more convenient.

9.2 The Ten Problems

Any attempt to build a system of the kind described above will encounter a number of problems, some technical and others more philosophical in nature. The remainder of this paper describes ten of these problems, and proposes tentative solutions for some of them.

9.2.1 The Arity of Statements

The 'straw-man proposal' of [10] postulates that each element type in XML corresponds to a unary predicate which takes the element instance as an argument. It similarly assumes that every attribute can be described by a dyadic predicate whose arguments are the attribute instance and its host element instance. In this style, using "." to denote the element instance, skeleton sentences for the element types used in the example given above might be written:

- **p:** *is_paragraph*(.)
- **del:** *is_deletion*(.)
- **add:** *is_insertion*(.)

Because each predicate takes just one argument, which is always the element instance being glossed, the translation rules can be successfully represented just as a set of name pairs. Even name pairs are unnecessary if the names of element types are used in the formal system as names of predicates.

In skeleton sentences formulated in English, the word this can be used to denote the element instance:

- **p:** *this* is a paragraph.
- **del:** *this* is a deletion (*or*: is deleted material).
- **add:** *this* is an insertion (*or*: is material added later).

As described in [10], however, colloquial XML vocabularies often include element types which seem to invite paraphrase by predicates of more than one argument. For example:

- **TEI.2:** There exists a text;[4] *this* (the `TEI.2` element) is an electronic representation or encoding of *that* text.
- **bibl:** There exists a bibliographic item (i.e. a book, an article, anything that can be described by an item in a bibliography); *this* (the `bibl` element) contains (*or*: is) a bibliographic description of *that item*.
- **head:** *This* is the title of the *immediately enclosing* `div` | `div0` | `div1` | `div2` *. . . element*.

The solution is straightforward: skeleton sentences must be able to use predicates which take two or more arguments. Adopting the same convention as before for references to the current instance of the given element type:

- **TEI.2:** $(\exists x)(\mathit{text}(x) \wedge \mathit{encoding}(.) \wedge \mathit{encodes}(.,x))$
- **bibl:** $(\exists x)(\mathit{bibitem}(x) \wedge \mathit{bibdesc}(.) \wedge \mathit{bibitem\_desc}(x,.))$
- **head:** $(\exists x)(\mathit{section}(x) \wedge \mathit{section\text{-}title}(.) \wedge \mathit{title\_section}(.,x))$

The last formulation is not entirely satisfactory: it says that the head element is the title of some section of the text, but it does not capture the fact that it is the title of the same section as is represented or encoded by the enclosing div element.

9.2.2 The Form of Inference Rules

As the TEI *head* element illustrates, polyadic predicates often need to refer to more than one element or attribute in the document. Typically, these stand in some predictable relation to each other; in the case of *head*, one is the *head* element itself and the other is its parent element.

Given a universe of discourse that includes the elements and attributes of the document instance, and appropriate predicates for describing the relations (e.g. that between parent and child or that between preceding and following sibling), it is possible to describe the appropriate inference in a way familiar from other formal systems involving inference. Using a conventional notation for inference rules, one may write[5]

$\mathit{element}(x)$
$\mathit{element}(y)$
$\mathit{name}(z)$
$\mathit{element\_type}(x, \text{“}\mathit{head}\text{“})$

[4] Or, “There exists a document,” or “. . . a text witness. . . .” Different interpreters may take different views of just what it is that a `TEI.2` element allows the reader to infer. See further section 9.2.8.

[5] For simplicity this formulation assumes not that y encodes a section of a text but that y is a section of the text. See also section 9.2.8.

$$\mathit{element\_type}(y,z)$$
$$z \in \{“\mathit{div}“, “\mathit{div}0“, “\mathit{div}1“, “\mathit{div}2“, \ldots\}$$
$$\mathit{parent\_child}(y,x)$$

$$\mathit{section\text{-}title}(x)$$
$$\mathit{section}(y)$$
$$\mathit{title\_section}(x,y)$$

In English: for any elements x and y, where the element-type name of x is “head” and the element-type name z of y is one of “div”, “div0”, “div1”, etc., if y is the parent of x, then one may infer that x is a section-title, that y is a section, and that x is the title of y.

The inference rule can also be written as a universally quantified conditional (as is sometimes preferred).

$$(\forall x,y,z)(\mathit{element}(x) \wedge \mathit{element}(y) \wedge \mathit{name}(z)$$
$$\wedge \mathit{element\_type}(x, “\mathit{head}“)$$
$$\wedge \mathit{element\_type}(y,z) \wedge z \in “\mathit{div}“, “\mathit{div}0“, “\mathit{div}1“, “\mathit{div}2“, \ldots$$
$$\wedge \mathit{parent\_child}(y,x)$$
$$\rightarrow \mathit{section\text{-}title}(x) \wedge \mathit{section}(y) \wedge \mathit{title\_section}(x,y))$$

The set of inferences S will contain sentences derived from either of these rules by replacing the variables with references to individuals in the universe of discourse.[6]

It is sometimes convenient, however, to omit much of the notational machinery of the forms just given and (as suggested in [10]) to document the inferences licensed by particular markup constructs in the form of ‘skeleton sentences’ (sentence schemata) with blanks to be filled in from appropriate information in the context.

The rule for all *head* elements, for example, might be:

$$\mathit{section\text{-}title}(\underline{\quad\;}) \wedge \mathit{section}(\underline{\quad\;}) \wedge \mathit{title\_section}(\underline{\quad\;}, \underline{\quad\;})$$

with the specification that the first and third blanks are to be filled in with references to the *head* element itself, and the second and fourth with references to the parent element of the *head*, if that parent is a *div*, or a *div*0, or a *div*1, or a *div*2, etc.

How best to record information about how to fill in the blanks is the topic of the next section.

9.2.3 Deixis

When a skeleton sentence for a colloquial XML vocabulary refers to more than one element or attribute, the rule will typically be associated with one or the other of the elements or attributes referred to; the information in the FTSD must specify how to

[6] Note that under normal circumstances the consequence “section(___)” is likely to be redundant, since the rule for the parent element will independently license the inference that it is (or encodes) a section. It is retained here for clarity, since the redundancy does no harm: whether it is generated by the rule for *div* or the rule for *head*, or both, it will be in S.

find the other. The items referred to will typically stand in some prescribed relation to each other: one is the parent, or an ancestor, or a sibling, of the other, or one is the element identified by a particular IDREF attribute, and so forth. That is, it will normally be possible to navigate from one to the other by some well understood path, and it is the navigation path which must be recorded. The path expression will denote different results, depending on the starting point, so it will be, in the linguistic sense, a *deictic* expression.

As illustrated in the preceding section, the necessary relations can be defined by means of standard logical predicates. It proves more convenient, however, to formulate the deictic expressions using a notation invented for expressing such relations concisely and conveniently, e.g. caterpillar expressions [2], CSS selectors [4], or XPath expressions [5, 1].

In what follows, XPath expressions will be used for deixis; instead of blanks, skeleton sentences will contain XPath expressions; to distinguish them from other parts of the expression, they will be enclosed (following a notational convention in XSLT and XQuery) in braces.

In this modified notation, the skeleton sentence for *head* given in the preceding section takes the following form:

$$\begin{aligned}&\mathit{section\text{-}title}(\{.\})\\&\wedge \mathit{section}(\{\mathit{parent} :: \mathit{div} \mid \mathit{parent} :: \mathit{div}0 \mid \mathit{parent} :: \mathit{div}1 \mid \mathit{parent} :: \mathit{div}2\})\\&\wedge \mathit{title\_section}(\{.\},\{\mathit{parent} :: \mathit{div} \mid \mathit{parent} :: \mathit{div}0 \mid \mathit{parent} :: \mathit{div}1 \mid \mathit{parent} :: \mathit{div}2\})\end{aligned}$$

The title element as a child of a *bibl* element provides another simple example. In English, one of the required inferences is:

> *This* is a title of *the bibliographic item* described by *the nearest containing bibl element*.

A rough paraphrase of this sentence in logical notation is:[7]

$$\begin{aligned}&\mathit{title}(\{.\})\\&\wedge(\exists x)(\mathit{bibitem}(x)\\&\wedge \mathit{bibitem\_desc}(x,\{\mathit{parent} :: \mathit{bibl}\})\\&\wedge \mathit{bibitem\_title}(x,\{.\}))\end{aligned}$$

9.2.4 Inheritance

Colloquial XML vocabularies often use a form of inheritance to propagate properties from parent to child. The *xml:lang* attribute defined by the XML specification and used by [3] (and other vocabularies) provides an example. Given a TEI document containing a *body* element whose start-tag is `<body xml:lang="en">`, and which is to be referred to, in the sentences of S, by the arbitrary identifier e_{3141}, the reader is (oversimplifying slightly) licensed to infer that

[7] The paraphrase is rough because it fails to capture the claim implicit in the phrase "the bibliographic item described by the parent *bibl* element", namely that there is exactly one such bibliographic item. See further Section 9.2.9.

$\textit{is-in-English}(e_{3141})$

For simplicity, this inference assumes that the property of being in a given language is to be postulated of the document and its constituent parts, and not of some other object distinct from the document; see further Section 9.2.8.

Alternatively one might write:

$\textit{utterance\_language}(e_{3141}, \text{"}\textit{en}\text{"})$

That is, the element e_{3141} is in the language identified by the ISO 639 language code "*en*", i.e. English.

But it is legitimate to ask about the language not only of *body* elements, but also of paragraphs and sections. And the documentation for the *xml:lang* attribute makes clear that from the information given on the *body* element e_{3141} the reader is licensed (in the absence of contrary indications) to infer not only that e_{3141} itself is in English, but that each part of it is in English. That is, for each descendant d of e_{3141} the markup licenses (at a first approximation) the inference that

$\textit{utterance\_language}(d, \text{"}\textit{en}\text{"})$

or that

$\textit{is-in-English}(d)$

If the meaning of the *xml:lang* attribute were really as simple as just described, then the predicate *is-in-English* would fall into the class of *dissective* predicates identified by [6]: "A one-place predicate is said to be dissective if it is satisfied by every part of every individual that satisfies it" (p. 38). In practice, though, a document described as being in English can contain material in other languages without violating the documented meaning of *xml:lang*; for further discussion see Section 9.2.5 below.

Note that not all properties are inherited in the same way: the property of being in a given language applies equally to the document as a whole and to (most of) its parts, but the property of being a document, or a paragraph, or a poem, or a line of verse, does not similarly hold for the component parts of the document, paragraph, poem, or line.

9.2.5 Overriding

The account of inheritance given above in Section 9.2.4 is too simple: in many cases, inheritance specifies a default value for a property, but a defeasible one, which can be overridden by a locally specified value. The *xml:lang* attribute is a well known example of this design pattern: the language property of an element e is given by the *xml:lang* attribute specified on e, if there is one, and otherwise by the *xml:lang* attribute on the nearest ancestor which has an explicit value for the attribute.

An operational system to enumerate the immediate inferences licensed by the markup in a document can generate the appropriate inferences either top-down or bottom-up. The difference is of no particular theoretical interest, but the choice may have practical consequences.

One way to propagate language information top-down is to use two inference rules. The first licenses an instance of the *utterance_language* predicate introduced in Section 2.4:

$$\begin{array}{l}
\mathit{element}(x)\\
\mathit{attribute}(y)\\
\mathit{string}(z)\\
\mathit{attribute\_localname}(y, \text{‘}\mathit{lang}\text{‘})\\
\mathit{attribute\_ns}(y, \text{‘}\mathit{http}: //\mathit{www.w}3.\mathit{org}/\mathit{XML}/1998/\mathit{namespace}\text{‘})\\
\mathit{attribute\_value}(y, z)\\
\hline
\mathit{utterance\_language}(x, z)
\end{array}$$

The second inference rule propagates the information downward:

$$\begin{array}{l}
\mathit{element}(x)\\
\mathit{element}(y)\\
\mathit{string}(z)\\
\mathit{utterance\_language}(x, z)\\
\mathit{parent\_child}(x, y)\\
\neg(\exists w)(\mathit{attribute}(w)\\
\wedge\, \mathit{attribute\_parent}(w, y)\\
\wedge\, \mathit{attribute\_localname}(w, \text{‘}\mathit{lang}\text{‘})\\
\wedge\, \mathit{attribute\_ns}(w, \text{‘}\mathit{http}: //\mathit{www.w}3.\mathit{org}/\mathit{XML}/1998/\mathit{namespace}\text{‘}))\\
\hline
\mathit{utterance\_language}(w, z)
\end{array}$$

For every element *e*, the first rule ensures that if *e* has an explicit value for *xml:lang*, then *e* uses the specified value for its language property. The second rule ensures that if *e* has no local value for *xml:lang*, then it will inherit the language value of its parent.

The language property can also be calculated bottom-up, for example by instantiating the following skeleton sentence for each element in the document:

utterance_language({.},
{`ancestor-or-self::* [@xml:lang] [1] / @xml:lang`})

If an element has neither an *xml:lang* value of its own nor any ancestor with one, the expression just given will assign it the empty string as its language identifier. This is not an unusual convention, but if it is desired instead to use the ISO 639-2 code “und” for undetermined languages, then the XPath 2.0 may be used to make the expression conditional:

```
utterance_language({.},
{if (ancestor-or-self::* [@xml:lang])
then ancestor-or-self::* [@xml:lang] [1] / @xml:lang
else "und" })
```

Both techniques just outlined follow conventional practice in assuming that XML elements have a language property; if it is desired, however, to exclude XML elements from the first argument of *utterance_language* and allow only character strings (here, the character data contained by the XML elements), then it will be preferable to use a rule like the following in the FTSD, applied to every text node in the document:

```
utterance_language({string(.)},
{ancestor-or-self::* [@xml:lang] [1] / @xml:lang})
```

9.2.6 Conflict and Union

In the case of *xml:lang*, the conflict between a specification of `xml:lang="en"` on one ancestor and `xml:lang="de"` on another is easily detected and resolved.

In other cases, detecting conflict may be harder. Do the XHTML elements *b* (bold) and *i* (italic), or analogous constructs in other markup languages, override each other, or supplement each other?

If an *i* element inside a *b* element is intended to override the boldface and to signal that its contents are italic (not bold italic), it will be helpful to write the skeleton sentences for the two element types in terms of the same property. If in the universe of discourse both elements and text nodes may have a styling property, whose values are (for example) `"bold"`, `"italic"`, or `"normal"`, then the following skeleton sentence, applicable to every element and every text node in the XML document, will correctly convey the intention.

```
node_styling({.},
{if (ancestor-or-self::*[self::b or self::i][1][self::b])
    then "bold"
else if (ancestor-or-self::*[self::b or self::i][1][self::i])
    then "italic" else "normal"})
```

Handling the case of nested XHTML *i* and *em* elements, or analogous constructs in other languages, will require similar rules.

If on the other hand the *b* and *i* elements are taken to be orthogonal, so that any content appearing in both a *b* and an *i* is bold italic, then it is straightforward to write the skeleton sentences in terms of two distinct properties.

```
node_font-weight({.},
{if (ancestor-or-self::b) then "bold" else "normal"})
```

```
node_font-style({.},
{if (ancestor-or-self::i) then "italic" else "normal"})
```

9.2.7 Milestones

As illustrated above, colloquial XML vocabularies typically define the meaning of markup constructs in terms of the tree structure characteristic of XML. The properties associated with an element or a text node depend on the element-types and attributes of the element itself and of its ancestors.

The 'milestone' idiom used in some vocabularies to mark non-hierarchical phenomena, by contrast, depends less on the tree structure of the XML document than on its form as a sequence of characters interspersed with markup: each milestone element (such as the *pb* element in [3]) specifies a property value not for descendants of the milestone element (typically there are none) but for the text nodes and elements which follow the milestone in the data stream.

The meaning of a milestone, however, is easy to specify, as long as the language used for deictic expressions allows it to be expressed.[8] In XPath, the *preceding* axis provides the necessary information.

The TEI *pb* element specifies the page number of the following material in its *n* attribute, and in its *ed* attribute it identifies the edition(s) whose pagination is thus recorded. The following skeleton sentence for text nodes will assign to each text node in the document the appropriate page number for edition *e*; the sentence should be instantiated once for each edition whose pagination is recorded. The relation *textnode_ed_pagenum* is a set of triples whose members are (1) text nodes, (2) edition identifiers (strings), and (3) page numbers.

$$\mathit{textnode\_ed\_pagenum}(\{.\},\ e,\ \{\texttt{preceding::pb[@ed="e"][1]/@n}\})$$

Elements will not, in the general case, have a single page number. If it is desired to assert, for each element *x*, a sentence of the form "$\mathit{element\_ed\_pagenum}(x,e,n)$" for each edition *e* and page number *n* from which material appears in some text node of *x*, the following conditional will license the necessary inferences.

$$(\forall x)(\forall t)(\forall e)(\forall n)(\mathit{element}(x) \wedge \mathit{textnode}(t) \wedge \mathit{ancestor\_descendant}(x,t) \wedge \\ \mathit{textnode\_ed\_pagenum}(t,e,n) \rightarrow \mathit{element\_ed\_pagenum}(x,e,n))$$

9.2.8 The Universe of Discourse

Any satisfactory account of the interpretation of markup in documents is obliged to specify a universe of discourse. What individuals, what kinds of individuals, can be referred to in the set *S* of inferences licensed by the markup? What properties do they have? What relations hold among them?

For several reasons, these can be contentious issues.

XML provides no assistance in this matter: in the interests of flexibility, XML avoids constraining the universe of discourse for XML vocabularies. Designers of XML vocabularies can thus assume any universe of discourse they desire.

[8] This is not a given. CSS selectors [4], for example, do not allow the rules given here to be expressed.

For many colloquial XML vocabularies, however, the universe of discourse is not specified formally or fully: such a detailed specification might be essential for a proper formal tag-set description, but it is not essential in practice for conventional XML documentation, or for the successful use of colloquial vocabularies. Many users of an XML vocabulary will have no particular interest in any formalization of the domain beyond the use of XML itself to define element and attribute types and to mark instances of those types in documents. A painstaking enumeration of the various classes of individual in the expected domain of discourse can even prove confusing, if the required distinctions are sufficiently subtle to make explication difficult.

Moreover, vocabularies developed for use in heterogeneous communities may practice a studied agnosticism with regard to the nature of the entities about which the markup is expected to provide information. The TEI *Guidelines*, for example, specify that the tag *p* is used to mark paragraphs, but carefully avoid any attempt to specify just what a paragraph is or to list its distinguishing characteristics. Is a paragraph a sequence of characters? A portion of one or more pages in a manuscript? An abstract object with various suitable properties? The *Guidelines* would be more complete if a more explicit account of the nature of paragraph-hood had been provided, but the intended user community does not have consensus on the issue, and any more precise formulation might be expected to alienate at least some potential users. In effect, the *Guidelines* take *paragraph* to be a given class of individuals, and do not attempt to specify explicitly what relations might hold between that class and other classes of individual.

In practice, attempts to agree even among a small group of interested parties on the universe of discourse to be assumed even for small toy examples can prove difficult. Deep convictions about the nature of the world, the reality or otherwise of sets and other abstract objects, and correct modeling practice may all be involved.

Some careful observers, for example, will wish for a clear, crisp distinction between the XML document and the domain-specific objects it represents. A *customer* element in an electronic purchase order is not a customer but a representation of a customer. The sentences in the set of inferences *S* will be characterized by the fact that their universe of discourse will be that of the application domain: they will talk about customers and products and shipping dates, not about XML elements and attributes. XML elements and attributes, or other artefacts of the representation, will be present, if at all, only in the preconditions of inference rules.

By analogy, one may argue that a *p* element in a TEI document is not a paragraph but a representation of a paragraph. From this point of view, many of the examples given in previous sections of this paper need revision. Holders of this view might for example argue that it is not XML elements or text nodes, but the natural-language utterances they encode, which have the property of being in a language. The rule given in Section 9.2.5 above for the language property of elements might be rewritten, with the aid of an *encodes* relation holding between XML elements and natural-language utterances, as follows.

$(\exists u)(\mathit{utterance}(u) \wedge \mathit{encodes}(\{.\},u) \wedge \mathit{utterance\_language}(u,$
$\{$`if (ancestor-or-self::* [@xml:lang])`
`then ancestor-or-self::* [@xml:lang] [1] / @xml:lang`
`else "und"`$\}))$

Other careful practitioners may object to this reformulation on philosophical grounds. The principle of Occam's Razor leads them to reject the existence of utterances as distinct from the sound patterns or physical writing media by which utterances are conveyed. They will accordingly prefer the earlier formulation of this rule. The same practitioners may object, on similar grounds, to the postulation of *text* as a class of abstract objects, and prefer to restrict the sentences of S to a universe of discourse perhaps containing (physical) documents, but not (abstract) texts.

From a TEI-encoded document D identified as a transcription of some manuscript M, for example, some formulations of TEI semantics might generate the following list of licensed inferences.

- There exists a manuscript M.
- The catalog number of M is ...
- There exists a text T such that M is a manuscript representation of T.
- The language of T is German.
- D is an electronic representation of T, transcribed from M.

An alternative formulation would dispense with the abstract object T:

- There exists a manuscript M.
- The catalog number of M is ...
- D is a transcription of M.
- The language of M is German.
- The language of D is German.

9.2.9 Definite Descriptions and Multiple References to the Same Individual

In some cases, several markup constructs in a document license inferences about the same individual in the universe of discourse.

```
<bibl id="soa3" n="Goodman␣1977">
<author>Goodman, Nelson</author>.
<date>1977</date>.
<title level="m">The structure of appearance</title>.
<edition>Third edition</edition>.
<pubPlace>Boston</pubPlace>:
<publisher>Reidel</publisher>.
</bibl>
```

The following TEI *bibl* element may illustrate the point.

The *bibl* element itself licenses inferences that can be expressed in English as:

- There exists a bibliographic item *b*.
- This *bibl* element contains (or is) a bibliographic description of *b*.

The *title* element appearing within the *bibl* licenses not only the inference

- The string "The structure of appearance" is a title.

but also

- The string "The structure of appearance" is the title of the bibliographic item described by the parent *bibl* element (i.e. *b*).

The mechanism for enumerating the inferences in *S* must make clear that the bibliographic item mentioned in the inferences from the *bibl* element and the bibliographic item mentioned in the inferences from the *title* element are the same bibliographic item.

To take another example, the Open Archives Initiative protocol for metadata harvesting [7] defines a markup language for queries sent to OAI servers and for responses to those queries. The response header provides information about the request which elicited the response (for example, a request for metadata in a particular format about a particular resource), and unless an error has occurred the response body provides the information requested (for example, the current metadata record for the resource identified in the query). The fact that the resource described by the metadata record is the same individual as the resource identified in the query is central to the meaning of the markup.

The usual method of translating definite descriptions in symbolic logic is due to Russell. Using this method, a reference to "the x such that Px" is translated into the assertion that there is an x such that Px, and that for all y such that Py, $y = x$. The inference given above as

> The string 'The structure of appearance' is the title of the bibliographic item described by the parent *bibl* element (i.e. *b*).

is accordingly translated into

$$
\begin{aligned}
&(\exists b)(\textit{bibliographic-item}(b)\\
&\wedge \textit{describes}(\{\texttt{parent::bibl}\}, b)\\
&\wedge (\forall y)(\textit{bibliographic-item}(y) \wedge \textit{describes}(\{\texttt{parent::bibl}\}, y) \rightarrow y = b)\\
&\wedge \textit{bibitem\_title}(b, \{\texttt{string(.)}\}))
\end{aligned}
$$

In English: There exists some *b*, and *b* is a bibliographic item, and *b* is described by the *bibl* element which is the parent of this (*title*) element, and for any thing *y*, if *y* is a bibliographic item and *y* is described by the *bibl* element which is the parent of this (*title*) element, then *y* is identical to *b* – and the string value of this (*title*) element is the title of *b*.

Some logical formalisms provide shorthand notations for asserting the uniqueness of entities; such shorthands may be expected to be helpful in practical work but they will not be discussed further here.

9.2.10 Certainty and Responsibility

A special challenge is posted by markup which explicitly marks some inferences normally licensed by markup as doubtful, uncertain, or controversial. Extensive support for markup of uncertainty, for specifying alternative interpretations, and for attributing particular interpretations to particular authorities in markup languages like the TEI *Guidelines* or the MECS-WIT language used in preparing the Bergen Electronic Edition of Wittgenstein's *Nachlaß* [12].

The markup may indicate, for example, that a particular proper noun in the document (e.g. "Essex", marked by XML element e_{3141}) is probably a reference to a person (identified by a database key as p_{2234}), but possibly instead a reference to a geographic entity (p_{7621}). If there were no doubt about the matter, the set of inferences S might include the sentence

$$\textit{person-name\_id}(e_{3141}, p_{2234})$$

When there is some doubt, however, the process for generating set S must find some other way of formulating the information.

A satisfactory solution to this problem may require a choice of methods for handling probabilistic or defeasible reasoning; such a choice goes beyond the scope of this paper, which will limit itself to sketching a few possible methods of recording uncertainty.

When the inferences are to be formulated in natural language, the solution may be as simple as writing

> The word "Essex" here (in element e_{3141}) is probably a reference to the person whose database ID is p_{2234}. (Alternatively it may be a reference to geographic location p_{7621}.)

When the inferences in S are to be formulated in a formal notation, each predicate can be augmented by an additional argument indicating a level of probability. For illustration, the examples here use a number between 0 (falsehood) and 1 (truth, certainty). The following sentences assign probabilities of 75% and 20% to the two analyses indicated, leaving a 5% chance that the word is something else entirely.

$$\textit{person-name\_id\_p}(e_{3141}, p_{2234}, 0.75)$$
$$\textit{place-name\_id\_p}(e_{3141}, p_{7621}, 0.20)$$

If only a few inferences are unproblematic, it may be convenient to define predicates which omit the probability, and to define them as implying the more complex predicate with 100% probability.

$$(\forall x)(\forall y)(\textit{person-name\_id}(x,y) \rightarrow \textit{person-name\_id\_p}(x,y,1.00))$$
$$(\forall x)(\forall y)(\textit{place-name\_id}(x,y) \rightarrow \textit{place-name\_id\_p}(x,y,1.00))$$

A different method is to reify the assertions in order to talk about them without asserting them. Using a notation for structured terms similar to that used in logic programming, the example could be written as

$$sentence\_probability(person\text{-}name\_id(e_{3141}, p_{2234}), 0.75)$$
$$sentence\_probability(place\text{-}name\_id(e_{3141}, p_{7621}), 0.20)$$

Alternatively, each sentence in S could be given an identifier.

In the absence of any mechanisms for working with probabilities or for defeasible inference, one simple but drastic method is to use disjunction, thus capturing the uncertainty, though not the relative likelihood of the two analyses.

$$person\text{-}name\_id(e_{3141}, p_{2234}) \vee place\text{-}name\_id(e_{3141}, p_{7621})$$

Other methods have been proposed in the extensive recent discussion of methods of reifying statements in the context of work on Topic Maps and on the Semantic Web; even a brief survey of the alternatives lies beyond the scope of this paper. As a practical matter, satisfactory results for the expression of uncertainty will only be achieved when the nature of the system intended to use the information is better understood.

References

[1] Berglund, A., Boag, S., Chamberlin, D., Fernández, M., Kay, M., Robie, J., Siméon, J.: XML path language (XPath) 2.0. W3C Recommendation, World Wide Web Consortium (2007), http://www.w3.org/TR/xpath20

[2] Brüggemann-Klein, A., Wood, D.: Caterpillars: a context specification technique. Markup Languages: Theory & Practice 2(1), 81–106 (2000)

[3] Burnard, L., Bauman, S.: TEI P5: Guidelines for electronic text encoding and interchange. Tech. rep. TEI Consortium (2007), http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html

[4] Çelik, T., Etemad, E., Glazman, D., Hickson, I., Linss, P., Williams, J.: Selectors level 3. W3C Working Draft, World Wide Web Consortium, Cambridge, Tokyo, Sophia-Antipolis (2009), http://www.w3.org/TR/css3-selectors/

[5] Clark, J., Derose, S.: XML path language (XPath): Version 1.0. W3C Recommendation, World Wide Web Consortium, Cambridge, Tokyo, Sophia-Antipolis (1999), http://www.w3.org/TR/xpath

[6] Goodman, N.: The structure of appearance, 3rd edn. Reidel, Boston (1977)

[7] Lagoze, C., Van de Sompel, H., Nelson, M., Warner, S.: The open archives initiative protocol for metadata harvesting: Version 2.0. Tech. rep., Open Archives Initiative, (2002), http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm

[8] Marcoux, Y., Sperberg-McQueen, C., Huitfeldt, C.: Formal and informal meaning from documents through skeleton sentences: Complementing formal tag-set descriptions with intertextual semantics and vice-versa. In: Proceedings of Balisage: The Markup Conference 2009, Montréal, Canada. Balisage Series on Markup Technologies, vol. 3 (2009), http://balisage.net/Proceedings/vol3/html/Sperberg-McQueen01/BalisageVol3-Sperberg-McQueen01.html

[9] Ott, W.: Metrische Analysen zu Vergil Aeneis Buch VI. Niemeyer, Tübingen (1973)

[10] Sperberg-McQueen, C., Huitfeldt, C., Renear, A.: Meaning and interpretation of markup. Markup Languages: Theory & Practice 2(3), 215–234 (2001), http://www.w3.org/People/cmsmcq/2000/mim.html

[11] Turski, W., Maibaum, T.: The Specification of Computer Programs. Addison-Wesley, Wokingham (1987)

[12] Wittgenstein, L.: Wittgenstein's Nachlaß: The Bergen electronic edition. Wittgenstein Archives at the University of Bergen. Oxford University Press, Oxford (2000)

Chapter 10
Markup Infrastructure for the Anaphoric Bank: Supporting Web Collaboration

Massimo Poesio, Nils Diewald, Maik Stührenberg, Jon Chamberlain,
Daniel Jettka, Daniela Goecke, and Udo Kruschwitz

Abstract. Modern NLP systems rely either on unsupervised methods, or on data created as part of governmental initiatives such as MUC, ACE, or GALE. The data created in these efforts tend to be annotated according to task-specific schemes. The Anaphoric Bank is an attempt to create large quantities of data annotated with anaphoric information according to a general purpose and linguistically motivated scheme. We do this by pooling smaller amounts of data annotated according to rich schemes that are by and large compatible, and by taking advantage of Web collaboration. In this chapter we discuss the markup infrastructure that underpins the two modalities of Web collaboration in the project: expert annotation and game-based annotation.

10.1 Introduction

Modern, statistical computational linguistics crucially relies on the availability of large amounts of annotated data, but such data requires substantial investment to produce. The problem is particularly serious for semantic annotation tasks such as anaphoric annotation or word-sense annotation. The lack of an established consensus on the linguistics of such phenomena and the potential open-endedness of the task (e.g., virtually every expression is anaphoric in some respect) mean that a

Massimo Poesio · Jon Chamberlain · Udo Kruschwitz
University of Essex, School of Computer Science and Electronic Engineering,
Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
e-mail: {poesio,jchamb,udo}@essex.ac.uk

Nils Diewald · Maik Stührenberg · Daniel Jettka · Daniela Goecke
Bielefeld University, Faculty of Linguistics and Literary Studies,
Universitätsstraße 25, D-33615 Bielefeld, Germany
e-mail: {nils.diewald,maik.stuehrenberg,daniel.jettka,
daniela.goecke}@uni-bielefeld.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 175–195.
springerlink.com

substantial effort is required to define the guidelines[1] and the judgments required of annotators are often subtle. In addition, the mediocre performance of existing systems makes semi-automatic annotation unfeasible. As a result, semantically annotated corpora have been created in two types of contexts: to support initiatives such as MUC or ACE; or to provide data for other projects, such as PhD dissertations. In the first example of annotation large amounts of data have been annotated, but using simplified and task-specific guidelines (for a critique of the MUC scheme, see [7]). In the second, linguistically rich annotation schemes have been used (see e.g., the schemes used for the GNOME annotation [25], Navarretta's dissertation work [23], or early work in the Sekimo project [28]) but the amount of data annotated is rather small.

The situation has been greatly improved with the OntoNotes project [13], that is creating a 1 million word corpus annotated for coreference, argument structure, and word-senses. OntoNotes is likely to become the main reference corpus for studying semantics and training semantic annotators in the near future. Data created by OntoNotes will have limitations: only a few languages (English, Arabic and Chinese) and genres (news), and only some of the information of interest to study anaphora resolution are covered. (E.g., no agreement information is annotated, or associative reference.)

However, the increasing convergence between markup schemes –e.g., the development of pivot representations, see below– and, in the case of anaphora, among annotation schemes, makes it possible to pursue a different type of solution by pooling together richly annotated but smaller resources. Such resources may usefully complement larger annotated corpora (e.g., to study aspects of anaphoric interpretation that are less well understood), or to train specialised classifiers (e.g., for gender detection). The *Anaphoric Bank*[2] can be seen as a club whose participants are groups willing to share, or to participate in the collaborative creation of, anaphorically annotated corpora. It focuses on linguistically rich annotation, on genres such as narratives or spoken language that, although linguistically key, are not considered of interest in practical applications and on languages other than English. The Anaphoric Bank attempts to achieve the creation of large anaphorically annotated corpora for different languages by:

1. pooling together corpora which have already been annotated according to compatible schemes;
2. providing a framework in which new annotated resources for anaphora can be created in such a way that facilitates their reuse, in particular utilising Web collaboration;
3. encouraging such annotation.

These goals have become achievable as the result of recent developments in XML-based annotation, in particular the development of 'pivot' representation formats, including the GRAF representation format developed within ISO TC37 SC4 [14],

[1] An example is the effort spent on this in the creation of OntoNotes, see [13].

[2] http://www.anaphoricbank.org/

PAULA (Potsdamer Austauschformat für Linguistische Annotation – Potsdam Interchange format for Linguistic Annotation) [10], SGF (Sekimo Generic Format) [31] and its successor XSTANDOFF [32]. In addition, standard tools for anaphoric annotation such as MMAX [22] or Serengeti [33, 9] lead to more uniformity among annotations. These developments make it possible to share resources by developing import and export converters from the markup format of these standard annotation tools to the pivot representation. In addition, the greater agreement on annotation schemes for anaphora (cf. [19]) makes it easier to standardise the task. In this chapter we will discuss this infrastructure supporting the Anaphoric Bank project, focusing in particular on the creation of new resources.

The structure of the chapter is as follows. In Section 10.2 we discuss how we expect new data to be added to the Anaphoric Bank. A novelty of the project is that two different ways of creating new data through Web collaboration are supported: through what we will call *Expert Annotation*, and by playing a game. We will explain how both modes of annotation have already been applied to build a substantial annotated corpus as part of the AnaWiki[3] research project. In Section 10.3 we discuss the uniform XML infrastructure supporting these two types of contributions, and in particular the SGF format and how a relational database representation of the format is used by both types of resource creation.

10.2 How Data Is Added to the Anaphoric Bank

We envisage two different ways of contributing to the Anaphoric Bank: by sharing data that has already been annotated, and by annotating new data on the Web. We can control data added in the second way. Augmenting the corpus using the first method requires criteria to be defined in order to avoid adding data annotated according to incompatible markup languages or annotation schemes. We will begin by discussing such criteria, before explaining how we expect to integrate data annotated using different markup languages. Finally, we will explain how we propose to use Web collaboration to augment the bank.

10.2.1 Filtering Criteria

Application-oriented definitions of the coreference or anaphoric task differ from linguistically oriented definitions along three main dimensions:

1. whether all potentially anaphoric nominals should be treated as markables, or only those that are mentions of a restricted set of entities, as done e.g., in MUC or ACE [12];
2. whether the entire nominal should be marked, or only the head noun without any postmodifiers (e.g., whether in *the man who shot Liberty Valance*, only *the man* should be treated as markable, or whether instead the relative clause should be included);

[3] http://www.anawiki.org/

3. how the ‘coreference’ or ‘anaphoric’ relation should be defined. In MUC and ACE, the (predication) relation between *Microsoft stock* and *$3* in *Microsoft stock was $3 today* is marked as coreference. The consequence is that potentially the coders should mark *$3* and *$2.75* as co-referring in *Microsoft stock was $3 today, up from $2.75 yesterday* [7].

Given that there is already plenty of data annotated according to the MUC/ACE guidelines, and that anyway this scheme is being abandoned in more recent anaphoric annotation efforts, it was decided to restrict the Anaphoric Bank to text annotated to ‘linguistic’ guidelines–i.e., annotations in which all nominals are treated as markables, the full boundaries of markables are identified, identity is restricted to pure identity and does not include predication (although a separate predication relation can be marked, as specified by the MATE and OntoNotes guidelines).

10.2.2 Data That Has Already Been Annotated

The development of powerful anaphoric annotation tools such as MMAX2[4], Palinka[5] [24], WordFreak[6] [21] and Serengeti has greatly facilitated the creation of anaphorically annotated resources both on a small and larger scale. However, the lack of uniformity among their markup languages (e.g., Palinka stores annotations inline; MMAX2, Serengeti and WordFreak all employ standoff markup, but in the case of MMAX2 it is token standoff, whereas in the case of Serengeti and WordFreak it is character standoff) means that even sharing data annotated according to uniform annotation schemes requires some form of conversion between formats.

Fortunately a solution has been proposed by using the already mentioned pivot representation formats. We envisage using one of these pivot formats (e.g., PAULA or SGF) as the uniform representation format for the data in the Anaphoric Bank, and developing import / export scripts to convert data annotated according to other formats into such representations. The Potsdam group already developed import scripts from Palinka and MMAX2 into PAULA.

10.2.3 Using the Expert Annotation Tool

So far we have looked at collecting data that has already been annotated. The Anaphoric Bank offers an alternative to this which allows access to shared corpora for those interested parties that have no data to provide: research groups or interested individuals (e.g., research students and other researchers) can annotate data through the Web as a way to be granted access to the collections maintained in the Anaphoric Bank.

[4] `http://mmax2.net/`

[5] `http://clg.wlv.ac.uk/projects/PALinkA/`

[6] `http://sourceforge.net/projects/wordfreak/`

10.2.3.1 Serengeti Annotator

A collaborative annotation project like the Anaphoric Bank requires a special architecture regarding the submission and management of the collected data. Compared to tools with separated data stores, browser-based annotation tools with a central data management offer some advantages in this aspect:

1. All corpus data is managed on a central server. New corpus data can be added at any time and is immediately available to all members of the annotation project. All annotation data can be accessed, compared and validated at any time.
2. The annotation scheme is centrally managed. It can be adjusted at any time with immediate effect to the annotators and the collected data.
3. The software is maintained centrally. New functions can be added at any time with immediate effect to the annotators.

Applying web-based tools for collaborative annotation projects minimises the effort required by the annotators for the annotation process.

Serengeti [33, 9] is a tool that adheres to these principles. It was developed in the Sekimo project for the annotation of semantic relations, targeted initially for Mozilla Firefox,[7] and currently present in version 0.8.15.[8] The AJAX-driven application (Asynchronous JavaScript and XML, cf. [11]) is written in Perl on the server side and uses a MySQL database (see 10.3.2).

Serengeti allows the fine-grained definition of markables (that is textual spans that can take part in semantic relations) as well as the annotation of anaphoric relations within a simple graphical user interface.

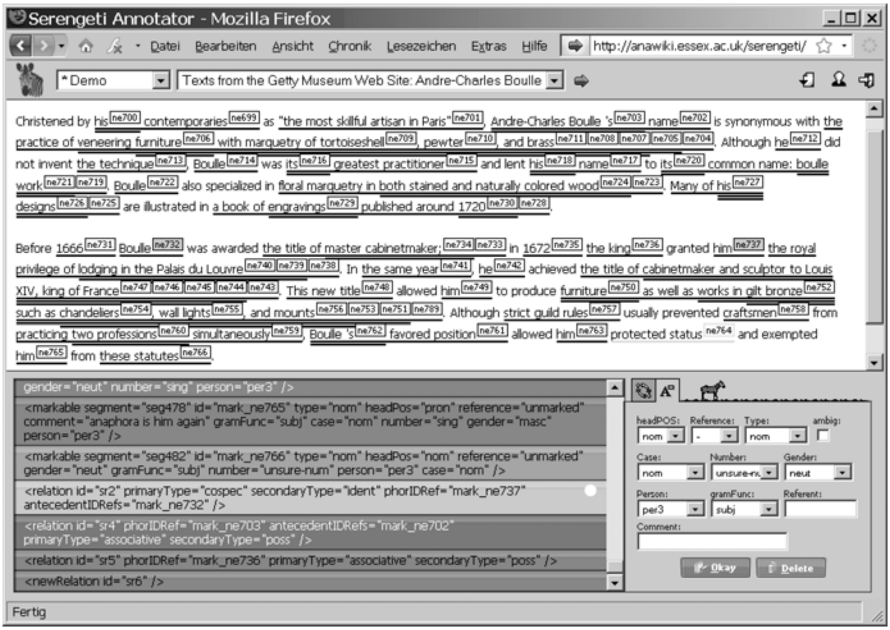
The GUI is subdivided into several parts (cf. Figure 10.1). Beneath the main area, where the document is rendered, all annotated relations and defined markables are listed as empty XML elements, reflecting the underlying annotation scheme. Next to the list are two tabbed forms for the type selection of markables and relations. Predefined markables in the text are underlined, followed by clickable boxes to choose them as anaphora or antecedents when creating a relation. By marking a span of the text, a new markable can be created that can take part in further relations.[9]

In a second step an administrating user is able to validate the work of the annotators (e.g., considering possible game data, see 10.2.4) to create gold standard annotations that can be part of the Anaphoric Bank.

[7] Firefox is freely available for several operating systems, published by the Mozilla Project Group under the terms of the Mozilla Public License (`http://www.mozilla.com/firefox/`).

[8] `http://anawiki.essex.ac.uk/serengeti/`

[9] For a detailed description of the annotation process, see [8].


Serengeti Annotator - Mozilla Firefox
Datei
Bearbeiten
Ansicht
Chronik
Lesezeichen
Extras
Hilfe
http://anawiki.essex.ac.uk/serengeti/
*Demo
Texts from the Getty Museum Web Site: Andre-Charles Boulle

Christened by his[ne700] contemporaries[ne699] as "the most skillful artisan in Paris"[ne701], Andre-Charles Boulle 's[ne703] name[ne702] is synonymous with the practice of veneering furniture[ne706] with marquetry of tortoiseshell[ne709], pewter[ne710], and brass[ne711][ne708][ne707][ne705][ne704]. Although he[ne712] did not invent the technique[ne713], Boulle[ne714] was its[ne716] greatest practitioner[ne715] and lent his[ne718] name[ne717] to its[ne720] common name: boulle work[ne721][ne719]. Boulle[ne722] also specialized in floral marquetry in both stained and naturally colored wood[ne724][ne723]. Many of his[ne727] designs[ne726][ne725] are illustrated in a book of engravings[ne729] published around 1720[ne730][ne728].

Before 1666[ne731] Boulle[ne732] was awarded the title of master cabinetmaker;[ne734][ne733] in 1672[ne735] the king[ne736] granted him[ne737] the royal privilege of lodging in the Palais du Louvre[ne740][ne739][ne738]. In the same year[ne741], he[ne742] achieved the title of cabinetmaker and sculptor to Louis XIV, king of France[ne747][ne746][ne745][ne744][ne743]. This new title[ne748] allowed him[ne749] to produce furniture[ne750] as well as works in gilt bronze[ne752] such as chandeliers[ne754], wall lights[ne755], and mounts[ne756][ne753][ne751][ne789]. Although strict guild rules[ne757] usually prevented craftsmen[ne758] from practicing two professions[ne760] simultaneously[ne759], Boulle 's[ne762] favored position[ne761] allowed him[ne763] protected status ne764 and exempted him[ne765] from these statutes[ne766].

```
gender="neut" number="sing" person="per3" />
<markable segment="seg478" id="mark_ne765" type="nom" headPos="pron" reference="unmarked"
comment="anaphora is him again" gramFunc="subj" case="nom" number="sing" gender="masc"
person="per3" />
<markable segment="seg482" id="mark_ne766" type="nom" headPos="nom" reference="unmarked"
gender="neut" gramFunc="subj" number="unsure-num" person="per3" case="nom" />
<relation id="sr2" primaryType="cospec" secondaryType="ident" phorIDRef="mark_ne737"
antecedentIDRefs="mark_ne732" />
<relation id="sr4" phorIDRef="mark_ne703" antecedentIDRefs="mark_ne702"
primaryType="associative" secondaryType="poss" />
<relation id="sr5" phorIDRef="mark_ne736" primaryType="associative" secondaryType="poss" />
<newRelation id="sr6" />
```

headPOS: nom
Reference: -
Type: nom
ambig: □
Case: nom
Number: unsure-nu
Gender: neut
Person: per3
gramFunc: subj
Referent:
Comment:

Fig. 10.1 Graphical user interface of Serengeti.

10.2.3.2 The Data

The input format for the expert annotation tool is the SGF (cf. 10.3.1) which has been developed in the Sekimo project to store multiple annotation layers. The input for Serengeti are three annotation layers defined by distributed XML Schema instances providing information on logical document structure, anaphoric relations and markables. The schema for the logical document structure contains definitions about paragraph and sentence elements. A development of the ARRAU scheme [27] is used to define relations and markables (based on the MATE/GNOME schemes [26]) and is imported as an annotation layer in SGF.

Relations are specified by attribute information on their primary relation type (i.e. cospecification or bridging [5]) and their secondary relation type which allows for a further distinction of subtypes of the primary type (cf. [33]). Furthermore, references to the anaphora and antecedents represented by markables in the corresponding annotation layer can be declared. The markable XML Schema defines markable elements and their potential attributes. The attributes contain information on several features like grammatical function, morphological information (number, gender, case, person), the part-of-speech of the markable's head and some others. While markables can be various parts of speech, in our case they are considered to be nominal entities which can be part of an anaphoric relation.

Special filter plugins for these layers were established to allow Serengeti the import of pre-annotated data. For AnaWiki we apply a pipeline of scripts to get from raw text to preannotated SGF format:

1. Automated normalisation, sentence splitting, and tokenisation of the input text using the *openNLP*[10] toolkit.
2. Automated POS tagging using the *Berkeley Parser*[11].
3. Automated markable recognition.
4. Heuristic identification of additional features associated with markables (e.g., person, case, number etc.). The output is MAS-XML.
5. MAS-XML is converted into SGF using the XSLT stylesheet described in Section 10.3.1.3.

The text layout is rendered according to the logical document structure and markables are introduced as described above. These predefined annotations are shared by all annotators, while modifications regarding markables and relations (adding, deleting, editing) only take effect in the annotator's own annotation.

10.2.4 Using a Non-expert Annotation Game

One drawback of the expert annotation tool is that it requires linguistically trained users, i.e. experts. Collaborative resource creation using the general Web population offers a different approach. The motivation for this is the observation that a group of individuals can contribute to a collective solution which has a better performance and is more robust than an individual's solution as demonstrated in simulations of collective behaviours in self-organising systems [17].

Wikipedia[12] is perhaps the best example of collaborative resource creation, but it is not an isolated case. The gaming approach to data collection, termed *Games with a purpose*, has received increased attention since the success of the ESP game [1]. Interestingly, the *Games with a purpose* concept has now also been adopted by the Semantic Web community in an attempt to collect large-scale ontological knowledge because currently "the Semantic Web lacks sufficient user involvement almost everywhere" [30].

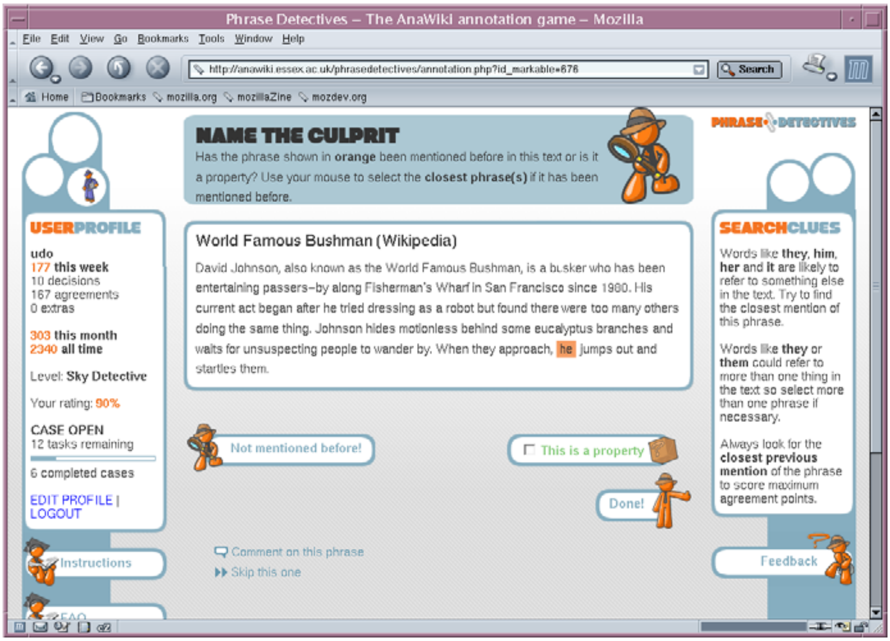
10.2.4.1 The Phrase Detectives Game

Phrase Detectives offers a game interface for the Anaphoric Bank to collect annotations from the general Web population [4]. This creates a pool of potential annotators that is magnitudes larger than the expert annotators we can hope to recruit using the expert annotation tool Serengeti. It does however raise the question of data quality. Two obvious causes for poor quality annotation are a lack of understanding of the task and malicious behaviour on the part of the annotator. A range of control mechanisms are in place to ensure quality annotation in Phrase Detectives [20].

[10] `http://opennlp.sourceforge.net/`

[11] `http://nlp.cs.berkeley.edu/`

[12] `http://www.wikipedia.org/`


Phrase Detectives – The AnaWiki annotation game – Mozilla
File Edit View Go Bookmarks Tools Window Help
http://anawiki.essex.ac.uk/phrasedetectives/annotation.php?id_markable=676
Search
Home Bookmarks mozilla.org mozillaZine mozdev.org
PHRASE DETECTIVES
Not mentioned before!
□ This is a property
Done!
Instructions
FAQ
Feedback
Comment on this phrase
Skip this one

NAME THE CULPRIT

Has the phrase shown in **orange** been mentioned before in this text or is it a property? Use your mouse to select the **closest phrase(s)** if it has been mentioned before.

USERPROFILE

udo
177 **this week**
10 decisions
167 agreements
0 extras

303 **this month**
2340 **all time**

Level: **Sky Detective**

Your rating: 90%

CASE OPEN
12 tasks remaining

6 completed cases

EDIT PROFILE | LOGOUT

World Famous Bushman (Wikipedia)

David Johnson, also known as the World Famous Bushman, is a busker who has been entertaining passers-by along Fisherman's Wharf in San Francisco since 1980. His current act began after he tried dressing as a robot but found there were too many others doing the same thing. Johnson hides motionless behind some eucalyptus branches and waits for unsuspecting people to wander by. When they approach, he jumps out and startles them.

SEARCHCLUES

Words like **they, him, her** and **it** are likely to refer to something else in the text. Try to find the closest mention of this phrase.

Words like **they** or **them** could refer to more than one thing in the text so select more than one phrase if necessary.

Always look for the **closest previous mention** of the phrase to score maximum agreement points.

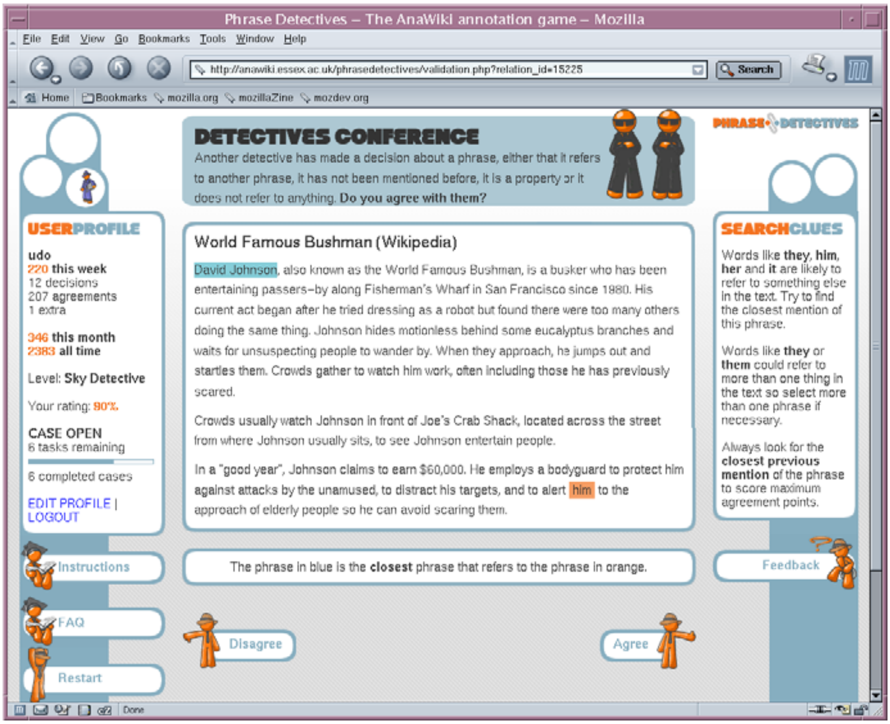
Fig. 10.2 A screenshot of the Annotation Mode.

There are two ways to annotate within the game: by selecting a markable that corefers to another highlighted markable (Annotation Mode – see Figure 10.2); or by validating a decision previously submitted by another player (Validation Mode – see Figure 10.3).

Players begin the game at the training level where they are given a set of annotation tasks created from the Gold Standard, i.e. texts that have been annotated by experts using the Serengeti annotation tool. They are given feedback and guidance when they select an incorrect answer and points when they select the correct answer. When the player gives enough correct answers they graduate to annotating texts that will be included in the corpus.

Occasionally, a graduated player will be covertly given a Gold Standard text to annotate. A bonus screen will be shown when the player has completed annotating the text indicating what the player selected incorrectly, with bonus points for agreeing with the Gold Standard. This is the foundation of a player rating system to judge the quality of the player's annotations.

The game is designed to motivate players to annotate the text correctly by using comparative scoring (awarding points for agreeing with the Gold Standard), and collaborative scoring (players gain points by agreeing with other players, these points may be awarded retrospectively once new annotations have been submitted by other players).


Phrase Detectives – The AnaWiki annotation game – Mozilla
File Edit View Go Bookmarks Tools Window Help
http://anawiki.essex.ac.uk/phrasedetectives/validation.php?relation_id=15225
Search
Home Bookmarks mozilla.org mozillaZine mozdev.org
PHRASE DETECTIVES
Instructions
FAQ
Restart
Feedback
Disagree
Agree
Done

DETECTIVES CONFERENCE

Another detective has made a decision about a phrase, either that it refers to another phrase, it has not been mentioned before, it is a property or it does not refer to anything. **Do you agree with them?**

USERPROFILE

udo
220 **this week**
12 decisions
207 agreements
1 extra

346 **this month**
2383 **all time**

Level: **Sky Detective**

Your rating: 90%.

CASE OPEN
6 tasks remaining

6 completed cases

EDIT PROFILE | LOGOUT

World Famous Bushman (Wikipedia)

David Johnson, also known as the World Famous Bushman, is a busker who has been entertaining passers–by along Fisherman's Wharf in San Francisco since 1980. His current act began after he tried dressing as a robot but found there were too many others doing the same thing. Johnson hides motionless behind some eucalyptus branches and waits for unsuspecting people to wander by. When they approach, he jumps out and startles them. Crowds gather to watch him work, often including those he has previously scared.

Crowds usually watch Johnson in front of Joe's Crab Shack, located across the street from where Johnson usually sits, to see Johnson entertain people.

In a "good year", Johnson claims to earn $60,000. He employs a bodyguard to protect him against attacks by the unamused, to distract his targets, and to alert him to the approach of elderly people so he can avoid scaring them.

SEARCHCLUES

Words like **they**, **him**, **her** and **it** are likely to refer to something else in the text. Try to find the closest mention of this phrase.

Words like **they** or **them** could refer to more than one thing in the text so select more than one phrase if necessary.

Always look for the **closest previous mention** of the phrase to score maximum agreement points.

The phrase in blue is the **closest** phrase that refers to the phrase in orange.

Fig. 10.3 A screenshot of the Validation Mode.

10.2.4.2 The Data

Ambiguity is an inherent problem in all areas of NLP [18]. Here we are not interested in solving this issue but in capturing ambiguity where it is appropriate. If an anaphora is ambiguous, then the annotated corpus should capture this information. We are therefore not aiming at selecting 'the best' or most common annotation but to preserve all inherent ambiguity (which is supported by the output formats as discussed later on).

The beta version of Phrase Detectives went online in May 2008, with the first live release in December 2008. Initially over 100,000 words of text from Project Gutenberg[13] and Wikipedia were automatically parsed to identify the markables and added to the game. The game now accesses a corpus of more than a million words in different languages [20].

In less than three months of live release the game has collected over 100,000 annotations of anaphoric relations provided by more than 500 players. To put this in perspective, the GNOME corpus, produced by traditional methods, included around 3,000 annotations [25] and the Sekimo corpus around 4,000 annotations of anaphoric relations [37].

[13] http://www.gutenberg.org/

10.3 Architecture for the Anaphoric Bank

Apart from the pivot formats already mentioned in Section 10.2.2 there are two parts of the architecture that should be examined further: the XML part and the database part.

10.3.1 SGF – The XML Architecture Part

Usually one does not want to start over with a fresh corpus but wants to use an existing corpus. As already mentioned in Section 10.2.2, different pivot formats are available. One of these formats is SGF, the Sekimo Generic Format.

As a successor of the Prolog fact base format used in the first project phase [36] the Sekimo Generic Format (SGF) has been developed in the Sekimo project following these design goals: import and storage of multiple annotation layers, possibility of analysing semantic relations without any transformation beforehand, usage as an exchange format for the Serengeti web-based annotation tool (and other similar tools). Since SGF is not focussed on the single task of anaphora resolution which was intended in the Sekimo project, the format uses a standoff approach [34] following the Annotation Graph's formal model [2]. This makes it possible to use SGF for a large variety of linguistic annotations, including lexical chaining [35], multimodal annotation or diachronic corpora. For this reason the format tries to reuse the structure and features of existing annotation formats. As a result, SGF itself consists only of the declaration of a base layer which provides the primary data (i.e., the data that is annotated) and its segmentation, and serves as a container for standoff representations of the original inline annotation (these are converted automatically using an XSLT stylesheet). Figure 10.4 shows a graphical representation of SGF's `corpusData` element[14].

Either textual or multimodal data can serve as primary data in SGF (in the former case one may include the primary data in the SGF instance or use an external file, in the latter case the primary data file is referenced via a URI attribute at all times), the use of multiple primary data instances is supported as well (e.g., for annotating multimodal or diachronic corpora). Metadata can be supplied at various locations of an SGF instance, either underneath the `corpus` element (for information regarding the whole corpus) or underneath a `corpusData` entry (denoting metadata related to a single corpus item). In addition, the `meta` element can be located as a child of the `primaryData` or the `annotation` element. In all cases it is possible to include the metadata in the SGF instance or to refer to an external file. No restrictions apply to the structure of the metadata (even non-XML metadata is allowed), therefore the reuse of established metadata specifications such as OLAC [29], IMDI [15, 16], DublinCore [6], or the Component Metadata Infrastructure (CMDI) [3] is encouraged.

[14] Note that an SGF instance's root element is either the `corpusData` or the `corpus` element, thus allowing to store a whole corpus or a single primary data aligned with its annotations.

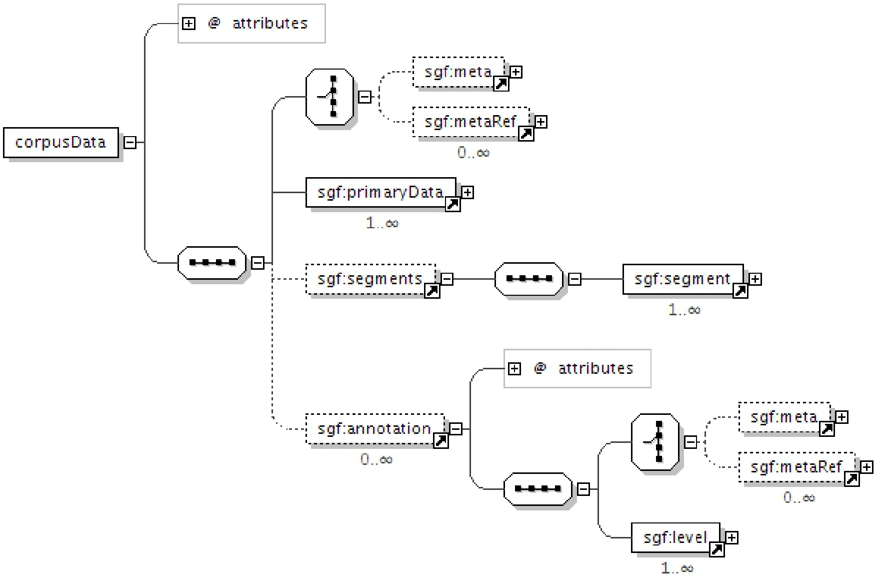


Fig. 10.4 SGF's `corpusData` element containing (optional) metadata, the primary data, its segmentation and annotation level(s).

10.3.1.1 Segmentation and Import of Annotations

Segmentation of the primary data is application driven, i.e. for textual primary data segmentation is usually character based.

Segments are established by importing a single inline annotation of the primary data (e.g., as a result of the application of a parser, a chunker, or similar linguistic resources, cf., e.g., listing 10.2). Similar to approaches such as PAULA or GRAF, for each annotation element the start and end positions relative to the character stream are computed (cf. listing 10.1 and Section 10.3.1.3) and a `segment` element is created underneath the `segments` parent.

Listing 10.1 Using the character positions for defining segmentation of the input text.

```
  T  h  e     s  u  n     s  h  i  n  e  s     b  r  i  g  h  t  e  r  .
00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19]20|21|22|23|24
```

Segments delimiting the same part of the primary data are only created once, even if used in different annotation layers, since the segmentation is independent of the annotation (even overlapping segments are possible). After the creation of the corresponding `segment` element, a converted (i.e. standoff) representation of the original annotation is stored. An example instance is shown below (cf. listings 10.2 and 10.3).

Listing 10.2 Inline annotation as a result of a morpheme tagging process of the input sentence “The sun shines brighter”.

```
<morphemes xmlns="http://www.text-technology.de/sekimo/morphemes">
  <morpheme>The</morpheme>
  <morpheme>sun</morpheme>
  <morpheme>shine</morpheme>
  <morpheme>s</morpheme>
  <morpheme>bright</morpheme>
  <morpheme>er</morpheme>.
</morphemes>
```

Note that SGF distinguishes between the annotation *level* as the concept used for the annotation and the annotation *layer* as the XML serialisation, i.e. it is possible to sum up different XML serialisations of the same linguistic concept (e.g., logical document structure, POS annotation) underneath the same `level` element (cf. Figure 10.5). As a second option it is allowed to combine different annotation levels (i.e. linguistic concepts) together with their respective layers underneath the same `annotation` element which can be useful to subsume different linguistic concepts that are realised in a single XML document grammar (e.g., discourse entities and semantic relations).

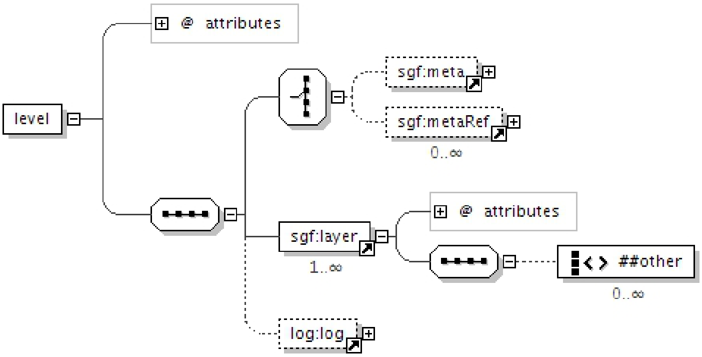


Fig. 10.5 SGF’s `level` element containing the converted standoff representations of the annotation levels (i.e. the annotation layer).

Since the `layer` element is a wrapper for elements derived from different XML namespaces the original annotation format has to be changed only slightly: elements with a former mixed content model are converted into pure container elements, elements containing text nodes are converted into empty elements, and the `sgf:segment` attribute is added to former non-empty elements as an attribute. This conversion applies both to the XML instance and the underlying document grammar (SGF relies on XML schema descriptions). In the latter case it is possible to only import the sgf:segment attribute as an optional attribute and therefore use the converted schema representation for both the original (inline) representation and the converted representation as part of the SGF instance. Listing 10.3 shows the SGF instance containing the morpheme annotation level.

Listing 10.3 The SGF instance including the converted morpheme annotation layer.

```
<sgf:corpusData xmlns:sgf="http://www.text-technology.de/sekimo"
     sgfVersion="1.1" xml:id="s_m1">
   <sgf:meta><!-- meta data goes in here --></sgf:meta>
   <sgf:primaryData start="0" end="24" xml:lang="en">
      <textualContent>The sun shines brighter.</textualContent>
   </sgf:primaryData>
   <sgf:segments>
      <sgf:segment xml:id="seg1" type="char" start="0" end="24"/>
      <sgf:segment xml:id="seg2" type="char" start="0" end="3"/>
      <sgf:segment xml:id="seg3" type="char" start="4" end="7"/>
      <sgf:segment xml:id="seg4" type="char" start="8" end="13"/>
      <sgf:segment xml:id="seg5" type="char" start="13" end="14"/>
      <sgf:segment xml:id="seg6" type="char" start="15" end="21"/>
      <sgf:segment xml:id="seg7" type="char" start="21" end="23"/>
   </sgf:segments>
   <sgf:annotation xml:id="a_morph">
      <sgf:level xml:id="a_morph_layer" priority="0">
         <sgf:meta><!-- meta data goes in here --></sgf:meta>
         <sgf:layer xmlns:morph="http://www.text-technology.de/sekimo/
              morphemes">
            <morph:morphems sgf:segment="seg1">
               <morph:morphem sgf:segment="seg2"/>
               <morph:morphem sgf:segment="seg3"/>
               <morph:morphem sgf:segment="seg4"/>
               <morph:morphem sgf:segment="seg5"/>
               <morph:morphem sgf:segment="seg6"/>
               <morph:morphem sgf:segment="seg7"/>
            </morph:morphems>
         </sgf:layer>
      </sgf:level>
   </sgf:annotation>
</sgf:corpusData>
```

10.3.1.2 Adding Layers

For storing a single annotation layer SGF is quite verbose. However, if we want to add another annotation, e.g., a syllable annotation ranging over the very same input text, such as the one shown in listing 10.4, we cannot combine the elements derived from this annotation layer with the already established morpheme annotation because of overlapping structures which are not allowed in XML instances.

Listing 10.4 A syllable annotation.

```
<syllables xmlns="http://www.text-technology.de/sekimo/syllables"
     xml:lang="en">
   <syllable>The</syllable>
   <syllable>sun</syllable>
   <syllable>shines</syllable>
   <syllable>brigh</syllable>
   <syllable>ter</syllable>.
</syllables>
```

Since SGF uses a standoff approach it is easily possible to add the additional syllable annotation level to the already established SGF instance. Given that most of the syllable elements share their ranges with the segments defined by the elements of the morpheme level, only three new segment elements have to be defined: seg8, seg9 and seg10 in listing 10.5.

Listing 10.5 The SGF instance containing both annotation levels.

```
<sgf:corpusData xmlns:sgf="http://www.text-technology.de/sekimo" xml:id=
    "s_m_s" sgfVersion="1.1">
   <sgf:meta><!-- meta data goes in here --></sgf:meta>
   <sgf:primaryData start="0" end="24" xml:lang="en">
      <textualContent>The sun shines brighter.</textualContent>
   </sgf:primaryData>
   <sgf:segments>
      <sgf:segment xml:id="seg1" type="char" start="0" end="24"/>
      <sgf:segment xml:id="seg2" type="char" start="0" end="3"/>
      <sgf:segment xml:id="seg3" type="char" start="4" end="7"/>
      <!-- manually shortened, cf. listing 3 ... -->
      <sgf:segment xml:id="seg7" type="char" start="21" end="23"/>
      <sgf:segment xml:id="seg8" type="char" start="8" end="14"/>
      <sgf:segment xml:id="seg9" type="char" start="15" end="20"/>
      <sgf:segment xml:id="seg10" type="char" start="20" end="23"/>
   </sgf:segments>
   <sgf:annotation xml:id="a_syll">
      <sgf:level xml:id="l_syll">
         <sgf:meta><!-- meta data goes in here --></sgf:meta>
         <sgf:layer xmlns:morph="http://www.text-technology.de/sekimo/
              syllables" priority="0">
            <syll:syllables sgf:segment="seg1" xml:lang="en">
               <syll:syllable sgf:segment="seg2"/>
               <syll:syllable sgf:segment="seg3"/>
               <syll:syllable sgf:segment="seg8"/>
               <syll:syllable sgf:segment="seg9"/>
               <syll:syllable sgf:segment="seg10"/>
            </syll:syllables>
         </sgf:layer>
      </sgf:level>
   </sgf:annotation>
   <sgf:annotation xml:id="a_morph">
     <!-- See listing 3, line 16 - 28 -->
   </sgf:annotation>
</sgf:corpusData>
```

In contrast to similar approaches the hierarchical structure of the imported annotation layers remains intact (i.e. there is still a parent child relationship between the syllables and syllable elements) as well as the attribute information (if there is any – in this example the xml:lang attribute). The only information that is dismissed are the text nodes which are referred to by the segment elements and the respective sgf:segment attributes.

For the Anaphoric Bank SGF uses the three annotation levels that are discussed in Section 10.2.3.2, namely the logical document structure, the markable level and the semantic relation level derived from the further developed ARRAU scheme. The resulting SGF instances are imported into the web-based annotation tool Serengeti that serves as the Anaphoric Bank's Expert Annotation Tool (cf. Section 10.2.3).

10.3.1.3 SGF Tools

The conversion of an inline annotation into an SGF instance containing a single annotation level is performed by the XSLT stylesheet `inline2SGF.xsl`. The transformation requires an XSLT 2.0 processor (e.g., Saxon[15] which was used for the transformation). To call the transformation one has to provide the name of an input file containing an inline annotation. The primary data, i.e., the non-annotated text file that deals as the basis of all annotation tasks, can be provided as stylesheet parameter *$primary-data* or can be inferred from the textual content of the input document. The exact execution call differs with respect to the used processor and with the stylesheet parameters used in the transformation call[16].

The first basic task of the stylesheet is to create segments on the basis of the XML element boundaries in the inline annotation. By means of the primary data file, start and end positions of the XML elements can be calculated, stored as `segment` element and referenced later on. There is the possibility to create optional segments for white-spaces and other non-character data, too. Their inclusion can be controlled by a boolean stylesheet parameter.

Afterwards the elements from the input annotation are copied into single layers by separating them with respect to their XML namespace. Thus for every namespace in the input, an `annotation` element is created and the converted (i.e. standoff) representation of the annotation bound to this namespace is copied as a `layer` underneath the `annotation` element.

In addition to the conversion of a single inline annotation to an SGF instance containing a single `annotation` element, a second XSLT stylesheet `mergeSGF.xsl` is available for merging SGF files based on the same primary data. By this means different annotations can be combined into a single SGF instance without manually dealing with duplicate `segment` elements or duplicate segment identifiers. This stylesheet was optimized in speed and memory consumption to work in production scenarios.

10.3.1.4 Validation and Analysis

Since SGF makes heavy use of XML namespaces for the separation of annotation layers and XML's inherent ID/IDREF mechanism for linking segments to their respective annotation elements, each annotation layer must provide its own XML document grammar. SGF itself is based on XML schema and demands XML schemas for the imported annotation layers as well. In return it is possible to not only validate the SGF instance as a whole but also the entirety of annotation layers and – with the usage of XSLT or XQuery – to do cross layer validation.

SGF can be used for analysing correlations between elements derived from different annotation layers. For a detailed discussion of the format and its application

[15] Information regarding the Saxon XSLT processor can be found at `http://saxon.sourceforge.net` and `http://www.saxonica.com/`.

[16] See the *Resources* Section at `http://www.text-technology.de/sekimo` for a documentation of the tools mentioned.

in the field of anaphora resolution cf. [31] and [37]. Both, SGF and its currently developed successor, XSTANDOFF [32], are freely available under the GNU Lesser General Public License (LGPL v3) including the accompanied tools discussed in Section 10.3.1.3 and a large variety of example annotations[17].

10.3.2 Database Format

Natively the Sekimo Generic format is formulated in XML Schema. But for the application in the Anaphoric Bank a relational database representation of SGF has been developed, transforming all information without loss from XML to a relational database management system and vice versa. Figure 10.6 shows the conceptual database model of the relational database representation of SGF used in Serengeti as well as in Phrase Detectives.

This coherent design in corpus data management allows for fast access of the data, by storing regularly accessed data more efficiently than in XML. We prefer SQL to plain XML or XML databases because of its performance and the ease of integration with other system parts, for example the user management.

For the database management an application programming interface (API) was implemented as Perl classes, used by Serengeti and special administration tools for importing and exporting corpus data. This SGF-API, working as an *object relational mapper* for the database, can be extended by plugins, serving as filters for the import and export as well as objects (in an object oriented sense) for the traversing and manipulation of special corpus data. For the AnaWiki project, three filters were established for markables, relations and the logical document structure (cf. 10.2.3.2). Thus the database representation of the AnaWiki data can be separated from the SGF core (see Figure 10.7).

Plugins detect their corresponding layers during import by recognising the XML namespaces. The complete layer then is assigned to the plugin. When traversing, modifying (e.g., in Serengeti) or exporting the data, the plugin provides all necessary methods by extending a prototyped layer object of the API. Layers in unknown namespaces are stored in a representation compatible with the document object model, providing methods for traversing and modifying according to the DOM specification[18].

The Phrase Detectives game has additional tables in the database to store information about the players, scores, ratings etc. as well as additional information required for documents and a simplified version of the `ana_markable` table. This table (`pd_markable`) connects to the original file and immediately after import is identical to the `ana_markable` table. Once the document is either used in the game or in Serengeti the tables become asynchronous. The game stores markables

[17] See
`http://www.text-technology.de/sekimo`
and `http://www.xstandoff.net/` for further details.

[18] The Document Object Model interface is specified in
`http://www.w3.org/TR/DOM-Level-3-Core/`.

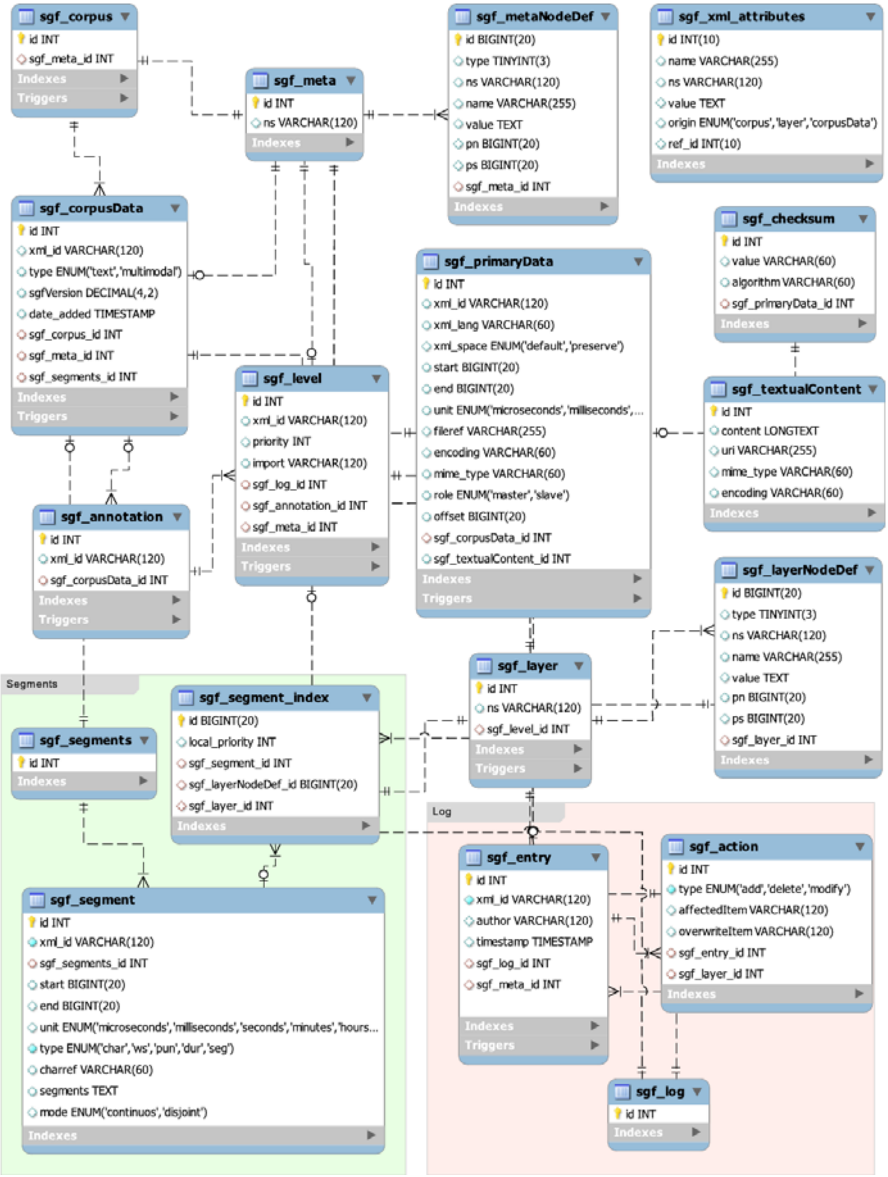


Fig. 10.6 The relational database representation of SGF.

and relations slightly different to Serengeti but exports into MAS-XML and SGF format so the two annotations can be compared. Exported documents from the game can be imported into Serengeti and vice versa.

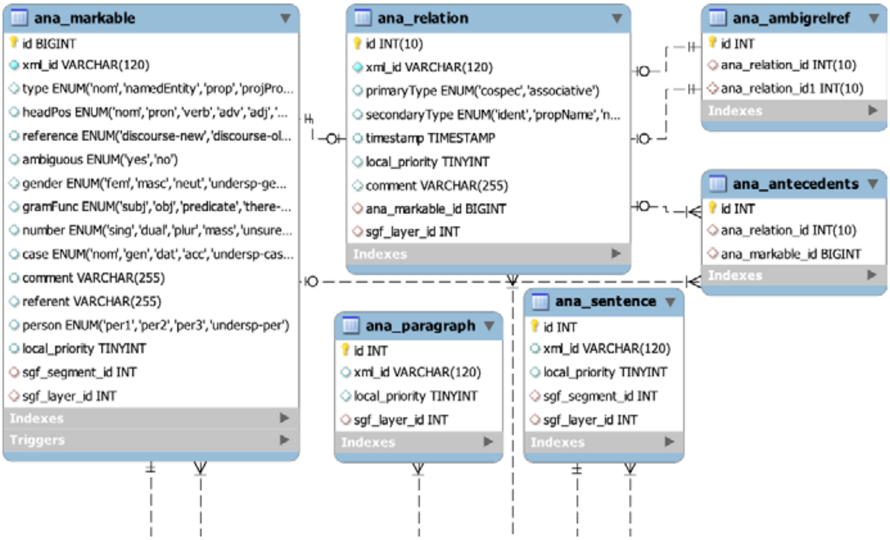


Fig. 10.7 The relational database representation of the AnaWiki layers.

10.4 Conclusion

To our knowledge, the Anaphoric Bank is the first attempt to build a large anaphorically annotated corpus by collaboration–via sharing of previously annotated resources or through Web annotation. Recent developments in XML technology such as pivot representations and Web annotation tools greatly facilitate this effort but a number of technical issues still have to be addressed. In this chapter we discussed the solutions developed to facilitate the sharing of data created via Web collaboration. Work is ongoing on the selection of a pivot representation and the development of import / export converters to allow for contribution.

Acknowledgments

The Anaphoric Bank is a joint research effort. Current participants include, apart from the University of Essex, United Kingdom and the Bielefeld University, Germany, the Copenhagen Business School, Denmark; EML Research, Heidelberg, Germany; LORIA, Nancy, France; University of Potsdam, Germany; PURCS, Brazil; University of Trento, Italy; and the University of Wolverhampton, United Kingdom.

The development of Phrase Detectives and of the Anaphoric Bank infrastructure was supported by the EPSRC grant AnaWiki, EP/F00575X/1.

Serengeti and SGF have been developed in the Sekimo project, a part of the distributed Research Group *Text-technological modelling of information*[19], funded by the German Research Foundation.

We want to thank the anonymous reviewers who helped to improve this manuscript.

[19] See `http://www.text-technology.de/` for further details.

References

[1] von Ahn, L.: Games with a purpose. Computer 39(6), 92–94 (2006)

[2] Bird, S., Liberman, M.: Annotation graphs as a framework for multidimensional linguistic data analysis. In: Proceedings of the Workshop ”Towards Standards and Tools for Discourse Tagging”, Association for Computational Linguistics, pp. 1–10 (1999), http://xxx.lanl.gov/abs/cs.CL/9907003

[3] Broeder, D., Kemps-Snijders, M., Uytvanck, D.V., Windhouwer, M., Withers, P., Wittenburg, P., Zinn, C.: A data category registry- and component-based metadata framework. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC), European Language Resources Association (ELRA), Valletta, Malta, pp. 43–47 (2010)

[4] Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase Detectives: A Web-based collaborative annotation game. In: iSemantics (2008)

[5] Clark, H.H.: Bridging. In: Johnson-Laird, P.N., Wason, P.C. (eds.) Thinking: Readings in Cognitive Science, pp. 411–420. Cambridge University Press, Cambridge (1977)

[6] DCMI Usage Board, DCMI Metadata Terms. DCMI Recommendation, Dublin Core Metadata Initiative (2006),
http://dublincore.org/documents/dcmi-terms/

[7] van Deemter, K., Kibble, R.: On coreferring: Coreference in MUC and related annotation schemes. Computational Linguistics 26(4), 629–637 (2000)

[8] Diewald, N.: Serengeti – A brief Starting Guide. Technical manual (2008),
http://www.text-technology.de/publications/serengeti_guide.pdf

[9] Diewald, N., Stührenberg, M., Garbar, A., Goecke, D.: Serengeti – Webbasierte Annotation semantischer Relationen. LDV Forum 23(2) (2008)

[10] Dipper, S.: XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In: Proceedings of Berliner XML Tage 2005 (BXML 2005), Berlin, Germany, pp. 39–50 (2005)

[11] Garrett, J.J.: Ajax: A new approach to web applications (2005),
http://adaptivepath.com/ideas/essays/archives/000385.php,
http://adaptivepath.com/ideas/essays/archives/000385.php

[12] Hirschman, L.: MUC-7 coreference task definition, version 3.0. In: Chinchor, N. (ed.) Proceedings of the 7th Message Understanding Conference (1998),
http://www.muc.saic.com/proceedings/muc_7_toc.html

[13] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proc. HLT-NAACL (2006)

[14] Ide, N., Suderman, K.: GrAF: A Graph-based Format for Linguistic Annotations. In: Proceedings of the Linguistic Annotation Workshop, Association for Computational Linguistics, Prague, Czech Republic, pp. 1–8 (2007)

[15] IMDI (ISLE Metadata Initiative) Metadata Elements for Session Descriptions. version 3.0.4. Reference Document, MPI, Nijmegen (2003),
http://www.mpi.nl/IMDI/documents/Proposals/IMDI_MetaData_3.0.4.pdf

[16] IMDI (ISLE Metadata Initiative) Metadata Elements for Catalogue Descriptions. version 3.0.0. Tech. rep., MPI, Nijmegen (2004),
http://www.mpi.nl/IMDI/documents/Proposals/IMDI_Catalogue_3.0.0.pdf

[17] Johnson, N.L., Rasmussen, S., Joslyn, C., Rocha, L., Smith, S., Kantor, M.: Symbiotic Intelligence: Self-Organizing Knowledge on Distributed Networks Driven by Human Interaction. In: Proceedings of the Sixth International Conference on Artificial Life. MIT Press, Cambridge (1998)

[18] Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice-Hall, Englewood Cliffs (2008)

[19] Krasavina, O., Chiarcos, C.: PoCoS – Potsdam Coreference Scheme. In: Proceedings of The Linguistic Annotation Workshop, Association for Computational Linguistics, pp. 156–163 (2007), http://acl.ldc.upenn.edu/W/W07/W07-1525.pdf

[20] Kruschwitz, U., Chamberlain, J., Poesio, M.: (Linguistic) Science Through Web Collaboration in the ANAWIKI Project. In: Proceedings of WebSci 2009, Athens (2009)

[21] Morton, T., LaCivita, J.: WordFreak: An Open Tool for Linguistic Annotation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada, pp. 17–18 (2003)

[22] Müller, C., Strube, M.: Multi-level annotation of linguistic data with mmax2. In: Braun, S., Kohn, K., Mukherjee, J. (eds.) Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods, English Corpus Linguistics, Peter Lang, vol. 3, pp. 197–214 (2006)

[23] Navarretta, C.: Abstract anaphora resolution in Danish. In: Dybkjaer, L., Hasida, K., Traum, D. (eds.) Proc. of the 1st SIGdial Workshop on Discourse and Dialogue, ACL, pp. 56–65 (2000)

[24] Orăsan C, PALinkA: A highly customisable tool for discourse annotation. In: Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue, Sapporo, Japan (2003)

[25] Poesio, M.: Discourse annotation and semantic annotation in the GNOME corpus. In: Proc. of the ACL Workshop on Discourse Annotation, Barcelona, pp. 72–79 (2004)

[26] Poesio, M.: The MATE/GNOME scheme for anaphoric annotation, revisited. In: Proceedings of SIGDIAL, Boston (2004)

[27] Poesio, M., Artstein, R.: The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In: Proceedings of The ACL Workshop on Frontiers in Corpus Annotation, Association for Computational Linguistics, pp. 76–83 (2005), http://acl.ldc.upenn.edu/W/W05/W05-0311.pdf

[28] Sasaki, F., Wegener, C., Witt, A., Metzing, D., Pönninghaus, J.: Co-reference annotation and resources: A multilingual corpus of typologically diverse languages. In: Proceedings of the 3nd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, Spain (2002)

[29] Simons, G., Bird, S.: OLAC Metadata. OLAC: Open Language Archives Community (2003), http://www.language-archives.org/OLAC/metadata.html

[30] Siorpaes, K., Hepp, M.: Games with a purpose for the semantic web. IEEE Intelligent Systems 23(3), 50–60 (2008)

[31] Stührenberg, M., Goecke, D.: SGF – An integrated model for multiple annotations and its application in a linguistic domain. In: Proceedings of Balisage: The Markup Conference, Montreal, Kanada (2008), http://www.balisage.net/Proceedings/html/2008/Stuehrenberg01/Balisage2008-Stuehrenberg01.html

[32] Stührenberg, M., Jettka, D.: A toolkit for multi-dimensional markup: The development of SGF to XStandoff. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec, Balisage Series on Markup Technologies (2009)

[33] Stührenberg, M., Goecke, D., Diewald, N., Cramer, I., Mehler, A.: Webbased Annotation of Anaphoric Relations and Lexical Chains. In: Proceedings of The Linguistic Annotation Workshop, Association for Computational Linguistics, pp. 140–147 (2007), http://acl.ldc.upenn.edu/W/W07/W07-1523.pdf

[34] Thompson, H.S., McKelvie, D.: Hyperlink semantics for standoff markup of read-only documents. In: Proceedings of SGML Europe 1997: The next decade – Pushing the Envelope, Barcelona, pp. 227–229 (1997), http://www.ltg.ed.ac.uk/~ht/sgmleu97.html

[35] Waltinger, U., Mehler, A., Stührenberg, M.: An integrated model of lexical chaining: application, resources and its format. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds) KONVENS 2008 – Ergänzungsband Textressourcen und lexikalisches Wissen, Berlin, pp. 59–70 (2008)

[36] Witt, A., Goecke, D., Sasaki, F., Lüngen, H.: Unification of XML Documents with Concurrent Markup. Literary and Lingustic Computing 20(1), 103–116 (2005)

[37] Witt, A., Stührenberg, M., Goecke, D., Metzing, D.: Integrated linguistic annotation models and their application in the domain of antecedent detection. In: Mehler, A., Kühnberger, K.U., Lobin H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modelling, Learning and Processing of Text Technological Data Structures, Studies in Computational Intelligence, Springer, Heidelberg (2011)

Chapter 11
Integrated Linguistic Annotation Models and Their Application in the Domain of Antecedent Detection

Andreas Witt, Maik Stührenberg, Daniela Goecke, and Dieter Metzing

Abstract. Seamless integration of various, often heterogeneous linguistic resources in terms of their output formats and a combined analysis of the respective annotation layers are crucial tasks for linguistic research. After a decade of concentration on the development of formats to structure single annotations for specific linguistic issues, in the last years a variety of specifications to store multiple annotations over the same primary data has been developed. The paper focuses on the integration of the knowledge resource *logical document structure information* into a text document to enhance the task of automatic anaphora resolution both for the task of candidate detection and antecedent selection. The paper investigates data structures necessary for knowledge integration and retrieval.

11.1 Introduction

Anaphora Resolution (AR) describes the process of identifying the correct antecedent for a given anaphoric element and, in general, consists of three steps: (1) identification of anaphoric elements, (2) creation of a candidate set for each anaphora and (3) detection of the correct antecedent from the candidate set. In this paper we will focus on the second and third step and we will investigate the question how to create an appropriate candidate set.

In recent approaches that define anaphora resolution as a pairwise decision, the candidate set is created by choosing all candidates that precede a given anaphora

Andreas Witt
Institut für Deutsche Sprache, Zentrale Forschung, R5, 6 - 13, D-68016 Mannheim, Germany
e-mail: witt@ids-mannheim.de

Maik Stührenberg · Daniela Goecke · Dieter Metzing
Bielefeld University, Faculty of Linguistics and Literary Studies, Universitätsstraße 25,
D-33615 Bielefeld, Germany
e-mail: {maik.stuehrenberg,daniela.goecke,
dieter.metzing}@uni-bielefeld.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 197–218.
springerlink.com

or by using a fixed search window (e.g. in terms of sentences) and by collecting all discourse entities in this window [e.g. 52, 41, 46, 58]. Taking all preceding candidates into account works well for small texts, however for long texts this might lead to large candidate sets. The definition of an appropriate size of the search window is important inasmuch as a small window leads to errors due to the fact that the search window does not cover the correct antecedent at all and as a large window leads to large candidate sets which increases the possibility of preferring a wrong candidate over the correct one (for a discussion of the window size's impact on precision and recall values see [52]). Furthermore the computational effort increases due to the large number of candidates.

We argue that the approach of a fixed search window is not appropriate for long texts and thus not for all text types but that the search window has to be *flexible* in order to include the correct antecedent but to exclude those candidates that are least likely. How can we decide on the likelihood of an antecedent candidate? Current approaches of anaphora resolution are learning based, i.e. the likelihood of antecedent candidates is trained on a set of positive and false examples. However, in these approaches the candidate set is either created by taking all preceding discourse entities into account or by using a fixed window; for a given candidate set the most likely candidate is chosen. In our approach we investigate constraints in order to create an appropriate search window. We decide against a fixed search window due to two reasons:

1. With a fixed search window only antecedents can be found that lie within the given window size.
2. The search window cannot be enlarged arbitrarily as the size of the candidate list has negative impact on the resolution process.

Previous corpus investigation shows that linear distance between anaphora and antecedent is an important factor when creating an antecedent candidate set. Figure 11.1 shows linear distance of anaphoras of pronominal as well as of non-pronominal type in the corpus under investigation. The majority of pronominal anaphoras find their antecedents at a small distance whereas non-pronominal anaphoras find their antecedents even across large distances: For 26.8% of all non-pronominal anaphoras, the antecedent is found at a distance of two or more paragraphs. These anaphoras form 20.9% of all anaphoras occurring in the corpus under investigation. Previous investigations of the same corpus regarding the size of the search windows focused on linear distance in terms of discourse entities rather than sentences or paragraphs. For 50% of the direct anaphoric relations and 55.78% of the indirect anaphoric relations, the anaphoric element finds it antecedent within a distance up to 15 discourse entities (see [17]).

In this paper we will investigate how to resolve those anaphoras whose candidates lie outside a fixed window of one paragraph or 15 discourse entities. We will investigate the impact of hierarchical structure, especially logical document structure (LDS), on anaphora resolution. Why logical document structure might help to resolve anaphoric relations? The term logical document structure refers to the structure of a text in the sense of its formal composition and is in contrast to the

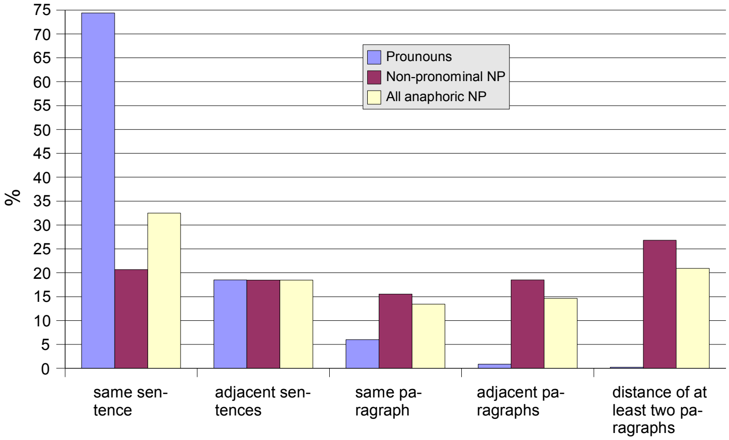


Fig. 11.1 Linear distance of anaphora and antecedent.

text's contentual composition, e.g. in terms of Introduction, Body or Conclusion. These contentual categories are realized by categories of the LDS, i.e. introduction, body and conclusion are realized as separate sections. The influence of the logical document structure on the choice of an antecedent might either be a direct influence on the markables (or antecedent life span) or an influence on the search window (see [15]). Thus, we investigate how to describe accessibility of antecedent candidates both in terms of linear as well as in terms of hierarchical distance. Accessibility is of special interest as linear distance between anaphora and antecedent might be large. The term *linear distance* is based on text structure and refers to syntagmatic distance between anaphora and antecedent in terms of words, discourse entities, sentences or paragraphs. *Hierarchical distance* describes distance between anaphora and antecedent on the basis of a hierarchical structure in terms of a tree structure, for example as found in discourse structure or logical document structure.

The remainder of the article is structured as follows: In Section 11.2, we provide the theoretical background of anaphora resolution and describe our categorial framework of anaphoric relations. In Section 11.3 we give an overview of logical document structure, describe the annotation of LDS and formulate our research questions regarding the use of LDS for anaphora resolution. In Section 11.4 we present annotation models that allow the investigation of different types of information and in Section 11.5 we will present the results of a corpus study investigating the use of LDS for anaphora resolution.

11.2 Anaphora Resolution

Anaphora Resolution (AR) describes the process of identifying for a given anaphoric element its correct antecedent in the previous textual context. The anaphoric element picks up its antecedent linguistically. In case of coreference, anaphora and antecedent refer to the same entity whereas in case of cospecification the anaphoric element picks up its antecedent linguistically but the two expressions are not coreferent. According to the relations that hold between the discourse entities, anaphora can be divided into direct anaphora and indirect anaphora. For direct anaphora, the antecedent is explicitly mentioned in the previous context (Example (1)) whereas for indirect anaphora the antecedent is not mentioned explicitly but has to be inferred from the context (Example (2)).

(1) I met a man yesterday. He told me a story.
(Example taken from [7], p. 414)

(2) I looked into the room. The ceiling was very high.
(Example taken from [7], p. 415)

Apart from the distinction of direct/indirect anaphora, discourse referents may be coreferent or not. In Example (1) the linguistic units “a man” and “he” are cospecified and refer to the same entity whereas “the room” and “the ceiling” in Example (2) do not although they are closely related due to world knowledge.

In this article we will investigate both direct and indirect anaphora as well as pronominal and definite description anaphora. We will focus on the question how to detect possible antecedent candidates from the set of discourse referents and how to select the correct one from the candidate set. The question how to detect possible candidates is of special interest as the linear distance between anaphora and antecedent might be large thus leading to a large set of candidates when using a fixed search window. In order to resolve anaphoric relations different types of information are needed. Information on discourse structure and referential accessibility is needed apart from information on POS, congruency, grammatical function and linear distance.

The corpus study is based on a corpus of German scientific articles that have been annotated manually for anaphoric relations. The annotation scheme comprises two primary relation types (direct and indirect anaphora) and a set of secondary relation types both for direct as well as for indirect anaphora. The annotation scheme is described in detail in [16]. The annotation has been done using the annotation tool SERENGETI [11] and has been checked for inter-annotator-agreement using kappa values [18]. Additional information for the resolution process has been added to the corpus by annotating the data automatically using the dependency parser MACHINESE SYNTAX[1] which provides lemmatization, POS information, dependency structure, morphological information and grammatical function. Based on this information, discourse entities have been detected automatically afterwards by identifying nominal heads (i.e. nouns or pronouns) and their pre-modifiers. Information

[1] http://www.connexor.eu/technology/machinese/machinesesyntax/

on logical document structure has been provided by the partner project C1 (see also Section 11.3.3).

11.3 Logical Document Structure

The aim of this section is to describe logical document structure as a structuring means of texts. LDS is a hierarchical structure: An article consists of sections which consist of subsections which consist of paragraphs. Furthermore, LDS describes the structure of a text – not its realization in a given medium, i.e. different realizations refer to the same structuring elements, e.g. paragraph boundaries or footnotes. Paragraph boundaries can be realized by line breaks with indentation or by blank lines (with or without following indentation). In print media, footnotes are often found at the bottom of the page whereas in hypertexts they are found at the end of the text and are linked via hyperlinks. In the next subsections we will provide a formal description of logical document structure and give an overview how LDS can be used for linguistic tasks.

11.3.1 What Is Logical Document Structure?

Formally, LDS forms a tree structure: Each section can contain several adjacent subsections, i.e. there are no overlapping arcs, and each subsection has exactly one parent section that contains it. Figure 11.2 shows the typical structure of a scientific article.

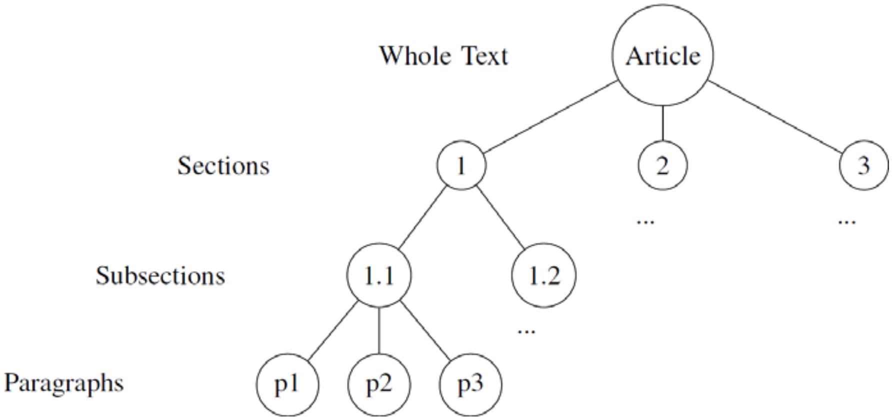


Fig. 11.2 Logical document structure of an article.

For a formal description of logical document structure as a tree, we follow the definition of [1]:

1. A single node by itself is a tree. This node is also the root of the tree.
2. Suppose n is a node and $T_1, T_2, ..., T_k$ are trees with roots $n_1, n_2 ..., n_k$ respectively. We can construct a new tree by making n the parent of nodes $n_1, n_2 ..., n_k$. In this tree n is the root and $T_1, T_2, ..., T_k$ are the subtrees of the root. Nodes $n_1, n_2 ..., n_k$ are called the *children* of node n.

(*ibid.* p. 75)

Knowledge about the expressiveness and complexity of LDS is important as it determines the means to describe and to annotate LDS in linguistic data. In terms of information modeling, the structuring elements shown in Figure 11.2 form a properly nested tree and thus follow the model of an ordered hierarchy of content objects (*OHCO*, cf. [10]). Each properly nested tree can be annotated using XML since the underlying formal model of XML is the tree – although extensions to this rule may apply according to the document grammar formalism that is used to define a specific markup language: e.g. DTDs are considered as tree-equivalent (extended) context-free grammars [cf. 22, p. 199] and [cf. 38, for a further discussion]. Any given XML annotation can be accessed by using XML tools: XPATH to traverse the tree and XSLT for further analyses. However, apart from their textual content, texts do contain objects that have to be converted into a tree structure in order to be annotated using XML, e.g. tables (cf. [31], p. 55ff). The application of LDS for linguistic tasks as well as its annotation for the corpus under investigation is described in the next subsections.

11.3.2 Application of Logical Document Structure for Linguistic Tasks

Information on logical document structure is applied for different linguistic tasks, e.g. language generation or genre detection. In this article we investigate the question whether LDS can be applied for the task of anaphora resolution.

Regarding language generation, [37] apply LDS (*abstract document structure* following the authors' terminology) in order to describe the abstract representation of a text – in contrast to its rhetorical structure or its realization (rendering). Whereas rhetorical structure is used to model the semantic content of a text, abstract document structure is used to model the hierarchical structure of textual entities. Abstract document structure is realized as a text using appropriate layout.

[34] describe the generation of referring expressions in hierarchically structured domains. [33] applies this framework for the domain of documents. Each document can be described as a hierarchical domain due to its hierarchical structure. For the task of language generation, a referring expression should allow for an easy identification of its referent. For hierarchically structured domains, information on the domain can be used to improve referring expressions in order to reduce the amount of search necessary to identify the referent. For a given document item, it

is necessary to identify the amount of information that is necessary to detect the referent, e.g. in order to refer appropriately to a picture item, information is needed whether the picture is located in the actual section or in another section.

Regarding anaphora resolution, the influence of the LDS on the choice of an antecedent might be either (a) a direct influence on the discourse entities (or antecedent life span), (b) an influence on the selection of a candidate according to the anaphora's and antecedent's position regarding the LDS or (c) an influence on the search window (comparable to different window sizes according to the NP type of the anaphora) (see also [15]). The first type is related to the fact that discourse entities "only serve as antecedents for anaphoric expressions within pragmatically determined segments" (cf. [52], p. 549).

Regarding LDS, previous investigation shows that some discourse entities are more prominent throughout the whole document than others, e.g. markables occurring in the abstract of a text might be accessible during the whole text whereas markables that occur in a list item or in a footnote-structure are less likely to be an antecedent for anaphoric elements in the main text. For a corpus of 4323 anaphoric relations 65.3% of all anaphora-antecedent-pairs are located in the same segment. Regarding the remaining anaphora-antecedent-pairs, we expect markables described in hierarchically higher elements (e.g. in a subsection) to be much more prone to finding their antecedents in structuring elements of a higher level (i.e. in a section) than in a preceding but hierarchically lower segment (i.e. in a preceding subsubsection). Thus, the influence on the search window may either enlarge the search window, i.e. the antecedent may be located outside the standard window (e.g. located in the whole paragraph or in a preceding one), or may narrow the search window, e.g. due to the start of a new chapter or section. Apart from defining an appropriate search window, the position of an antecedent candidate within a paragraph gives hints as to how likely that candidate is chosen as the correct one: 50.2% of the antecedents in the corpus are located paragraph-initial and 29.1% are located paragraph-final whereas only 20.2% are located in the middle of the paragraph. Thus, information on LDS might give information regarding the search window and for selecting the correct antecedent from a set of candidates (see also [47]).

In the following we will analyze how to apply these findings to antecedent detection and we will investigate the following research questions:

1. How are anaphora and antecedent located regarding LDS?
2. Does the position of the anaphora/the antecedent regarding LDS give hints for the antecedent choice?
3. Is it possible to define the search window by using information on the position of the anaphora/the antecedent?

In the next sections we will describe the annotation of LDS and the integration of different annotations layers for the task of analyzing their interrelationship and investigating the research questions.

11.3.3 XML-Annotation of Logical Document Structure

In order to investigate the influence of LDS on anaphora resolution we analyze a corpus regarding the research questions formulated above. The corpus under investigation has been annotated manually for anaphoric relations, additional information on lemmatization, POS, dependency structure, morphology and grammatical function as well as on discourse entities has been added afterwards (cf. Section 11.2). This information together with the annotation of the layer of logical document structure forms the basis for our analyses.

Apart from a set of newspaper articles that have been annotated in our project, we had the possibility to use an extensive set of annotations for scientific articles that have been annotated in the partner project C1. The annotation of the corpus data is based on an annotation scheme that has been defined by the partner projects C1 and B1 and which forms a subset of the DocBook annotation scheme with additional elements from (X)HTML. A detailed description of the annotation scheme as well as of the annotation procedure is given in [30, 28] and we will only give a brief overview here. For the annotation, a subset has been chosen from the complete set of elements from the DocBook standard (cf. [53]) which has been originally developed for technical documentation. The subset has been chosen in order to ease annotation by using only those elements that are needed for annotating the corpus of scientific articles. Another set of elements has been defined in order to describe elements that are not contained in the set of DocBook elements, e.g. elements for a table of contents which – in a standard DocBook creation process – is not annotated but created automatically from DocBook annotations. These elements are defined in a separate XML namespace. Another set of elements comprises XHTML-elements in order to describe e.g. link elements already annotated in the original corpus data. Altogether a set of 45 DocBook-elements and another 13 logical elements has been used for the annotation process. The annotation set thus comprises elements for describing the hierarchical structure of texts according to author, abstract, sections, paragraphs, footnotes, lists, list items, bibliography, tables, captions and the like.

The different annotation layers have been combined using markup unification which allows the combination of two XML annotation layers into a new XML instance [55]. The analysis of the research questions is based on the unified annotation data. This data is stored in a generic format that allows for creating different output formats, e.g. a candidate list (see Section 11.4.3), and for analyses using XSLT and XQuery. Different approaches for the integration of resources are presented in the next section.

11.4 Integration of Resources

In linguistic research often using only a single linguistic annotation layer is inadequate for dealing with specific tasks. This inadequacy does not only occur when one has to handle different linguistic levels, but can arise when working on a single representation level, e.g. [35] describe the problems when annotating multiword units on different lexical representation levels. Usually, annotation of linguistic representation levels is generated by linguistic resources, such as parsers, taggers, and the like. The integration of different resources is a crucial problem and, since the application of most linguistic resources results in heterogeneous output formats, i.e., XML instances following different document grammars that are only suitable for the given linguistic aspect this resource is aimed at, usually one encounters the problem of combining these different annotation layers that are all based on the same primary data. In this section we will present approaches to this problem.

11.4.1 Representation Formats

XML-based markup languages follow the formal model of a tree, i.e., the data that is structured by means of such a markup language is organized hierarchically as a tree (to be more specific: as a single tree) [59], similar to the above-mentioned OHCO model. Dealing with multi-dimensional annotation (i.e. multiple trees) and – as a result – with possibly overlapping structures is one of the key problems when working with XML-based annotation formats.

In the last years a variety of approaches has been developed to cope with overlapping structures. These proposals can be mainly divided into three categories: non-XML based approaches, XML-related approaches and XML-based approaches. The classic approach for dealing with multiple annotation layers is the use of separate documents or twin documents as [59] call them (if they share some annotation, the so-called sacred markup). [9] presents several formats that have been developed over the past years and that allow overlapping markup, starting from SGML's CONCUR feature [20] – a reimplementation approach named XCONCUR has been made by [21, 39, 56] –, over TEI milestones and fragmentation [5] and different standoff (i.e. the markup is separated from the primary data and stored in a separate document, [5, 51]) approaches up to specifications that leave the XML path, such as the Layered Markup and Annotation Language (LMNL, cf. [49, 8]) in conjunction with Trojan milestones following the HORSE (Hierarchy-Obfuscating Really Spiffy Encoding) or CLIX model. [44] discusses similar approaches for the formal representation of overlapping markup, adding colored XML [27] and the tabling approach described by [14] to the set of already stated proposals. Again, [59] compare state of the art in overlapping markup approaches, including alternatives to XML's data model (e.g. a directed acyclic graph structure (GODDAG, [45]) over the XML inherent tree) and its notation. In addition, the Prolog fact base approach discussed by [54, 55] or adding delay nodes to the XQuery 1.0 and XPath 2.0 Data Model (XDM) as virtual representation of nodes proposed by [29] allowing different nodes to share children describe other non-XML based specifications. Furthermore, XML-based specifications that follow the Annotation Graph paradigm [4], such as NITE [6], the Potsdamer Austauschformat für Linguistische Annotationen (PAULA, cf. [12, 13]), the Graph-based Format for Linguistic Annotations (GrAF, cf. [25]) developed by ISO/IEC TC37 or the Sekimo Generic Format (SGF) and its successor XStandoff (cf. section 11.4.2 and [47]) have been developed as well.

In case of using the classic approach of separate or twin documents the primary data (or source data, i.e. the textual data that is to be annotated) is saved together with a single annotation layer in separate files. Since only a single tree hierarchy is saved per file no overlapping structures occur. However, this approach might present problems in respect to the fact that the primary data is saved several times redundantly and analyzing relations between elements derived from different annotation layers may be cumbersome when dealing with multiple files without a linking element between the annotations. Although it is possible to use the character stream of the primary data as coordinates to align different annotation layers (cf. [55]), often changes to the primary data are introduced during the annotation process (in terms of added or deleted whitespace) raising further issues.

The Text Encoding Initiative proposes different XML-based solutions for dealing with complex markup (as multi dimensional markup is sometimes called): apart from stand-off markup, [5, chapters 16.9 and 20.4] there are milestone elements (empty elements that can be used as boundary markers, [5, chapter 20.2]) or fragmentations and joints (i.e., a series of elements is used in which each represent only a portion of the virtually larger element, [5, chapter 20.3]). In addition [57] describes a system that adopts TEI's feature structures [5, chapter 18] as a meta-format for representing heterogenous complex markup. The TEI tag set for feature structures supports a method for a general purpose data structure. A feature structure is built up of a `fs` element (feature structure) with an optional `type` attribute containing various instances of `f` elements (feature). Each `f` element bears a `name` attribute containing the feature's name. Possible child elements of the `f` element – apart from other feature structures (`fs`) – can be `binary`, `symbol`, `numeric` or `string` elements, allowing differentiation of the feature's value. Apart from the `string` element, which stores the value as its textual content, each of the named elements use a `value` attribute for this purpose. This simple mechanism can be used as a very general representation system. As an extension, feature and feature-value libraries can be established for re-using feature structure components in different instances. Re-entrant feature structures and Collections (complex feature structures) can be used as well. The connection between the primary data and the feature structure annotation(s) can be established by various linking mechanism described in the TEI guidelines (e.g. standoff techniques or XML ID/IDREF). For a concrete example of use cf. [57].

While most of the before-mentioned approaches target at the representation of multi-dimensional annotation, their usage in validating and analyzing multiple annotation layers is restricted: The non XML-based formats such as TeXMECS, LMNL or XCONCUR lack the support for XML's companion specifications such as XPath, XSLT or XQuery (although development has been started for an API for XCONCUR, [40] and query languages for overlapping markup have been proposed by [27, 23, 24, 2, 3]). Another problem arises by the fact that document grammars for validating overlapping markup structures are in the proposal state only, e.g. the Rabbit/Duck grammars proposed by [43] for GODDAG structures/TexMECS, XCONCUR-CL [39] or Creole (Composable Regular Expressions for Overlapping Languages etc., [50]) an extension to RELAX NG [26] developed in the LMNL community. For these reasons, if validating complex markup is an issue, it is easier to stick with XML-based approaches that can make use of the full scale of XML processing tools.

In the following section we will present the format developed in the Sekimo project for analyzing complex markup, the Sekimo Generic Format (SGF).

11.4.2 Sekimo Generic Format and XStandoff

The Sekimo Generic Format has been developed at Bielefeld University during the second phase of the Sekimo project. It is an XML-based successor of the Prolog fact base format for the storage and analysis of multiple annotated texts described in [55] and [19]. It follows a standoff annotation approach but combines all annotation levels that belong to the same primary data in a single XML instance, following the formal model of a multi-rooted tree. In fact, it is possible to store not only the annotations belonging to a single corpus item but several different corpus entries together with their respective annotation and metadata, the resources used during the annotation process and the document editing history.

The basic set-up of an SGF instance is quite simple: it consists of the primary data (either included in the instance or as a reference to an external file – or even multiple files when dealing with diachronic or multi-modal corpora), the segmentation (in terms of character position when dealing with texts, in terms of time spans or frames when dealing with non-textual primary data) and its annotation layers. In addition, optional metadata can be inserted at various positions and a log can be used for saving the document history (i.e. added, modified or deleted annotation elements).[2]

In contrast to other pivot formats such as TEI's feature structures, PAULA or GrAF, SGF tries to maintain as much of the original annotation format as possible, i.e. the only changes that are made concern the deletion of text nodes and the addition of the `sgf:segment` attribute that links to the corresponding `sgf:segment` element (via XML ID/IDREF) that defines the character span in the primary data storing the textual data part that is annotated by this specific element. A second distinguishing feature is that SGF usually stores all information, i.e. primary data, its segmentation and all respective annotation layers, in a single instance. SGF is flexible enough to allow in addition the use of multiple files or – at the opposite range – the storage of a whole corpus together with the resources used in its creation process in a single file. A graphical overview of an SGF instance is shown in Figure 11.3. As one can see, the original annotation layers (one containing POS annotation, the second a logical document structure) remain intact, including their

[2] The log functionality was primarily designed for the web-based annotation tool *Serengeti* but can be used in every other environment to track the changes that have been made to an SGF instance.

respective element hierarchy and attributes (the latter not shown in the simplified graphical overview).[3]

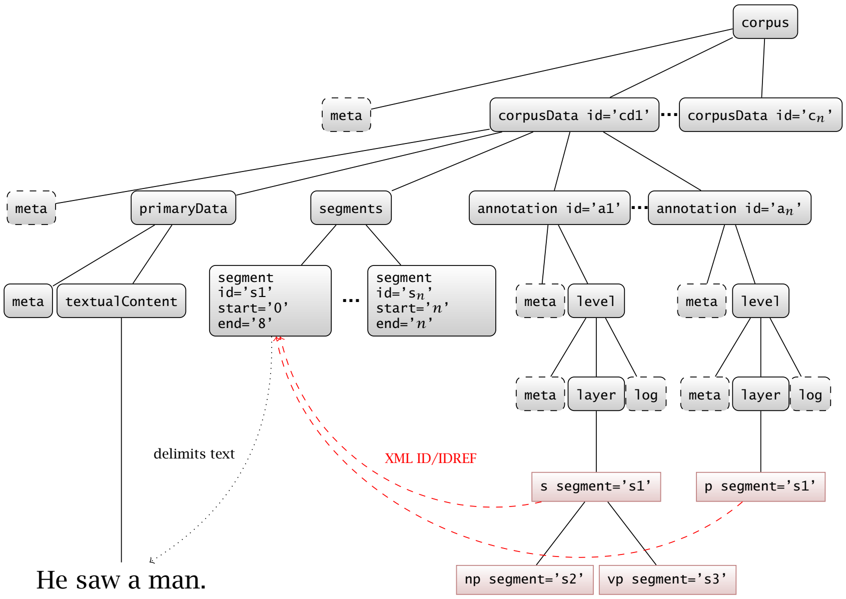


Fig. 11.3 A graphic overview of an SGF instance.

The format as such is designed as a set of XML schema files. In addition, there are converter scripts available as well, allowing the transformation of a single inline annotation into an SGF instance (`inline2SGF`), the merging of SGF annotation levels regarding the very same primary data input (`mergeSGF`), the deletion of SGF annotation levels (`removeLevel`) and a conversion from SGF to inline annotation using TEI milestone elements (`SGF2inline`).

We use SGF for two purposes: first, as a storage and exchange format that can be used in the web-based annotation tool *Serengeti* that has been developed in our project, and second, as a basis for corpus analysis, such as the relationship between elements of the logical document structure layer and anaphoras or antecedents respectively. For the latter it is possible to use standard XML related tools such as XSLT or XQuery to process and query SGF instances. Furthermore, it is possible to extract the reasonable parts from the culminated information stored in an SGF instance that are crucial for a specific task, which is shown in section 11.4.3.

A more detailed description of SGF can be found in [47], for a discussion of its use in the Anaphoric Bank project cf. [36] (in this volume).

The currently developed successor of SGF, called XStandoff (for both *extended* and *extensible* standoff format), introduces some changes to both the format and

[3] A real SGF instance is shown in [36] (in this volume).

the accompanied toolkit (see [48] for a detailed description), including the support for differentiating between containment and dominance relations in XML annotations (see [42] for a discussion) and an all namespace that can be used to subsume elements that are present in different annotation layers, amongst others. As a result, XStandoff is capable of expressing GODDAG structures (including cross-layer validation) while maintaining full compatibility to the XML standard. SGF and XStandoff are available under the GNU Lesser General Public License (LGPL v3)[4].

11.4.3 Antecedent Candidate List

Given the representation formats described in the previous section, the annotation layers for anaphoric relations and logical document structure (see Sections 11.2 and 11.3.3) are converted to SGF and can be analyzed afterwards. For the application domain of anaphora resolution, a set of candidates is identified via an XSLT script for each anaphoric relation and each anaphora together with its candidate set is stored in a candidate list (see Listing 11.1 for a shortened example of a candidate list).

The candidate list consists of several semRel elements each containing one anaphora element and several antecedentCandidate elements. Information on the relation type between the anaphora and its correct antecedent is stored as attribute information in the semRel element. The anaphor element describes properties of the anaphoric element as well as information on the correct antecedent, the antecedentCandidate elements store information on the antecedent candidates. All information is stored in terms of attributes. Congruency information is stored in the attributes num and gen. Additional information is given for part of speech (pos, npType), grammatical function (syntax), dependency structure (dependHead, dependValue) and lemma of head noun (lemma). The position of an element is described as position within the sentence (sentencePos), within the paragraph (paraPos) and in terms of its position regarding the whole document (sentencePosition, position). For all antecedentCandidate elements distance information in terms of sentences and discourse entities is added (sentenceDistance, deDistance). Information on the hierarchical structure of the respective candidates is stored as attribute information c1-docPath. The attribute c1-docAnteHierarchy stores information on the hierarchical relation between anaphora and antecedent candidate, in Listing 11.1 anaphora and correct antecedent are located in the same paragraph, i.e. their discourse entity elements are siblings. For the process of anaphora resolution each anaphora-candidate-pair is interpreted as a feature vector which is used for training a classifier (see also [41, 46, 58]). A detailed description of the candidate list creation process as well as of the XSLT processing script is given in [47].

[4] See http://www.xstandoff.net for downloads, example annotations and further details.

Listing 11.1 Example candidate list. Shortened and manually revised output

```xml
<?xml version="1.0" encoding="UTF-8"?>
<candidateList
  xmlns:sgf="http://www.text-technology.de/sekimo"
  xmlns:chs="http://www.text-technology.de/sekimo/chs"
  xmlns:c1-doc="http://www.text-technology.de/do-gi-docbook"
  <!-- [...] -->
  maxDeDistance="15"
  filename="ling-deu-010-sgf-c1doc.xml">
 <semRel relationID="sr71" type="cospecLink" subtype="ident"
     phorIDRef="de258" antecedentIDRefs="de249">
  <anaphor deID="de258" deType="nom" pos="N" syntax="@NH" lemma="
      monitoring-prozess" dependHead="w932" dependValue="mod"
      npType="pureNP" num="PL" gen="MSC" cas="DAT" sentencePos="
      6/6" paraPos="10/23" sentenceParaPos="2/4" position="241"
      sentencePosition="38" c1-docPath="/article[1]/sect1[3]/para
      [2]">Monitoring-Prozessen</anaphor>
  <!-- [...] -->
  <antecedentCandidate correctAntecedent="yes" deID="de249"
      deType="nom" pos="N" syntax="@NH" lemma="monitoring-prozess
      " dependHead="w914" dependValue="subj" npType="pureNP" num=
      "PL" gen="MSC" cas="NOM" sentencePos="2/4" paraPos="2/23"
      sentenceParaPos="1/4" position="233" sentencePosition="37"
      c1-docPath="/article[1]/sect1[3]/para[2]" deDistance="8"
      sentenceDistance="1" c1-docAnteHierarchy="siblings">
      Monitoring-Prozesse</antecedentCandidate>
  <!-- [...] -->
  <antecedentCandidate deID="de252" deType="nom" pos="N" syntax="
      @NH" lemma="aufmerksamkeits#fokus" dependHead="w910"
      dependValue="mod" npType="defNP" num="SG" gen="MSC" cas="
      DAT" sentencePos="4/4" paraPos="4/23" sentenceParaPos="1/4"
       position="235" sentencePosition="37" c1-docPath="/article
      [1]/sect1[3]/para[2]" deDistance="6" sentenceDistance="1"
      c1-docAnteHierarchy="siblings">m Aufmerksamkeitsfokus</
      antecedentCandidate>
  <!-- [...] -->
 </semRel>
</candidateList>
```

11.5 Results of a Corpus Study

The corpus annotated during the project has a total size of 14 documents, divided into six German scientific articles with complex document structure and eight German newspaper articles. The corpus comprises 3084 sentences with 55221 tokens and 11459 discourse entities. We've annotated 4185 anaphoric relations (3223 direct and 962 indirect).

The corpus under investigation consists of five German scientific articles, its size and information on anaphoric relations are given in Table 11.1. In our analyses we focus on semantic relations with only one antecedent due to the fact that relations with more than one antecedent only play a minor role (see column *#SemRels* $>$ *1*

Ante in Table 11.1). Pronominal anaphoras tend to find their antecedent at a small distance that almost always lies within a distance of 15 discourse entities (DE henceforth) therefore we focus our research questions on non-pronominal anaphoras (see Figure 11.4).

Table 11.1 Overview on the corpus.

Text	#Token	#DE	#Antecedent DE	#Anaphora DE	#Anaphoric relations	#Anaphoric rel. (1 Ante)	#Anaphoric rel. (>1 Ante)
ld-003	12423	2619	996	1347	1358	1311 (96.54%)	47 (3.46%)
ld-010	2248	501	139	174	183	177 (96.72%)	6 (3.28%)
ld-012	6467	1189	342	465	489	484 (98.98%)	5 (1.02%)
ld-014	9385	1529	424	496	500	488 (97.6%)	12 (2.40%)
ld-016	9286	1773	307	395	405	394 (97.28%)	11 (2.72%)
Σ	39809	7611	2208	2877	2935	2854 (97.24%)	81 *(2.76%)*

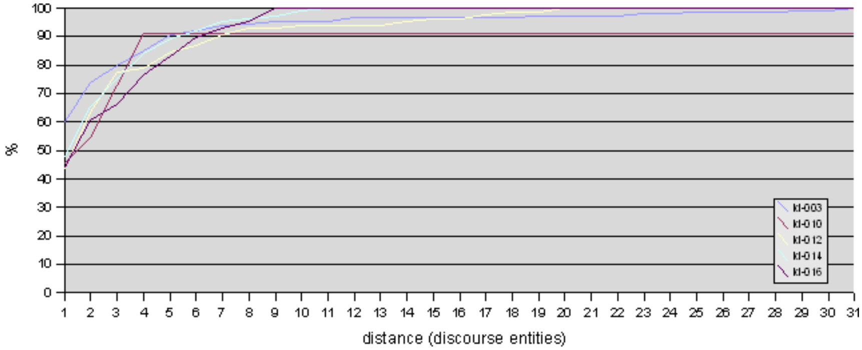


Fig. 11.4 Distance between pronominal anaphora and antecedent in discourse entities.

Distance information for non-pronominal anaphoras shows that linear distance is greater than for pronominal anaphoras (Figure 11.5). Both pronominal and non-pronominal anaphoras show fairly homogeneous behavior among the different texts which supports the assumption that distance information is stable among different texts of the same text type even if text length varies among these texts.

Figure 11.5 shows for a distance baseline of 15 discourse entities that – for the different texts – a minimum of 48.27% and a maximum of 61.81% of all anaphoras find their antecedents within this search window. We will now investigate our research questions:

1. How are anaphora and antecedent located regarding LDS?
2. Does the position of the anaphora/the antecedent regarding LDS give hints for the antecedent choice?

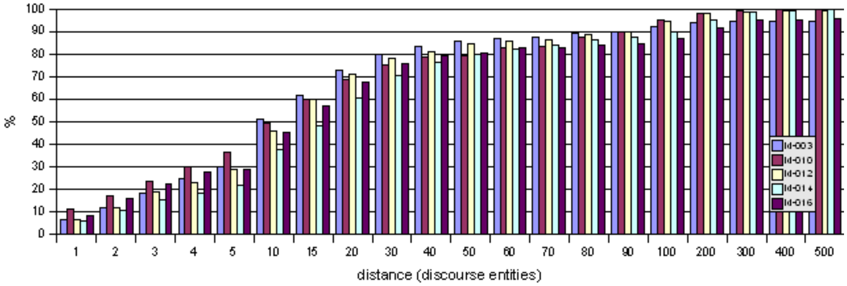


Fig. 11.5 Distance between non-pronominal anaphora and antecedent in discourse entities.

3. Is it possible to define the search window by using information on the position of the anaphora/the antecedent?

Regarding questions (1) and (2) we classify the discourse entities (DEs) occurring in the scientific articles into four categories: (1) DEs that are both in anaphora and in antecedent position, (2) DEs that are in anaphora position only, (3) DEs that are in antecedent position only and (4) DEs that are neither in anaphora nor in antecedent position. We then analyze the attribute `c1-docPath` and extract possible parent elements for each category of DEs. Apart from the fact that anaphoric DEs tend to not occur in title elements (only one of 1532 anaphoric-only DEs occurs in a title element), anaphoric and antecedent elements occur in the same elements of the logical document structure. Thus, sole information on the LDS parent element cannot be used as a constraint on antecedent detection, but might help to identify anaphoric elements. In order to constrain antecedent selection the complete LDS path has to be taken into account (see e.g. LDS-Filter3 below). Previous corpus evidence regarding the choice of DEs located in footnote elements or list items can be confirmed by the corpus under investigation. LDS information cannot be used as a hard constraint in a sense that the occurrence of a DEs in a given LDS structure prohibits the antecedent to be in antecedent position. Nevertheless, LDS can serve as a weak constraint when comparing structures of competing antecedent candidates.

Regarding question (3) we define a search window of 15 discourse entities to be the baseline and we compare different LDS filters against this baseline. Table 11.3 shows the results of the tests for anaphoric relations with non-pronominal anaphora. For the baseline we simply collect all discourse entities that are within a distance of no more than 15 discourse entities. As the annotation scheme used for corpus creation only allows non-pronominal discourse entities in antecedent position the number of candidates in each set might be smaller than 15. Therefore, we define candidate sets of exactly 15 elements by adding candidates to the baseline set. For each of the LDS filters we create candidate sets of exactly 15 elements, too.

The first LDS filter (Table 11.3: LDS-Filter1) selects antecedent candidates according to their values for the attribute `c1-docAnteHierarchy`. This attribute has different values according to the relationship of `c1-docPath`-values of anaphora and antecedent. The value *siblings* is chosen if the DE-elements of anaphora and antecedent have the same parent element and the value *same* describes LDS paths that contain exactly the same types of elements (e.g. /article[1]/sect1[1]/para[10] – /article[1]/sect1[3]/para[3]). The value *ante-ancestor* is chosen if the parent element of the antecedent DE is an ancestor to anaphora DE's parent element (e.g. /article[1]/sect1[1]/para[10] – /article[1]/sect1[1]/para[10]/ emphasis[1]), the value *ana-ancestor* is chosen accordingly. If none of the above values hold, *yes* indicates the anaphora's LDS path to be longer than the ancestor's path, *no* indicates the opposite. LDS-Filter1 chooses candidates with values *sibling, same* and *yes* according to the corpus findings given in Table 11.2: *sibling, same* and *yes* cover most of the anaphoric relations.

Table 11.2 LDS-Hierarchy of anaphora and antecedent.

	ld-003		ld-010		ld-012		ld-014		ld-016	
Total	1311		177		484		488		394	
siblings	751	(57.28%)	83	(46.89%)	277	(57.23%)	278	(56.97%)	255	(64.72%)
same	276	(21.05%)	28	(15.82%)	154	(31.82%)	125	(25.61%)	97	(24.62%)
ante-ancestor	31	(2.36%)	4	(2.26%)	5	(1.03%)	4	(0.82%)	1	(0.25%)
ana-ancestor	10	(0.76%)	1	(0.56%)	1	(0.21%)	10	(2.05%)	10	(2.54%)
yes	168	(12.81%)	40	(22.6%)	37	(7.64%)	39	(7.99%)	22	(5.58%)
no	75	(5.72%)	21	(11.86%)	10	(2.07%)	32	(6.56%)	9	(2.28%)

LDS-Filter2 is the same as LDS-Filter1 but filters only those candidates whose DE-distance value is greater than 15, thus the baseline set remains unfiltered.

LDS-Filter3 keeps the baseline set unfiltered, too. All LDS elements *glosslist*[5] are filtered from the candidate set as these do not occur in antecedent position, candidates with values *sibling, same* and *yes* are chosen afterwards.

Table 11.3 Results of LDS filters.

	ld-003	ld-010	ld-012	ld-014	ld-016
#Anaphoric Relations	1153 (100%)	166 (100%)	380 (100%)	370 (100%)	265 (100%)
	Coverage for different test cases				
(1) Baseline: deDistance≤15	61.67%	59.64%	59.21%	47.57%	56.6%
(2) CL Size=15 (no LDS-Filter)	65.39%	60.24%	62.89%	51.35%	61.51%
(3) CL Size=15 (LDS-Filter1)	61.49%	61.45%	64.21%	52.16%	61.13%
(4) CL Size=15 (LDS-Filter2)	65.13%	60.84%	63.16%	52.43%	62.26%
(5) CL Size=15 (LDS-Filter3)	65.22%	62.05%	63.16%	52.43%	62.26%

[5] This element was introduced to annotate definition and glossary lists (containing glossary items and the respective definition) which may be found in some of the scientific documents. Since anaphoras should not occur between an anaphora in the running text and an antecedent in a glossary definition we can safely apply the filter.

Table 11.3 shows for each scientific article and for each of the test cases the amount of anaphoric relations for which the correct antecedent candidate is contained in the candidate list. The results show that LDS filters do only play a minor role in the creation of an appropriate candidate set. The antecedent candidate set of 15 elements with no LDS filters is only slightly outperformed by the LDS filters. In fact, LDS-Filter1 decreases the amount of correct antecedents found as it filters correct antecedents that would have been found within the search window. LDS-Filter3 filters candidates whose value of the `c1-docPath`-attribute never occur in antecedent position. However, as distance between anaphora and antecedent can be very large, filtering for single candidates does not much improve the coverage. We can draw the conclusion, that LDS as a hard constraint cannot close the gap to full coverage of antecedent candidates in long texts. We argue to enlarge the candidate set to an appropriate size (see Figure 11.5) and to apply weak constraints in order to choose the correct antecedent from the set of candidates, e.g. based on information as given in Table 11.2.

11.6 Conclusion

The research described in this chapter started with the initial assumption that typical language technological tasks would benefit from considering not only textual content but also additional information that quite often is available in digital documents. Unfortunately, however, the results of our investigations do support our initial assumptions only weakly. The minor effects found led us to the conclusion not to use logical document structure as an absolute constraint for the (non-)accessibility of anaphora antecedents but using it only as an additional resource that might improve the task of anaphora resolution slightly. Moreover, we believe that a whole bunch of additional information sources could be taken into account to improve applications of language technology. (see also [32]) To enhance the accessibility of the diverse information types we propose to make this information available together with the text document in a standardised way. XStandoff could be used as an annotation technique that allows doing this in a powerful way.

Acknowledgments

The work presented in this paper is part of the project A2 Sekimo of the distributed Research Group *Text-technological Modelling of Information* funded by the German Research Foundation. We thank the partner project C1 SemDok for providing their annotation layers of logical document structure.

References

[1] Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Data Structures and Algorithms. Addison-Wesley, Reading (1983)

[2] Alink, W., Bhoedjang, R., de Vries, A.P., Boncz, P.A.: Efficient XQuery Support for Stand-Off Annotation. In: Proceedings of the 3rd International Workshop on XQuery Implementation, Experience and Perspectives, in Cooperation with ACM SIGMOD, Chicago, USA (2006)

[3] Alink, W., Jijkoun, V., Ahn, D., de Rijke, M.: Representing and Querying Multi-dimensional Markup for Question Answering. In: Proceedings of the 5th EACL Workshop on NLP and XML (NLPXML 2006): Multi-Dimensional Markup in Natural Language Processing, EACL, Trento (2006)

[4] Bird, S., Liberman, M.: Annotation graphs as a framework for multidimensional linguistic data analysis. In: Proceedings of the Workshop "Towards Standards and Tools for Discourse Tagging", Association for Computational Linguistics, pp. 1–10 (1999)

[5] Burnard, L., Bauman, S. (eds.): TEI P5: Guidelines for Electronic Text Encoding and Interchange. published for the TEI Consortium by Humanities Computing Unit, University of Oxford, Oxford, Providence, Charlottesville, Bergen (2007)

[6] Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML toolkit: data model and query language. Language Resources and Evaluation 39(4), 313–334 (2005)

[7] Clark, H.: Bridging. In: Johnson-Laird, P.N., Wason, P.C. (eds.) Thinking: Readings in Cognitive Science, pp. 411–420. Cambridge University Press, Cambridge (1977)

[8] Cowan, J., Tennison, J., Piez, W.: LMNL update. In: Proceedings of Extreme Markup Languages, Montréal, Québec (2006)

[9] DeRose, S.J.: Markup Overlap: A Review and a Horse. In: Proceedings of Extreme Markup Languages (2004)

[10] DeRose, S.J., Durand, D.G., Mylonas, E., Renear, A.H.: What is text, really? Journal of Computing in Higher Education 1(2), 3–26 (1990)

[11] Diewald, N., Goecke, D., Stührenberg, M., Garbar, A.: Serengeti - webbasierte annotation semantischer relationen. appears in: LDV-Forum GLDV-Journal for Computational Linguistics and language Technology (2009)

[12] Dipper, S.: Xml-based stand-off representation and exploitation of multi-level linguistic annotation. In: Proceedings of Berliner XML Tage 2005 (BXML 2005), Berlin, Deutschland, pp. 39–50 (2005)

[13] Dipper, S., Götze, M., Küssner, U., Stede, M.: Representing and Querying Standoff XML. In: Rehm, G., Witt, A., Lemnitzer, L. (eds.) Datenstrukturen für linguistische Ressourcen und ihre Anwendungen. Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007, pp. 337–346. Gunter Narr Verlag, Tübingen (2007)

[14] Durusau, P., O'Donnel, M.B.: Tabling the overlap discussion. In: Proceedings of Extreme Markup Languages (2004)

[15] Goecke, D., Witt, A.: Exploiting logical document structure for anaphora resolution. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy (2006)

[16] Goecke, D., Stührenberg, M., Holler, A.: Koreferenz, Kospezifikation und Bridging: Annotationsschema. Research group Text-technological Modelling of Information, Universität Bielefeld, Fakultät für Linguistik und Literaturwissenschaft, & Georg-August-Universität Göttingen, Seminar für Deutsche Philologie (2007)

[17] Goecke, D., Stührenberg, M., Wandmacher, T.: A hybrid approach to resolve nominal anaphora. LDV Forum – Zeitschrift für Computerlinguistik und Sprachtechnologie 23(1), 43–58 (2008)

[18] Goecke, D., Stührenberg, M., Witt, A.: Influence of text type and text length on anaphoric annotation. In: ELRA (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

[19] Goecke, D., Lüngen, H., Metzing, D., Stührenberg, M., Witt, A.: Different views on markup. distinguishing levels and layers. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, pp. 1–21. Springer, Heidelberg (2010)

[20] Goldfarb, C.F.: The SGML Handbook. Oxford University Press, Oxford (1991)

[21] Hilbert, M., Schonefeld, O., Witt, A.: Making CONCUR work. In: Proceedings of Extreme Markup Languages (2005)

[22] Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation, 2nd edn. Addison-Wesley, Reading (2000)

[23] Iacob, I.E., Dekhtyar, A.: Processing XML documents with overlapping hierarchies. In: JCDL 2005: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, pp. 409–409. ACM Press, New York (2005)

[24] Iacob, I.E., Dekhtyar, A.: Towards a query language for multihierarchical xml: Revisiting xpath. In: Proceedings of the 8th International Workshop on the Web & Databases (WebDB 2005), Baltimore, Maryland, USA, pp. 49–54 (2005)

[25] Ide, N., Suderman, K.: GrAF: A Graph-based Format for Linguistic Annotations. In: Proceedings of the Linguistic Annotation Workshop, Association for Computational Linguistics, Prague, Czech Republic, pp. 1–8 (2007)

[26] ISO/IEC 19757-2:2003, Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG (ISO/IEC 19757-2). International Standard, International Organization for Standardization, Geneva (2003)

[27] Jagadish, H.V., Lakshmanany, L.V.S., Scannapieco, M., Srivastava, D., Wiwatwattana, N.: Colorful XML: One hierarchy isn't enough. In: Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD 2004), pp. 251–262. ACM Press, New York (2004)

[28] Langer, H., Lüngen, H., Bayerl, P.S.: Text type structure and logical document structure. In: Proceedings of the ACL 2004 Workshop on Discourse Annotation, Barcelona, pp. 49–56 (2004), http://www.uni-giessen.de/germanistik/ascl/dfg-projekt/pdfs/aclws.pdf

[29] Le Maitre, J.: Describing multistructured XML documents by means of delay nodes. In: DocEng 2006: Proceedings of the 2006 ACM symposium on Document engineering, pp. 155–164. ACM Press, New York (2006)

[30] Lenz, E.A., Lüngen, H.: Dokumentation: Annotationsschicht: Logische Dokumentstruktur. Research group Text-technological Modelling of Information, Universität Dortmund, Institut für deutsche Sprache und Literatur, & Justus-Liebig-Universität Gießen, Fachgebiet Angewandte Sprachwissenschaft und Computerlinguistik (2004)

[31] Lobin, H.: Informationsmodellierung in XML und SGML. Springer, Heidelberg (2000)

[32] Metzing, D.: Diskurs-Anaphern. Texttechnologische Informationsmodellierung und benachbarte linguistische Forschungskontexte. In: Marello, C., Hölker, K. (eds.) Dimensionen der Analyse von Texten und Diskursen, LIT Verlag (to appear 2011)

[33] Paraboni, I.: Generating references in hierarchical domains: the case of document deixis. PhD thesis, Information Technology Research Institute, University of Brighton (2003)

[34] Paraboni, I., van Deemter, K., Masthoff, J.: Generating referring expressions: Making referents easy to identify. Computational Linguistics 33(2), 229–254 (2007)

[35] Pianta, E., Bentivogli, L.: Annotating Discontinuous Structures in XML: the Multiword Case. In: Proceedings of LREC 2004 Workshop on "XML-based richly annotated corpora", Lisbon, Portugal, pp. 30–37 (2004)

[36] Poesio, M., Diewald, N., Stührenberg, M., Chamberlain, J., Jettka, D., Goecke, D., Kruschwitz, U.: Markup infrastructure for the anaphoric bank: Supporting web collaboration. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modelling, Learning and Processing of Text-Technological Data Structures. Springer, Berlin (2011)

[37] Power, R., Scott, D., Bouayad-Agha, N.: Document structure. Computational Linguistics 29(2), 211–260 (2003)

[38] Rizzi, R.: Complexity of context-free grammars with exceptions and the inadequacy of grammars as models for xml and sgml. Markup Languages – Theory & Practice 3(1), 107–116 (2001)

[39] Schonefeld, O.: XCONCUR and XCONCUR-CL: A constraint-based approach for the validation of concurrent markup. In: Rehm, G., Witt, A., Lemnitzer, L. (eds.) Datenstrukturen für linguistische Ressourcen und ihre Anwendungen. Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007. Gunter Narr Verlag, Tübingen (2007)

[40] Schonefeld, O.: A simple API for XCONCUR. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec (2008)

[41] Soon, W.M., Lim, D.C.Y., Ng, H.T.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics 27(4), 521–544 (2001)

[42] Sperberg-McQueen, C., Huitfeldt, C.: Markup discontinued discontinuity in texmecs, goddag structures, and rabbit/duck grammars. In: Proceedings of Balisage: The Markup Conference, Balisage Series on Markup Technologies, vol. 1 (2008)

[43] Sperberg-McQueen, C.M.: Rabbit/duck grammars: a validation method for overlapping structures. In: Proceedings of Extreme Markup Languages (2006)

[44] Sperberg-McQueen, C.M.: Representation of overlapping structures. In: Proceedings of Extreme Markup Languages (2007)

[45] Sperberg-McQueen, C.M., Huitfeldt, C.: GODDAG: A data structure for overlapping hierarchies. In: King, P., Munson, E.V. (eds.) PODDP 2000 and DDEP 2000. LNCS, vol. 2023, pp. 139–160. Springer, Heidelberg (2004)

[46] Strube, M., Müller, C.: A machine learning approach to pronoun resolution in spoken dialogue. In: ACL 2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp. 168–175 (2003)

[47] Stührenberg, M., Goecke, D.: SGF – an integrated model for multiple annotations and its application in a linguistic domain. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec (2008)

[48] Stührenberg, M., Jettka, D.: A toolkit for multi-dimensional markup: The development of SGF to XStandoff. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec, Balisage Series on Markup Technologies, vol. 3 (2009)

[49] Tennison, J.: Layered markup and annotation language (LMNL). In: Proceedings of Extreme Markup Languages, Montréal, Québec (2002)

[50] Tennison, J.: Creole: Validating overlapping markup. In: Proceedings of XTech 2007: The Ubiquitous Web Conference, Paris, France (2007)

[51] Thompson, H.S., McKelvie, D.: Hyperlink semantics for standoff markup of read-only documents. In: Proceedings of SGML Europe 1997: The Next Decade –Pushing the Envelope, Barcelona, pp. 227–229 (1997)

[52] Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. Computational Linguistics 26(4), 539–593 (2001)

[53] Walsh, N., Muellner, L.: Doc-Book: The Definitive Guide. O'Reilly, Sebastopol (1999)

[54] Witt, A.: Meaning and interpretation of concurrent markup. In: Proceedings of ALLC-ACH 2002, Joint Conference of the ALLC and ACH, Tübingen (2002)

[55] Witt, A., Goecke, D., Sasaki, F., Lüngen, H.: Unification of XML Documents with Concurrent Markup. Literary and Lingustic Computing 20(1), 103–116 (2005)

[56] Witt, A., Schonefeld, O., Rehm, G., Khoo, J., Evang, K.: On the lossless transformation of single-file, multi-layer annotations into multi-rooted trees. In: Proceedings of Extreme Markup Languages, Montréal, Québec (2007)

[57] Witt, A., Rehm, G., Hinrichs, E., Lehmberg, T., Stegmann, J.: SusTEInability of linguistic resources through feature structures. Literary and Linguistic Computing (2009) (to appear)

[58] Yang, X., Su, J., Zhou, G., Tan, C.L.: Improving pronoun resolution by incorporating coreferential information of candidates. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain (2004)

[59] Zacchiroli PMFVS: Towards the unification of formats for overlapping markup. New Review of Hypermedia and Multimedia 14(1):57–94 (2008)

Part V
Document Structure Learning

Chapter 12
Machine Learning for Document Structure Recognition

Gerhard Paaß and Iuliu Konya

Abstract. The backbone of the information age is digital information which may be searched, accessed, and transferred instantaneously. Therefore the digitization of paper documents is extremely interesting. This chapter describes approaches for document structure recognition detecting the hierarchy of physical components in images of documents, such as pages, paragraphs, and figures, and transforms this into a hierarchy of logical components, such as titles, authors, and sections. This structural information improves readability and is useful for indexing and retrieving information contained in documents. First we present a rule-based system segmenting the document image and estimating the logical role of these zones. It is extensively used for processing newspaper collections showing world-class performance. In the second part we introduce several machine learning approaches exploring large numbers of interrelated features. They can be adapted to geometrical models of the document structure, which may be set up as a linear sequence or a general graph. These advanced models require far more computational resources but show a better performance than simpler alternatives and might be used in future.

12.1 Introduction

In the last years, there has been a rising interest in the easy access of printed material in large-scale projects such as Google Book Search [47] or the Million Book Project [37]. To make this material amenable to browsing and retrieval the logical structure of documents into titles, headings, sections, and thematically coherent parts has to be recognized. To cope with large collections this task has to be performed in an automatic way. The result produced by a document understanding system, given

Gerhard Paaß · Iuliu Konya
Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS),
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
e-mail: {Gerhard.Paass,iuliu.vasile.konya}@iais.fraunhofer.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 221–247.
springerlink.com

a text representation, should be a complete representation of the document's logical structure, ranging from semantically high-level components to the lowest level components.

Document structure recognition can exploit two sources of information. On the one hand the layout of text on the printed page often gives many clues about the relation of different structural units like headings, body text, references, figures, etc. On the other hand the wording and the contents itself can be exploited to recognize the interrelation and semantics of text passages.

Currently there exists a wide range of algorithms specialized for certain parts of document analysis. In large scale applications these approaches have to cope with the vast variety of printed document layouts. A recent comparison is given by [39] showing that no single algorithm is uniformly optimal. As argued by [5], versatility is the key requirement for successful document analysis systems. Even for the same publisher, the layout of its publications changes drastically over time. This is especially visible when dealing with publications spanning over many decades or even centuries. As a general rule, more recently printed documents are also more complex, and the difference between the layouts used by different publishers becomes more pronounced. Thus it is extremely difficult to have algorithms consistently delivering good results over the whole range of documents.

Machine learning approaches are a potential remedy in this situation. Starting form a training set of documents they are able to extract a large number of features relevant for document structure. In contrast to manually built rule systems they are capable to weight these features such that the change of a few features does not lead to a drastic loss of performance.

In this chapter we describe an approach based on minimum spanning trees. It is able to cope with multiple text columns and embedded commercials having a non-Manhattan layout and may be automatically adapted to the different layouts of each publisher. It has been used in large scale newspaper digitization projects. In chapter 3 we discuss some advanced approaches for detecting the structure of text based on the sequence of text objects and layout features. We introduce Conditional Random Fields, which characterize the interrelation of hidden structural states of text and are able to include a large number of dependent features.

12.2 Document Analysis for Large-Scale Processing

Despite intensive research in the area of document analysis, the research community is still far from the desired goal, a general method of processing images belonging to different document classes both accurately and automatically. While geometric layout analysis methods are fairly mature, logical layout analysis research is mainly focused on journal articles. The automatic discovery of logical document structure allows the application of a multitude of electronic document tools, including markup, hyperlinking, hierarchical browsing and component-based retrieval [40]. For this purpose, the usage of machine learning techniques to arrive at a good

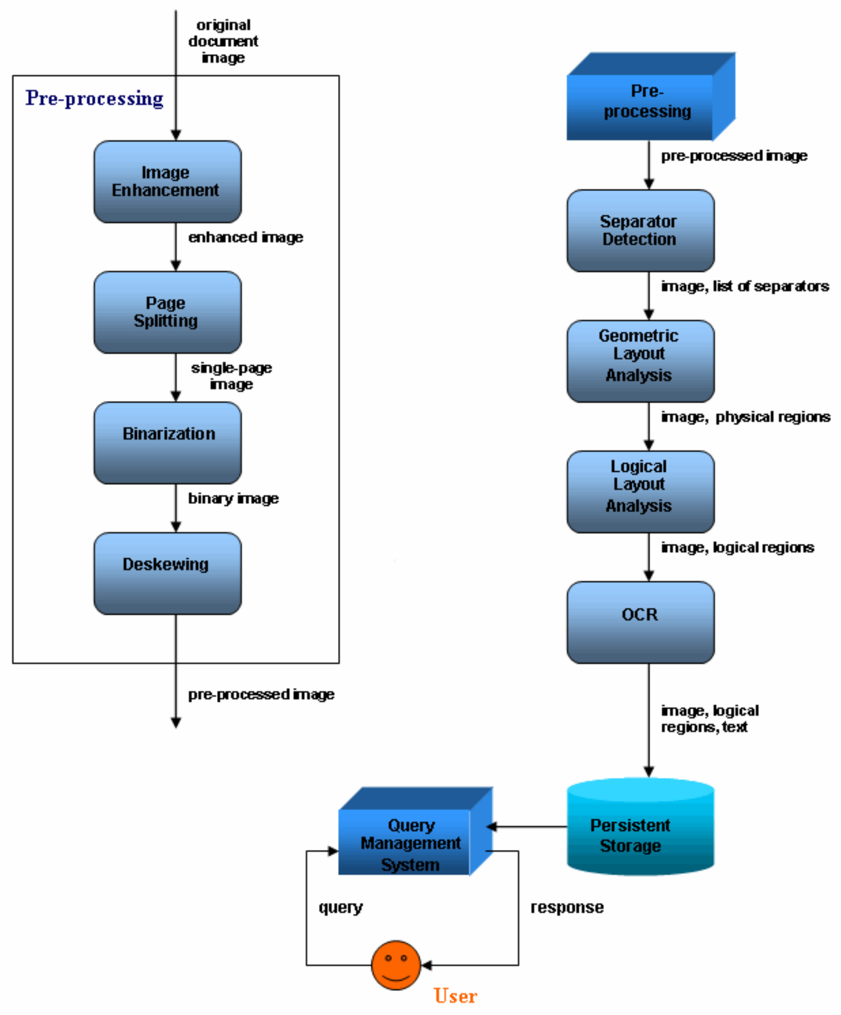


Fig. 12.1 Functional model of a complete, generic *Document Image Understanding* (DIU) system. The ordering of some subsystems may vary, depending on the application area.

solution has been identified by many researchers as being a promising new direction to take [30].

The current section is dedicated to the presentation of a rule-based module for performing logical layout analysis. The described module has been extensively used as part of an automatic system in the processing of large-scale (i.e. >100.000 pages) newspaper collections. As can be seen from Figure 12.1, a generic *Document Image Understanding* (DIU) system must incorporate many specialized modules. Logical layout analysis is considered to be one of the most difficult areas in document processing and is a focal point of current research activity. We will also discuss the applicability of the previously described machine learning approaches as a replacement for the traditional rule-based methods. As a prelude, for the sake of completeness, a brief overview on the current research in geometric layout analysis is presented before going into the state-of-the-art algorithms for logical layout analysis.

12.2.1 Geometric Layout Analysis

The purpose of geometric layout analysis (or page segmentation) is to segment a document image into homogeneous zones, and to categorize each zone into a certain class of *physical layout elements*. Most commonly, the physical layout elements are divided into text, graphics, pictures, tables, horizontal and vertical rulers. It is important to note that in the specialized literature there exists no consensus on the number of physical classes considered, the number depends mostly on the target domain.

Ideally, the page segmentation process should be based solely on the geometric characteristics of the document image, without requiring any a priori information (such as a specific document type – e.g. newspaper, engineering drawing, envelope, web page). Many current page segmentation algorithms are able to meet this condition satisfactorily. In the vast majority of cases, however, the input image is assumed to be noise free, binary, and skew-free.

Traditionally, page segmentation methods are divided into three groups: *top-down* (model-driven), *bottom-up* (data-driven) and *hybrid* approaches. In top-down techniques, documents are recursively divided from entire images to smaller regions. These techniques are generally very fast, but they are only useful when a priori knowledge about the document layout is available. To this class belong methods using projection profiles [21], X-Y cuts [21], or white streams [1]. Bottom-up methods start from pixels, merging them successively into higher-level regions, such as connected components, text lines, and text blocks. These methods are generally more flexible and tolerant to page skew (even multiple skew), but are also slower than top-down methods. Some popular bottom-up techniques make use of region growing [24, 25], run-length smearing [49], or mathematical morphology [19]. Many other methods exist which do not fit exactly into either of these categories; they were consequently called hybrid methods. Hybrid approaches try to combine the high speed of the top-down approaches with the robustness of the bottom-up approaches. Within this category fall all texture-based approaches, such as those employing Gabor filters, multi-scale wavelet analysis [14], or fractal signatures [44].

Many other algorithms for region detection have been proposed in the literature. For a more complete overview one may consult the most recent surveys and methods, such as [9, 29, 39]. Page segmentation methods are being evaluated from time to time, e.g. by [39] who compare the performance of six algorithms and, most recently, in the 2007 ICDAR competition described by [2]. As one may see from the results obtained in recent years, current page segmentation algorithms perform quite well in the task of separating text and non-text regions. An evaluation of the page

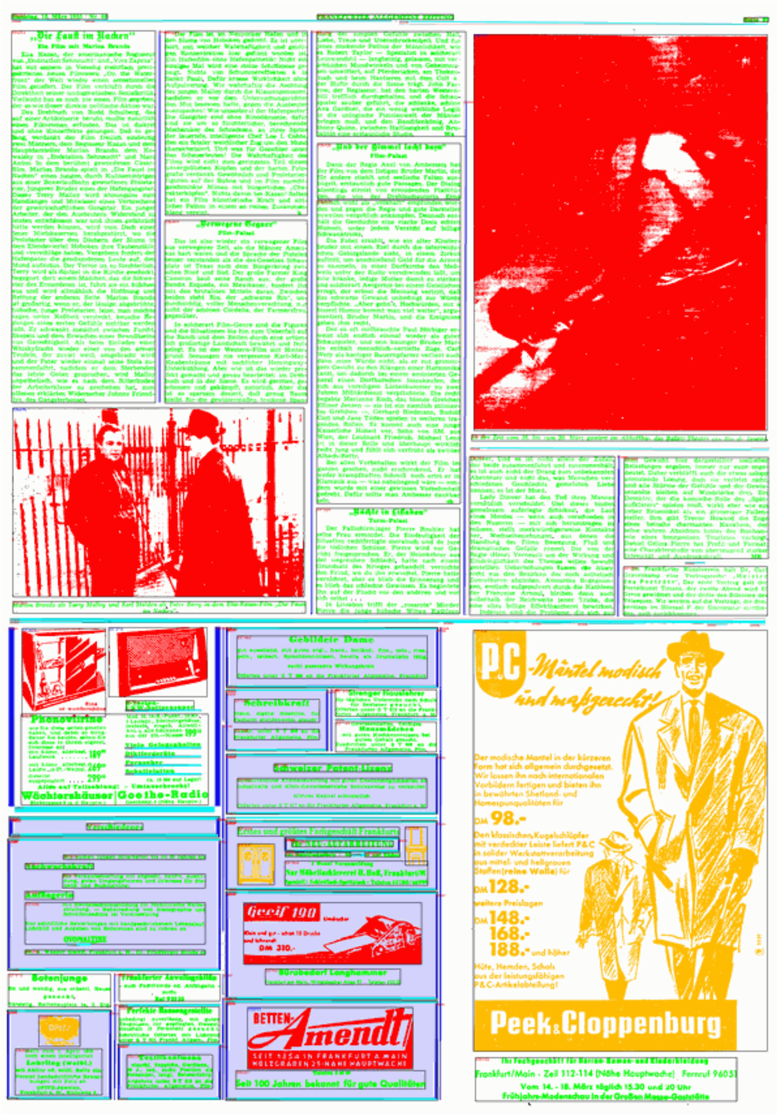
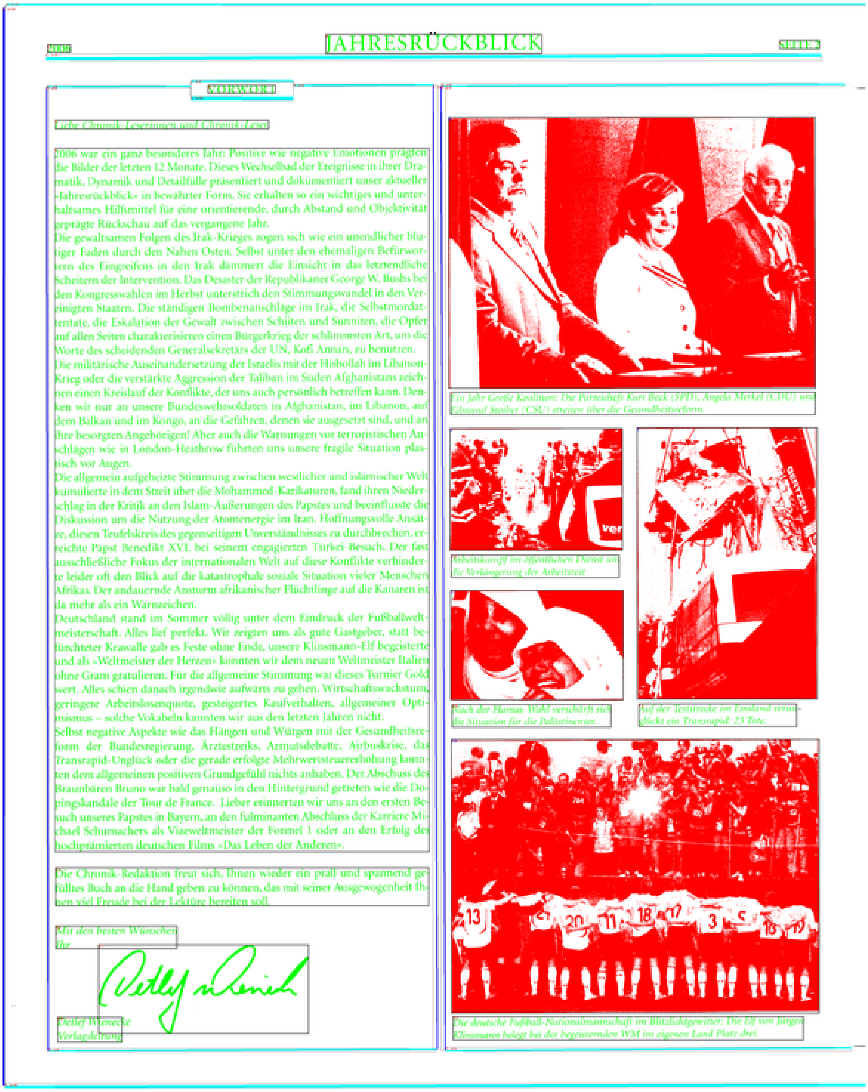


Fig. 12.2 Example of page segmented images from newspaper. Color legend: green= text, red= image, orange= drawing, blue= vertical separator, cyan= horizontal separator, darkened background= frame box

segmentation results produced by the module used in our DIU system on a set of 22 newspaper images coming from 6 different publishers has shown an accuracy of about 95% correctly separated text regions for the text-non-text separation task.



2006

JAHRESRÜCKBLICK

SEITE 2

VORWORT

Liebe Chronik-Leserinnen und Chronik-Leser

2006 war ein ganz besonderes Jahr: Positive wie negative Emotionen prägten die Bilder der letzten 12 Monate. Dieses Wechselbad der Ereignisse in ihrer Dramatik, Dynamik und Detailfülle präsentiert und dokumentiert unser aktueller »Jahresrückblick« in bewährter Form. Sie erhalten so ein wichtiges und unterhaltsames Hilfsmittel für eine orientierende, durch Abstand und Objektivität geprägte Rückschau auf das vergangene Jahr.

Die gewaltsamen Folgen des Irak-Krieges zogen sich wie ein unendlicher blutiger Faden durch den Nahen Osten. Selbst unter den ehemaligen Befürwortern des Eingreifens in den Irak dämmert die Einsicht in das letztendliche Scheitern der Intervention. Das Desaster der Republikaner George W. Bushs bei den Kongresswahlen im Herbst unterstrich den Stimmungswandel in den Vereinigten Staaten. Die ständigen Bombenanschläge im Irak, die Selbstmordattentate, die Eskalation der Gewalt zwischen Schiiten und Sunniten, die Opfer auf allen Seiten charakterisieren einen Bürgerkrieg der schlimmsten Art, um die Worte des scheidenden Generalsekretärs der UN, Kofi Annan, zu benutzen.

Die militärische Auseinandersetzung der Israelis mit der Hisbollah im Libanon-Krieg oder die verstärkte Aggression der Taliban im Süden Afghanistans zeichnen einen Kreislauf der Konflikte, der uns auch persönlich betreffen kann. Denken wir nur an unsere Bundeswehrsoldaten in Afghanistan, im Libanon, auf dem Balkan und im Kongo, an die Gefahren, denen sie ausgesetzt sind, und an ihre besorgten Angehörigen! Aber auch die Warnungen vor terroristischen Anschlägen wie in London-Heathrow führten uns unsere fragile Situation plastisch vor Augen.

Die allgemein aufgeheizte Stimmung zwischen westlicher und islamischer Welt kumulierte in dem Streit über die Mohammed-Karikaturen, fand ihren Niederschlag in der Kritik an den Islam-Äußerungen des Papstes und beeinflusste die Diskussion um die Nutzung der Atomenergie im Iran. Hoffnungsvolle Ansätze, diesen Teufelskreis des gegenseitigen Unverständnisses zu durchbrechen, erreichte Papst Benedikt XVI. bei seinem engagierten Türkei-Besuch. Der fast ausschließliche Fokus der internationalen Welt auf diese Konflikte verhinderte leider oft den Blick auf die katastrophale soziale Situation vieler Menschen Afrikas. Der andauernde Ansturm afrikanischer Flüchtlinge auf die Kanaren ist da mehr als ein Warnzeichen.

Deutschland stand im Sommer völlig unter dem Eindruck der Fußballweltmeisterschaft. Alles lief perfekt. Wir zeigten uns als gute Gastgeber, statt befürchteter Krawalle gab es Feste ohne Ende, unsere Klinsmann-Elf begeisterte und als »Weltmeister der Herzen« konnten wir dem neuen Weltmeister Italien ohne Gram gratulieren. Für die allgemeine Stimmung war dieses Turnier Gold wert. Alles schien danach irgendwie aufwärts zu gehen. Wirtschaftswachstum, geringere Arbeitslosenquote, gesteigertes Kaufverhalten, allgemeiner Optimismus – solche Vokabeln kannten wir aus den letzten Jahren nicht.

Selbst negative Aspekte wie das Hängen und Würgen mit der Gesundheitsreform der Bundesregierung, Ärztestreiks, Armutsdebatte, Airbuskrise, das Transrapid-Unglück oder die gerade erfolgte Mehrwertsteuererhöhung konnten dem allgemeinen positiven Grundgefühl nichts anhaben. Der Abschuss des Braunbären Bruno war bald genauso in den Hintergrund getreten wie die Dopingskandale der Tour de France. Lieber erinnerten wir uns an den ersten Besuch unseres Papstes in Bayern, an den fulminanten Abschluss der Karriere Michael Schumachers als Vizeweltmeister der Formel 1 oder an den Erfolg des hochprämierten deutschen Films »Das Leben der Anderen«.

Die Chronik-Redaktion freut sich, Ihnen wieder ein prall und spannend gefülltes Buch an die Hand geben zu können, das mit seiner Ausgewogenheit Ihnen viel Freude bei der Lektüre bereiten soll.

Mit den besten Wünschen
Ihr

Detlef Wienecke
Verlagsleitung

Ein Jahr Große Koalition: Die Parteichefs Kurt Beck (SPD), Angela Merkel (CDU) und Edmund Stoiber (CSU) streiten über die Gesundheitsreform.

Arbeitskampf im öffentlichen Dienst um die Verlängerung der Arbeitszeit

Nach der Hamas-Wahl verschärft sich die Situation für die Palästinenser.

Auf der Teststrecke im Emsland verunglückt ein Transrapid: 23 Tote.

Die deutsche Fußball-Nationalmannschaft im Blitzlichtgewitter: Die Elf von Jürgen Klinsmann belegt bei der begeisternden WM im eigenen Land Platz drei.

Fig. 12.3 Example of page segmented images from a chronicle. Color legend: green= text, red= image, orange= drawing, blue= vertical separator, cyan= horizontal separator, darkened background= frame box

12.2.2 Logical Layout Analysis

The purpose of logical layout analysis is to segment the physical regions into meaningful *logical units* according to their type (e.g. text lines, paragraphs), assign a *logical label* to each of the determined regions (e.g. title, caption), as well as to determine the *logical relationships* between the logical regions (e.g. reading order, inclusion in the same article). Note that in case the processed document type is a periodical, logical layout analysis is also referred to as *article segmentation*.

The set of available logical labels is different for each type of document. For example: title, abstract, paragraph, section, table, figure and footnote are possible logical objects for technical papers, while: sender, receiver, date, body and signature emerge in letters. Logical relationships are typically represented in a hierarchy of objects, depending on the specific context [9]. Examples of relations are cross references to different parts of an article or the (partial) reading order of some parts of a document. Taking into consideration all these aspects, it becomes clear that logical layout analysis can only be accomplished on the basis of some kind of a priori information (knowledge) about the document class and its typical layout, i.e. a model of the document. Such knowledge can be represented in very different forms (e.g. heuristic rules, formal grammars, probabilistic models such as Hidden Markov Models, a.s.o.). [9] contains a survey of the different document formats used in modern document image understanding systems.

The number of available logical layout analysis algorithms is much lower than that of geometrical layout analysis algorithms, as the difficulty of the task is significantly higher. This section will only present the main ideas of a few methods and the interested reader is advised to consult one of the dedicated survey papers (e.g. [9, 22, 29]).

[46] regarded both the physical layout and logical structure as trees. They transformed the geometrical layout tree into a logical layout tree by using a small set of generic rules suitable for multi-column documents, such as technical journals and newspapers. The physical tree is constructed using block dominating rules. The blocks in the tree are then classified into head and body using rules related to the physical properties of the block. Once this logical tree is obtained, the final logical labels are assigned to the blocks using another set of rules. The logical labels considered are: title, abstract, sub-title, paragraph, header, footer, page number, and caption. A virtual field separator technique is introduced, in which separators and frames are considered as virtual physical blocks in the physical tree. This technique allows the tree transformation algorithm to function with a low number of transformation rules. The authors tested their algorithm on 106 pages from various sources and reported a logical structure recognition accuracy of 88.7%. Errors were due to inaccurate physical segmentation, insufficient transformation rules, and the fact that some pages did not actually have hierarchical physical and/or logical structures.

A general algorithm for automatic derivation of logical document structure from physical layout was described by [40]. The algorithm is divided into segmentation of text into zones and classification of these zones into logical components. The logical document structure is obtained by computing a distance measure between a physical segment and predefined prototypes. The set of properties assigned to each prototype are the parameters from which each distance value is calculated. The properties include contours, context, successor, height, symbols, and children. Basic textual information was also used in order to obtain a higher accuracy. The algorithm was tested on 196 pages from 9 randomly selected computer science technical reports. The labeling result of each text block was characterized as correct, overgeneralized, or incorrect. Two metrics, precise accuracy and generalized accuracy, were used to evaluate the performance. Both average accuracy values were found to be greater than 86%.

[32] presented a system called DeLoS for the derivation of logical document structures. In their system, the algorithm is regarded to be the result of applying a general rule-based control structure as well as a hierarchical multi-level knowledge representation scheme. In this scheme, knowledge about the physical layouts and logical structures of various types of documents is encoded into a knowledge base. The system includes three types of rules: knowledge rules, control rules, and strategy rules. The control rules control the application of knowledge rules, whereas the strategy rules determine the usage of control rules. A document image is first segmented using a bottom-up algorithm, followed by a geometric classification of the obtained regions. Finally, the physical regions are input into the DeLoS system and a logical tree structure is derived. The DeLoS system was tested on 44 newspaper pages. Performance results were reported in terms of block classification accuracy, block grouping accuracy, and read-order extraction accuracy.

In recent years, research on logical layout analysis has shifted away from rigid rule-based methods toward the application of machine learning methods in order to deal with the required versatility. There are several examples for this. [15] employ machine learning in almost every aspect of document analysis, from page segmentation to logical labeling. Their methods are based on inductive learning of knowledge that was hand-coded in previous approaches. [10] use a set of training pages to learn specific layout styles and logical categories. An unknown page is recognized by matching the page's layout tree to the trained models and applying the appropriate zone categories from the best fit layout model. Similarly, the method of [7] finds for a given unlabeled page the best matching layout in a set of labeled example pages. The best match is used to transfer the logical labels to the unlabeled page. The authors see this as a light-weight yet effective approach. [35] use an artificial neural network as the basis of their approach. Instead of a *Multi Layer Perceptron* where the internal state is unknown, they implement a *Transparent Neural Network* that allows for introducing knowledge into the internal layers. The approach features a feedback mechanism by which ambiguous results can be resolved by proposing likely and unlikely results to the input layer based on the knowledge about the current context. The input layer can respond by switching between different feature extraction algorithms in order to determine, for example, the word count in a given block.

In the final Section 12.3 of the paper we describe some machine learning approaches, which capture the meaning of document parts by hidden state variables. A probabilistic model is used to describe the probabilistic relation of these state variables, which may have a rich set of interactions with observable surface features. *Conditional Random Fields* may be used to estimate the parameters of these models from training data. The models have the potential to automatically adapt to novel structures and layouts.

The logical layout analysis methods described so far have not been evaluated rigorously on layouts more complex than journal papers. The very complex newspaper layouts are for example the subject of [18]. This is one of very few publications on the matter of article segmentation. It appears that this reflects the difficulty of the task. Yet, the author realizes that the mass digitalization of newspapers will be one of the next steps after the current wave of book digitalization projects. He proposes a method for learning the layout of different newspapers in an unsupervised manner. In a first stage, a word similarity analysis is performed for each pair of neighboring text blocks. The second stage uses geometric and morphological features of pairs of text blocks to learn the block relations that are characteristic for a specific newspaper layout. Results with high confidence from the word similarity analysis serve as ground truth for the training of the second stage. This method gives promising results and further strengthens the machine learning approach to logical layout analysis.

It is very important to note that in the area of logical layout analysis, there are no standardized benchmarks or evaluation sets, not even algorithms for comparing the results of two different approaches. This is a gap that needs to be filled in future research, as a standardized evaluation is the only way to convincingly demonstrate advances in research on logical layout analysis.

12.2.3 Minimum Spanning Tree-Based Logical Layout Analysis

In case of *newspaper pages* or other *complex layout documents*, the logical layout analysis phase must be able to cope with multiple columns and embedded commercials having a non-Manhattan layout. Most importantly however, the approach has to be flexible enough so as to be readily adaptable (or to adapt automatically) to the different layouts of each publisher. The current section contains the concise description of an article segmentation method, which, based on the construction of a *Minimum Spanning Tree* (MST), is able to handle documents with a great variety of layouts.

Previous approaches using the MST in document layout analysis were proposed by [23] and [13]. [23] construct the MST from the centers of the connected components in the document image; by means of a histogram of slopes of the tree edges, the authors are able to detect the dominant orientation of the text lines. Their method is based on the assumption that inter-character distance is generally lower than inter-line spacing. The algorithm of [13] constructs the MST in a similar way by using the automatically determined inter-character (horizontal) and inter-line (vertical) spacing as splitting thresholds for the tree edges. It produces a segmentation of the document page into text regions as output. As input, the algorithm assumes that the input page is noise-free and contains only text, i.e. all non-text physical regions have been previously removed from the image via specialized filters. The most important problems observed by [13] are the sensitivity of the MST to noisy components and the fact that a single incorrectly split branch can potentially produce a poor segmentation.

The MST-based algorithm introduced in this section requires as input a list containing the bounding boxes of all connected components belonging to text regions as well as the lists of vertical and horizontal separators detected in the given newspaper page. The first step consists of simply grouping the connected components into *text lines*. This can be accomplished by means of many algorithms as, for example, the geometric algorithm proposed in [24]. Several features are computed for each text line, the most important being the stroke width, the x height and the capital letter height of the font, the set of intersected layout columns and its position therein (e.g. left- or right-aligned or centered). Based on these features, one can compute (by minimizing the total merging cost) an optimal set of *text regions* formed by vertically merging adjacent text lines having similar enough characteristics. This step can be accomplished by a dynamic programming approach. The costs of merging two vertically adjacent regions/lines is given by a measure of the similarity between their computed features. Note that here one may also include rules that take into account common formatting conventions, such as indentation at the beginning of each paragraph, in order to prevent text regions from spanning over several paragraphs. A single threshold is needed for this step, namely a stopping criterion for the merging of two regions. The threshold can be determined experimentally, and can subsequently be used for a wide range of publications, as shown by our experience.

At this point, it is possible to compute a *compound distance measure* between any two text regions as a weighted mean of the Euclidean distance between their bounding boxes and a value directly related to the "logical distance" between the two text blocks. The logical distance between two text blocks is asymmetrical and directly influenced by the number and type of separators present between the two text blocks, as well as by their feature similarity (as used for text region creation). The weights assigned to each of these components can and must be adjusted so as to match the different layouts used by a certain newspaper publisher. In order to be able to compute a meaningful logical distance between two blocks, we have additionally performed two steps before it: *detection of titles* and *detection of captions*. By using this additional information (which is in most cases relatively simple given a certain layout), one may compute more accurate logical distances between text blocks. For example a regular text block located before a title block in reading order will have a high logical distance to it (a probable article ending is located between them). A *hierarchy of titles* proved beneficial to be used in our tests, as it allows for the formulation of rules such as: a lower-level title located (in reading order) after a higher-level title with no other title in between has a low logical distance to it (as they are very likely part of the same article). By using the compound distance measure between text blocks, the MST of the document page can be constructed in the next step of the algorithm. It is important to notice that hereby the inherent noise sensitivity of the MST is significantly reduced, due to the usage of higher-order page components (i.e. logical blocks instead of just connected components). Next, the obtained tree is split into a set of smaller trees, each one ideally corresponding to an article. The splitting operation is done by removing the edges which have weights greater than a certain threshold value. The considered threshold value is closely related to the logical distance between blocks, and it should be adjusted according to the layout templates used by each publisher.

Finally, a suitable algorithm for determining the reading-order can be applied separately for each article. The determination of the reading order is a hard task and may depend not only on the geometric layout of a document (which varies widely among publishers even for the same document type), but also on linguistic and semantic content. For all our processing tasks as well as for all experiments described here we used the method proposed by [8], enriched with information regarding the layout columns present in the newspaper image. The method of [8] uses solely geometric information about the text blocks (i.e. bounding boxes, vertical overlaps) and internally performs a *topological sort* given a list of pairs of text regions, each sorted in reading order.

A *post-processing stage* was found to be useful in many cases where an obtained tree actually corresponds to merely a part of an article. This is usually the case when text blocks that do not have an overlain title section are identified as independent articles. Such situations can readily be detected at this stage, followed by a merge at the end of the previous article having a title (if such an article exists). The previous article can be found by searching backward (toward the head) in the list of articles sorted in reading order, where the search seeks the first article which has no horizontal separator between itself and the article under consideration. This procedure has the advantage that it is independent of the language-dependent algorithm used to determine the reading order that was previously employed.

The algorithm of layout analysis described in this section has the advantage of being very fast, robust to noise and easily adaptable to a wide range of document layouts. Its main shortcoming, however, is the need to manually adapt the logical distance measures for each publisher or layout type. Further, the current version of the algorithm does not need to take into account the text within each block, which may be informative in the case of more complex layouts. Thus, machine learning algorithms (such as those described in Section 12.3) for automating these tasks seem to be more promising.

12.2.4 Evaluation

All algorithms described in this section were incorporated in an in-house developed DIU system and were successfully used for segmenting several large ($> 10,000$ pages) newspaper collections. No formal evaluation of the article segmentation results was performed on the respective collections, as a meaningful evaluation can only be performed by humans, which is, of course, prohibitive for such large quantities of data. However, a formal testing of our methods was done on 100 multi-column chronicle pages from the year 2006. Examples of the layout can be observed

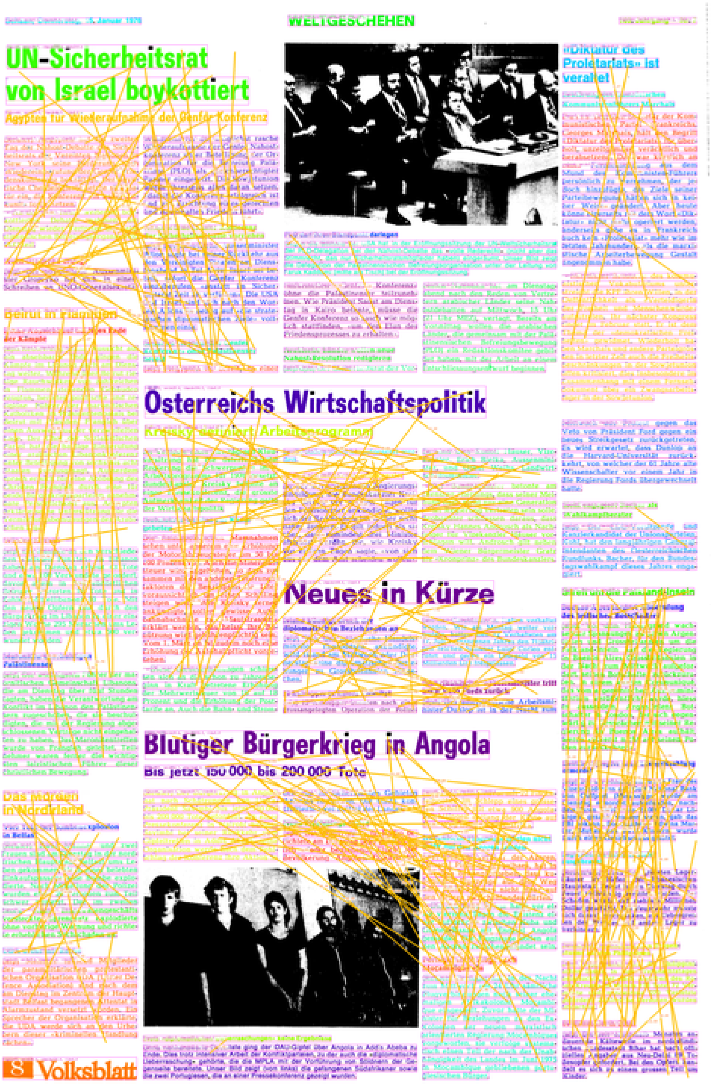

WELTGESCHEHEN
UN-Sicherheitsrat von Israel boykottiert
Ägypten für Wiederaufnahme der Genfer Konferenz
»Diktatur des Proletariats« ist veraltet
Beirut in Flammen
Österreichs Wirtschaftspolitik
Kreisky erläutert Arbeitsprogramm
Neues in Kürze
Blutiger Bürgerkrieg in Angola
Bis jetzt 150 000 bis 200 000 Tote
Das Morden in Nordirland
8 Volksblatt

Fig. 12.4 Example of MST-based article segmentation on newspaper image: initial graph edges.

in the figures 12.2, 12.3, and 12.6. The original input images had 24-bit color depth and a resolution of 400dpi (approx. 4000 × 5000 pixels). Under these conditions, the total processing time for the article segmentation (including text line- and region detection and labeling of titles and captions) on one document image was about 8 seconds on a computer equipped with an Intel Core2Duo 2.66GHz processor and

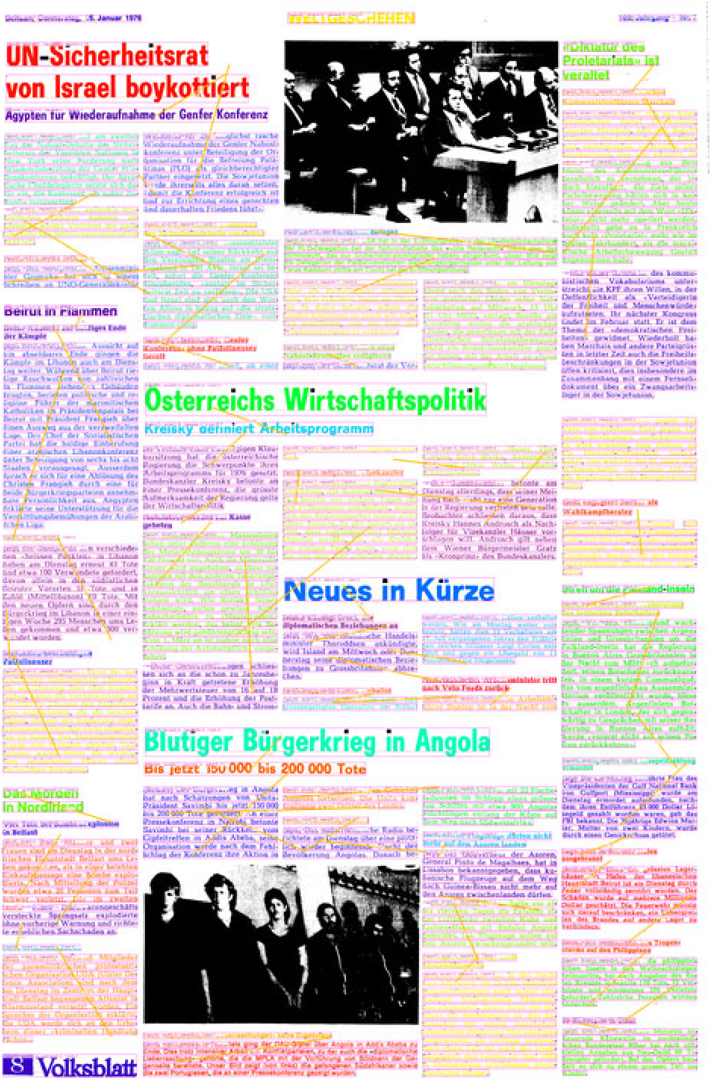

WELTGESCHEHEN
UN-Sicherheitsrat von Israel boykottiert
Ägypten für Wiederaufnahme der Genfer Konferenz
«Diktatur des Proletariats» ist veraltet
Beirut in Flammen
Österreichs Wirtschaftspolitik
Kreisky definiert Arbeitsprogramm
Neues in Kürze
Blutiger Bürgerkrieg in Angola
Bis jetzt 150 000 bis 200 000 Tote
Das Morden in Nordirland
Volksblatt

Fig. 12.5 Example of MST-based article segmentation on a newspaper image: result after the extraction of the Minimum Spanning Tree.

2GB RAM. In the test set there were 621 titles (including subtitles, roof titles and intermediary titles). For the detection and labeling task the manual rule set achieved a precision of 86% and a recall of 96% (resulting in an F-measure of 90.7%). For the detection of captions on the test set containing 255 instances, the rule set was able to achieve an F-measure of 97.6% (precision 98.4% and recall 96.8%). These values

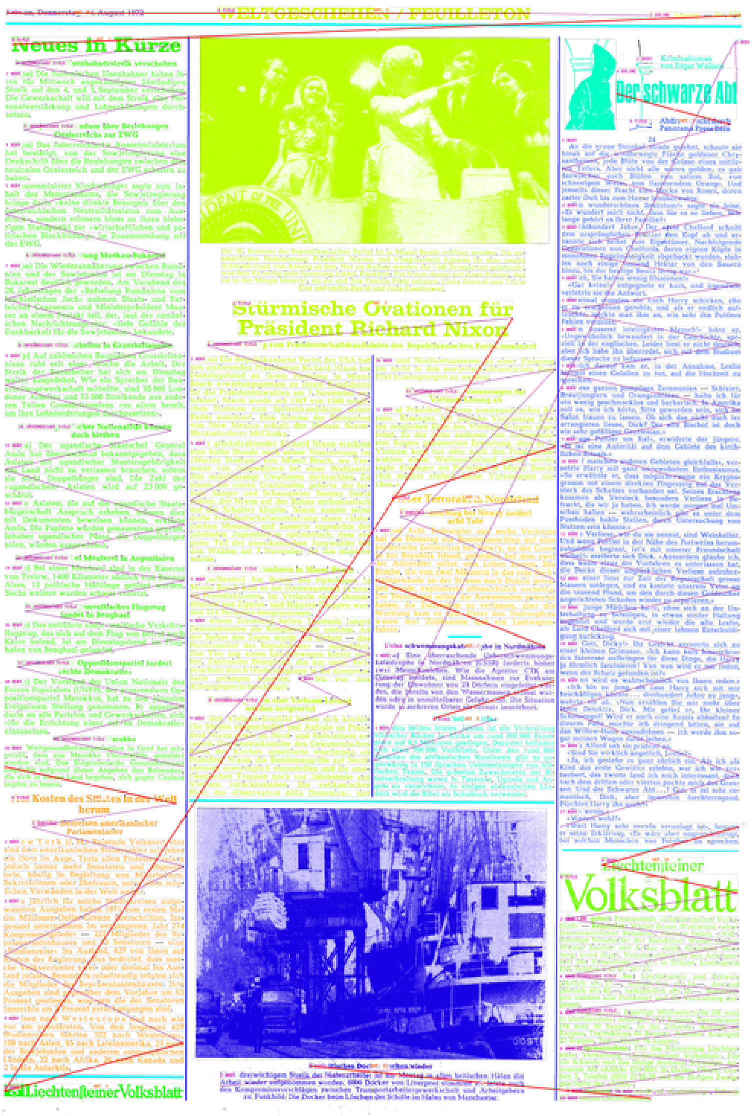
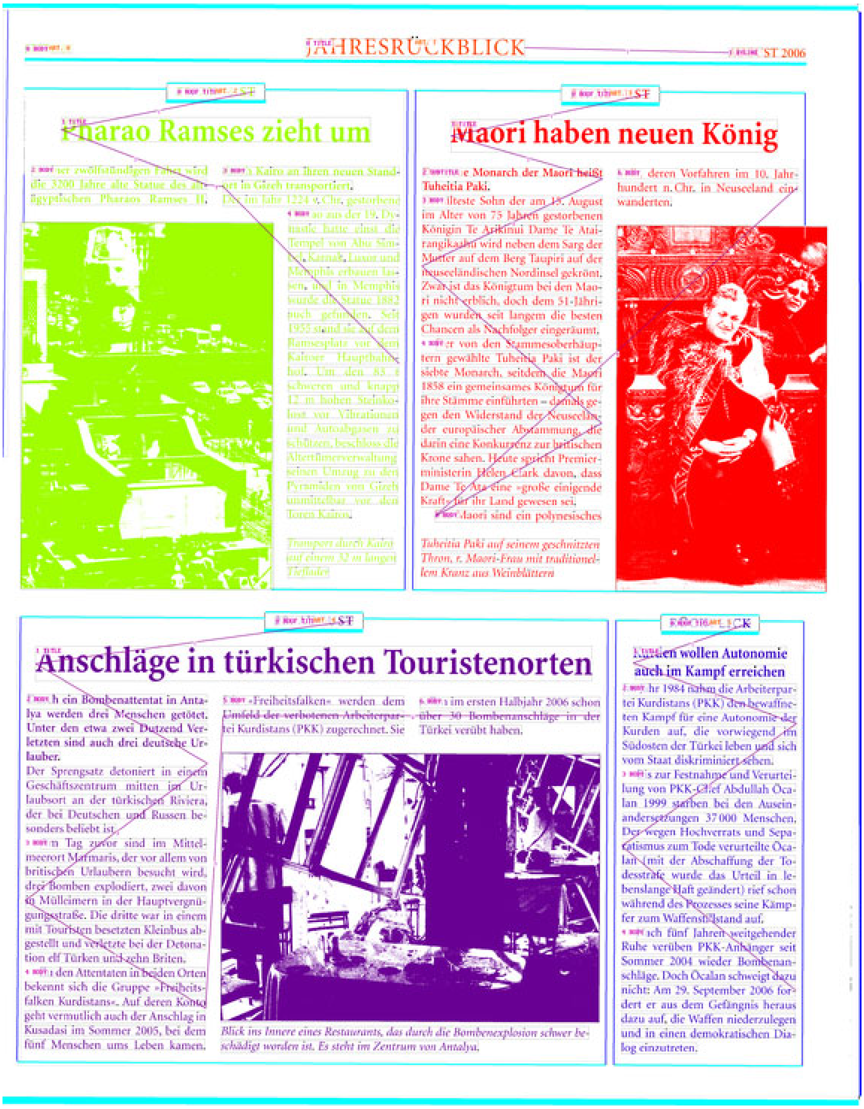

WELTGESCHEHEN / FEUILLETON
Neues in Kürze
Stürmische Ovationen für Präsident Richard Nixon
Der schwarze Abt
Liechtensteiner Volksblatt

Fig. 12.6 Example of article segmented images from a newspaper. Articles have distinct colors and the line segments indicate the detected reading order.

show that a relatively simple rule set is able to perform quite well on known layouts, thus giving hope that in the future such rule sets can be evolved automatically by means of machine learning methods. Based on the results produced by these two manual rule sets, the article segmentation algorithm was able to correctly segment 85.2% of the 311 articles present in the test set. While the vast majority of document



JAHRESRÜCKBLICK ST 2006

ST

Pharao Ramses zieht um

er zwölfstündigen Fahrt wird die 3200 Jahre alte Statue des altägyptischen Pharaos Ramses II. s Kairo an ihren neuen Standort in Gizeh transportiert. Der im Jahr 1224 v.Chr. gestorbene ao aus der 19. Dynastie hatte einst die Tempel von Abu Simbel, Karnak, Luxor und Memphis erbauen lassen, und in Memphis wurde die Statue 1882 auch gefunden. Seit 1955 stand sie auf dem Ramsesplatz vor dem Kairoer Hauptbahnhof. Um den 83 t schweren und knapp 12 m hohen Steinkoloss vor Vibrationen und Autoabgasen zu schützen, beschloss die Altertümerverwaltung seinen Umzug zu den Pyramiden von Gizeh unmittelbar vor den Toren Kairos.

Transport durch Kairo auf einem 32 m langen Tieflader

ST

Maori haben neuen König

e Monarch der Maori heißt Tuheitia Paki.

lteste Sohn der am 15. August im Alter von 75 Jahren gestorbenen Königin Te Arikinui Dame Te Atairangikaahu wird neben dem Sarg der Mutter auf dem Berg Taupiri auf der neuseeländischen Nordinsel gekrönt. Zwar ist das Königtum bei den Maori nicht erblich, doch dem 51-Jährigen wurden seit langem die besten Chancen als Nachfolger eingeräumt.

er von den Stammesoberhäuptern gewählte Tuheitia Paki ist der siebte Monarch, seitdem die Maori 1858 ein gemeinsames Königtum für ihre Stämme einführten – damals gegen den Widerstand der Neuseeländer europäischer Abstammung, die darin eine Konkurrenz zur britischen Krone sahen. Heute spricht Premierministerin Helen Clark davon, dass Dame Te Ata eine »große einigende Kraft« für ihr Land gewesen sei.

aori sind ein polynesisches , deren Vorfahren im 10. Jahrhundert n. Chr. in Neuseeland einwanderten.

Tuheitia Paki auf seinem geschnitzten Thron, r. Maori-Frau mit traditionellem Kranz aus Weinblättern

ST

Anschläge in türkischen Touristenorten

h ein Bombenattentat in Antalya werden drei Menschen getötet. Unter den etwa zwei Dutzend Verletzten sind auch drei deutsche Urlauber.

Der Sprengsatz detoniert in einem Geschäftszentrum mitten im Urlaubsort an der türkischen Riviera, der bei Deutschen und Russen besonders beliebt ist.

n Tag zuvor sind im Mittelmeerort Marmaris, der vor allem von britischen Urlaubern besucht wird, drei Bomben explodiert, zwei davon in Mülleimern in der Hauptvergnügungsstraße. Die dritte war in einem mit Touristen besetzten Kleinbus abgestellt und verletzte bei der Detonation elf Türken und zehn Briten.

i den Attentaten in beiden Orten bekennt sich die Gruppe »Freiheitsfalken Kurdistans«. Auf deren Konto geht vermutlich auch der Anschlag in Kusadasi im Sommer 2005, bei dem fünf Menschen ums Leben kamen.

»Freiheitsfalken« werden dem Umfeld der verbotenen Arbeiterpartei Kurdistans (PKK) zugerechnet. Sie s im ersten Halbjahr 2006 schon über 30 Bombenanschläge in der Türkei verübt haben.

Blick ins Innere eines Restaurants, das durch die Bombenexplosion schwer beschädigt worden ist. Es steht im Zentrum von Antalya.

CK

en wollen Autonomie auch im Kampf erreichen

hr 1984 nahm die Arbeiterpartei Kurdistans (PKK) den bewaffneten Kampf für eine Autonomie der Kurden auf, die vorwiegend im Südosten der Türkei leben und sich vom Staat diskriminiert sehen.

s zur Festnahme und Verurteilung von PKK-Chef Abdullah Öcalan 1999 starben bei den Auseinandersetzungen 37 000 Menschen. Der wegen Hochverrats und Separatismus zum Tode verurteilte Öcalan (mit der Abschaffung der Todesstrafe wurde das Urteil in lebenslange Haft geändert) rief schon während des Prozesses seine Kämpfer zum Waffenstillstand auf.

ch fünf Jahren weitgehender Ruhe verüben PKK-Anhänger seit Sommer 2004 wieder Bombenanschläge. Doch Öcalan schweigt dazu nicht: Am 29. September 2006 fordert er aus dem Gefängnis heraus dazu auf, die Waffen niederzulegen und in einen demokratischen Dialog einzutreten.

Fig. 12.7 Example of a document image (from a chronicle) segmented into articles. Articles have distinct colors and the line segments indicate the detected reading order.

images were segmented correctly, a few pages failed almost totally, thus generating most of the observed errors (e.g. two pages were responsible for more than 75% of the title labeling errors). Article split errors were the most common ones, totaling

13.2%. Most often these errors were generated as a direct consequence of a wrong page segmentation (i.e. split non-text regions such as tables).

12.3 Estimating Document Structure by Conditional Random Fields

Deterministic models for document structure recognition often cannot handle noise or ambiguity. Document pages are usually noisy due to printing, handling, and OCR processes, and this can lead to ambiguous or false results. Geometric structure analysis procedures also have performance uncertainties and so may provide uncertain input to the logical structure analysis process.

In this section we present a number of advanced models to cope with these uncertainties. *Supervised Machine Learning* algorithms are a very general paradigm to train models on annotated training data to reconstruct the most plausible solution from a large number of ambiguous features. In this section we discuss a number of probabilistically oriented machine learning models which address these problems. The relation between input and output is regarded as probabilistic to reflect uncertainty due to erroneous geometric layout analysis results and document noise. These models take into account the geometrical structure of the input documents to derive efficient learning strategies.

12.3.1 Basic Model

Our first model exploits the linear structure of a text for document structure recognition. Let us consider the problem that we want to identify the title and the author in the following text snippet

The new bestseller:
Tears of Love
by Paula Lowe

For reasons of simplicity, we represent the series of all words in this snippet together with line breaks as a single string in which items are coded by a T if they belong to the title, by an A if they belong to the author line, and by an O otherwise. This gives the vectors x of words and y of unknown states:

y	O	O	O	O	T	T	T	O	O	A	A
x	The	new	bestseller	\n	**Tears**	**of**	**Love**	\n	by	Paula	Lowe

Note that text data in documents has two characteristics: first, statistical dependencies exist between the words, we wish to model, and second, each word often has a rich set of features that can aid classification. For example, when identifying the title in a document we can exploit the format and font properties of the title itself, but the location and properties of an author and an abstract in the neighborhood can improve performance.

To infer the unknown states we represent the relation between sequences y and x by a conditional probability distribution $p(y|x)$. More specifically let the variables $y = (y_1,\dots,y_n)$ represent the labels of the word that we wish to predict with a set Y of possible values. Let the input variables $x = (x_1,\dots,x_n)$ represent the observed words and their properties. If $I = \{1,\dots,n\}$ is the set of indices of y then we denote the subvector corresponding to the indices in $A \subset I$ by y_A. Let $\phi_A(x,y_A) > 0$ be a *factor function* with x and the subvectors y_A as arguments and let $\mathscr{C}$ be a set of subsets of $A \subseteq I$. Each $\phi_A(x,y_A)$ is a function taking into account the relation between the labels in the subvector y_A, which often are the adjacent labels in the sequence. Then we represent the conditional distribution by a product of factor functions

$$p(y|x) = \frac{1}{Z(x)} \prod_{A \in \mathscr{C}} \phi_A(x,y_A) \tag{12.1}$$

Here $Z(x) = \sum_y \prod_{A\in\mathscr{C}} \phi_A(x,y_A)$ is a factor normalizing the sum of probabilities to 1.

The product structure enforces a specific dependency structure of the variables $y_i := y_{\{i\}}$. Consider the conditional distribution of y_i given all other variables $y_{D(i)} = (y_1,\dots,y_{i-1},y_{i+1},\dots,y_n)$. It may be written as

$$p(y_i|y_{D(i)},x) = \frac{p(y_i,y_{D(i)},x)}{\sum_{y_i\in Y} p(y_i,y_{D(i)},x)} = \frac{\prod_{B\in\mathscr{C},i\in B} \phi_B(x,y_B)}{\sum_{y_i\in Y} \prod_{B\in\mathscr{C},i\in B} \phi_B(x,y_B)} \tag{12.2}$$

as the factor functions $\phi_A(x,y_A)$ where $i \notin A$ cancel. Therefore the conditional probability of y_i is completely determined if the values of x and the y_B are known for all B which contain i. The factor functions $\phi_A(x,y_A)$ describe the *interactions* between the argument variables. Obviously $\mathscr{C}$ determines the dependency structure of the components of y. A probability distribution of this form is called *Conditional Random Field* (CRF) [27, 42]. As dependencies among the input variables x do not need to be explicitly represented, rich, global input features x may be used. For example, in natural language tasks, useful features include neighboring words and word bigrams, prefixes and suffixes, capitalization, membership in domain-specific lexicons, and semantic information from sources such as WordNet.

Usually there exists a number of different *features* for the same variables x, y_A. For $A = \{i\}$ for instance $\phi_A(x,y_i)$ may cover the feature that word x_i is in bold and $y_i = T$, i.e. is a title word. If we have K_A features for A then we may write $\phi_A(x,y_A) = \exp(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x,y_A))$. Here $\lambda_{A,k}$ is a real-valued *parameter* determining the importance of the real-valued *feature function* $f_{A,k}(x,y_A)$. The exponentiation ensures that the factor functions are positive. This yields the representation

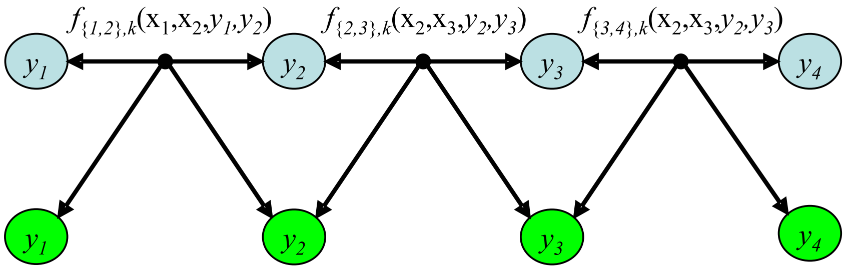

$f_{\{1,2\},k}(\mathrm{x}_1,\mathrm{x}_2,y_1,y_2)$
$f_{\{2,3\},k}(\mathrm{x}_2,\mathrm{x}_3,y_2,y_3)$
$f_{\{3,4\},k}(\mathrm{x}_2,\mathrm{x}_3,y_2,y_3)$
y_1
y_2
y_3
y_4
y_1
y_2
y_3
y_4

Fig. 12.8 A linear conditional random field with four states.

$$\begin{aligned} p(y|x) &= \frac{1}{Z(x)} \prod_{A\in\mathscr{C}} \exp\left(\sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x,y_A)\right) \\ &= \frac{1}{Z(x)} \exp\left(\sum_{A\in\mathscr{C}} \sum_{k=1}^{K_A} \lambda_{A,k} f_{A,k}(x,y_A)\right) \end{aligned} \tag{12.3}$$

Often the feature functions are binary with value $f_{A,k}(x,y_A) = 1$ if the feature is present and $f_{A,k}(x,y_A) = 0$ otherwise. If $\lambda_{A,k} = 0$ the corresponding feature has no influence. For non-negative feature functions positive values for $\lambda_{A,k}$ indicate that the feature increases $p(y_A|x)$, while negative values decrease the conditional probability and have to be estimated from training data by maximum likelihood.

A common special case is the *linear chain conditional random field*, where only interactions between y_t and y_{t-1} are allowed. If in addition we only take into account the corresponding inputs x_t and x_{t-1} the feature functions have the form $f_{\{t-1,t\},k}(x_{t-1},x_t,y_{t-1},y_t)$. Therefore only the adjacent states y_{t-1} and y_t influence each other directly. Figure 12.8 shows such a linear chain with four states. For simplicity reasons only a single type of feature function is shown.

Often it can be assumed, that the parameters do not depend on the particular t and hence $\lambda_{\{t-1,t\},k} = \lambda_{\{t,t+1\},k}$ for all t. This *parameter tying* drastically reduced the number of unknown parameters. More generally we may partition $\mathscr{C} = \{C_1,\ldots,C_Q\}$ where each C_q is a set of all A whose parameters are tied. Then we get the representation

$$p(y|x;\lambda) = \frac{1}{Z(x)} \exp\left(\sum_{C_p\in\mathscr{C}} \sum_{A\in C_p} \sum_{k=1}^{K_A} \lambda_{p,k} f_{A,k}(x,y_A)\right) \tag{12.4}$$

We may estimate the unknown parameters according to the maximum likelihood criterion. Assume that we have observed a number of independent identically distributed observations $(x^{(1)},y^{(1)}),\ldots,(x^{(N)},y^{(N)})$, for example, different documents that are already labeled with the states. Differentiating the log-likelihood function $\ell(\lambda) = \log\prod_n p(y^{(n)}|x^{(n)};\lambda)$ with respect to $\lambda_{p,k}$ yields

$$\frac{\partial\ell(\lambda)}{\partial\lambda_{p,k}} = \sum_{n=1}^{N}\left[\sum_{A\in C_p} f_{A,k}(x^{(n)},y_A^{(n)}) - \sum_{A\in C_p}\sum_{y_A\in Y_A} p(y_A|x^{(n)};\lambda) f_{A,k}(x^{(n)},y_A)\right] \tag{12.5}$$

where Y_A is the set of all possible y_A and $p(y_A|x^{(n)};\lambda)$ is the probability of y_A given $x^{(n)}$ and the current parameter values λ.

The first sum contains the observed feature values for $f_{A,k}(x^{(n)},y_A^{(n)})$ and the second sum consists of the expected feature values given the current parameter λ. If the gradient is zero both terms have to be equal. It can be shown that the log-likelihood function is concave and hence may be efficiently maximized by second-order techniques such as conjugate gradient or L-BFGS [42]. To improve generalization a quadratic penalty term may be added which keeps the parameter values small.

Gradient training requires the computation of the marginal distributions $p(y_A|x^{(i)})$. In the case of a linear chain CRF this can efficiently be done by the forward-backward algorithm requiring $2 \cdot N$ steps. Networks with cycles require more effort as the exact computation grows exponentially with the diameter. An approximate solution is provided by loopy belief propagation (see Section 12.3.4).

If the parameters are known we have to determine the most likely state configuration for a new input $x^+ = (x_1^+,\ldots,x_n^+)$

$$y^* = \arg\max_y p(y|x^+;\lambda) \tag{12.6}$$

which in the case of linear chain models can be efficiently calculated by dynamic programming using the Viterbi algorithm. During prediction the linear-chain CRF takes into account the correlations between adjacent states, which for many problems increase the prediction quality. Other problems requiring long-range correlations between states are described in Section 12.3.3 and Section 12.3.4.

12.3.2 Application of Linear-Chain CRFs to Structure Information Extraction

[34] applied linear chain CRFs to the extraction of structural information from scientific research papers. In their header extraction task they consider the first part of a paper which has to be labeled with the following states: title, author, affiliation, address, note, email, date, abstract, introduction, phone, keywords, web, degree, publication number, and page. A second reference task labels the references at the end of a paper with the following states: *author*, *title*, *editor*, *book title*, *date*, *journal*, *volume*, *tech*, *institution*, *pages*, *location*, *publisher*, *note*. They used the following features:

- *local features describing the current word x_i:* the word itself, whether it starts with a capital letter, whether it contains only capital letters, whether it contains digits, whether it contains only digits, whether it contains a dot, an instance of “-”, an acronym, a capital letter and a dot, whether it matches regular expressions for phone numbers, ZIP codes, URLs or emails;
- *layout features:* the word occurs at the beginning of line, in the middle of line, or at the end of line;

- *external lexicon features:* the word occurs in the list of authors, in the list of dates (e.g. Jan., Feb.), or in notes.

On a training set with 500 headers they achieve an average F1 of 94% for the different fields, compared to 90% for SVMs and 76% for HMMs. For the reference extraction task trained on 500 articles they yield and F1-value of 91.5% compared to 77.6% for an HMM. They found that the Gaussian prior consistently performs best.

[38] uses linear CFRs to extract information like conference names, titles, dates, locations, and submission deadlines from call for papers with the goal to compile conference calenders automatically. He models the sequence of words in a CFP and uses the following layout features: first / last token in the line, first / last line in the text, line contains only blanks / punctuations, line is indented, in first 10 / 20 lines of the text. Using a training dataset of 128 CFPs they achieve an average F1-value of about 60-70% for the title, date and other fields of a CFP. More difficult is the identification of the co-located main conference which has only an F1-value of 35%.

12.3.3 Discriminative Parsing Models

Document structure extraction problems can be solved more effectively by learning a discriminative *Context Free Grammar* (CFG) from training data. According to [48, 4] a grammar has several distinct advantages: long range, even global, constraints can be used to disambiguate entity labels; training data is used more efficiently; and a set of new more powerful features can be introduced. The specific problem [48] consider is of extracting personal contact, or address, information from unstructured sources such as documents and emails.

A CFG consists of a set of terminals $\mathscr{T} = \{w_1,\ldots,w_V\}$ and a set of non-terminals $\mathscr{N} = \{N_1,\ldots,N_n\}$, a designated start symbol N_1 and a set of rules or productions $\mathscr{R} = \{R_i : N_{j_i} \rightarrow \zeta_i\}$ where ζ_i is a sequence of terminals and non-terminals in $\mathscr{N} \cup \mathscr{T}$. A parse tree for a sequence $w_1,\ldots,w_m$ is a tree with $w_1,\ldots,w_m$ as leaf nodes and some $N_i \in \mathscr{N}$ as interior nodes such that the child nodes of an interior node are generated by a rule $R_i \in \mathscr{R}$. Associated with each rule is a score $S(R_i)$. The score of a complete parse tree is the sum of all scores of the rules used in the parse tree. The CKY (Cook-Kasami-Younger) algorithm (see [28]) can compute the parse with the highest score in time $O(n^3 \cdot |\mathscr{R}|)$, which is feasible for relatively small sequence length m.

Assume that a nonterminal N_{j_i} generates the terminals $w_a,\ldots,w_b$. Then the probability of a rule may be written by a log-linear expression

$$p(R_i) = \frac{1}{Z(\lambda(R_i),a,b,R_i)} \exp \sum_{k=1}^{F} \lambda_k(R_i) f_k(w_1,\ldots,w_m,a,b,R_i) \tag{12.7}$$

Here N_{j_i} directly or indirectly generates $w_a,\ldots,w_b$ by R_i. $f_1,\ldots,f_k$ is the set of features similar to the CRF features above, which may depend on all terms in the parenthesis. In principle, these features are more powerful than the linear-chain

CRF-features because they can analyze the sequence of words associated with the current non-terminal and not only for the direct neighboring words. $\lambda_k(R_i)$ is the weight of feature k for R_i and $Z(\cdot)$ is a factors ensuring that the probabilities add up to 1.

As for the CRF this log-linear model is not intended to describe the generative process for $w_1,\ldots,w_m$ but aims at discriminating between different parses of $w_1,\ldots,w_m$. For training [48] use a training set of documents manually labeled with the correct parse tree. They semiautomatically infer a set $\mathscr{R}$ of production rules and a set of features. The weights $\lambda_k(R_i)$ of the features for production rule R_i are determined by the perceptron learning algorithm, which successively increases weights for examples with active features and decreases weights for samples with inactive features.

They apply this approach to a CRF trained by the voted perceptron algorithm. They used a data set with about 1500 contact records with names addresses, etc. for training. For only 27% of the records in the training set an error occurred, while the linear chain CRF had an error rate of 55%. This means that taking into account non-local information by the parse tree approach cut the error in half.

12.3.4 Graph-Structured Model

Up to now we have analyzed document structures with an inherent sequence of elements for the linear chain CRF or discriminative parsing models. We now discuss graph-like structures with more complex dependencies.

As an example consider the problem of processing newspaper archives. After scanning and applying OCR, low-level algorithms may be used to identify elements like lines, paragraphs, images, etc. The 2-dimensional page analysis can go further and establish spatial, logical relations between the elements as, for example, “touch”, “below”, “right of” etc. Especially in newspapers with multi-column layout the sequence of paragraphs of an article in different columns or even on continuation pages is not unique. In the same way the assignment of tables, figures and images located somewhere on the page to an article is a challenging problem.

The low-level analysis generates a number of object o_i, for example, lines, paragraphs, articles, images etc. For some pairs of these object relations r_j may be specified, for example, *image* left-of *article*, *image* belongs-to *article*, *article* below *article*. Each object and each relation has an associated type $t(o_i)$ or $t(r_i)$. Depending on the type each object and each relation is characterized by type-specific attributes as, for example, *topic*, *title*, or *x-y-position*. This yields for each type t a type-specific attribute vector $x_{o_i}^{t(o_i)}$ for an object or and attribute vector $x_{r_i}^{t(r_i)}$ for a relation. Figure 12.9 shows a small example network of relations between articles and images of a newspaper. A *Probabilistic Relational Model* (PRM) [20, 31, 45] represents a joint distribution over the attributes $x_{o_i}^{t(o_i)}$ and $x_{r_i}^{t(r_i)}$ of objects and relations.

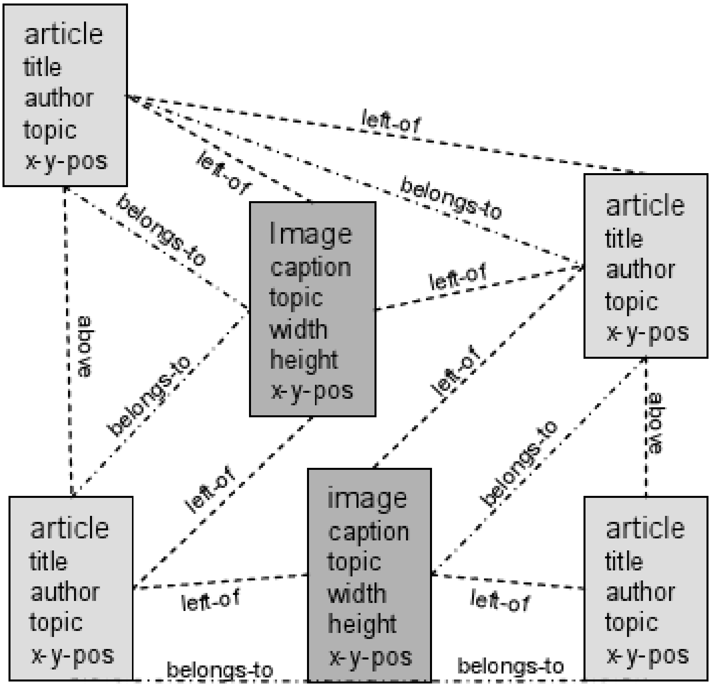


Fig. 12.9 Articles and images of a newspaper page are characterized by a number of attributes. Between a subset of pairs different types of relations exist.

Attributes of an object or relation can depend probabilistically on other attributes of the same or other objects or relations. For example, the probability of an image to belong to an article is higher if it is located close to that article. In the same way, the probability of an image to belong to an article is higher, if the topic of the caption and the topic of the article are similar. These dependencies can be exploited in a probabilistic relation model.

In a linear chain CRF we had a generic dependency template between the states of successive states in the chain. This resulted in using the same parameters independent of the step index or the specific sentence. In the same way probabilistic relational models may define a number of generic dependency templates depending on the types of the involved items. This approach of typing items and tying parameters across items of the same type is an essential component for the efficient learning of PRMs. It enables generalization from a single instance by decomposing the relational graph into multiple examples of each item type (e.g., all image objects), and building a joint model of dependencies between and among attributes of each type.

The resulting probabilistic dependency network is a graph-structured CRF (see Equation 12.4) where parameters are tied in a specific way. This model is discussed in depth by [42]. A number of variants of CRF models have been developed in recent years. Dynamic conditional random fields [43] are sequence models which allow multiple labels at each time step, rather than single labels as in linear-chain CRFs. Lumped label CRFs [33] allow to include observations, where only a subset of labels is observed and it is known that one of the labels in the subset is the true label. Finally, Markov logic networks [36] are a type of probabilistic logic network in which there are parameters for each first-order rule in a knowledge base. These first-order rules may, for example, be exploited to specify constraints between layout elements.

Parameter estimation for general CRFs is essentially the same as for linear chains, except that computing the model expectations requires more general inference algorithms. Whenever the structure of the relationships between elements form an undirected graph, finding exact solutions require special graph transformations and eventually the enumeration of all possible annotations on the graph. This results in the exponential complexity of model training and inference. To make it tractable, several approximation techniques have been proposed for undirected graphs; these include variational and Markov Chain Monte Carlo methods.

A number of alternatives exist:

- Gibbs sampling [17], where for each training example the labels are selected randomly according to the conditional distribution (12.2). The required probabilities can be estimated from the resulting joint distribution of labels.
- Loopy belief propagation [41], performing belief propagation, which is an exact inference algorithm for trees, ignoring part of the links.
- Pseudo-likelihood approaches [6] which instead of the predicted labels use the observed label values to predict a label from its environment.

[11] use a variant of probabilistic relational models to analyze the structure of documents. They aim at annotating lines and pages in layout-oriented documents which correspond to the beginning of sections and section titles. For a local network corresponding to linear chain CRFs, they get an F1-value of 73.4%; it is increased to 79.5% for a graph-structured probabilistic relational network.

There are other, more heuristic models that take into account graph-structured dependencies. [50], for example, consider the problem of sequence labeling and propose a two steps method. First they use a local classifier for the initial assignment of elements without taking into account dependencies. Then a relaxation process successively evaluates non-local dependencies in order to propagate information and to ensure global consistency. They test their approach on a collection of 12,000 course descriptions which have to be annotated with 17 different labels such as *lecturer*, *title*, *start time* or *end time*. Each description contains between 4 and 552 elements to be extracted. For a CRF they report an F1-value of 78.7%, for a probabilistic context free grammar using maximum entropy estimators to estimate probabilities they yield 87.4%, while the relaxation model arrives at an F1-value of 88.1%.

12.4 Conclusion

Electronic documents have many advantages over paper documents, including efficient retrieval and fast transmission. Consequently there has been extensive research on converting paper-based documents into electronic documents. In general, paper documents have an inherent structure partly manifested by layout that greatly improves the comprehension of document content. Document structure recognition aims at detecting the hierarchy of physical components, such as pages, columns, paragraphs, and figures, and transforms this into a hierarchy of logical components, such as titles, authors, and sections. This structural information is useful for indexing and retrieving information contained in documents.

This chapter described modern approaches to document structure recognition. It presented a rule-based system that is extensively used for processing newspaper collections. This system employs geometric layout analysis to segment a document image into homogenous zones and performs a logical layout analysis to determine the logical role of these zones. Our minimum spanning tree-based approach outlined here is described in more detail in [26]. It exhibits state-of-the-art performance and won the ICDAR 2009 Page Segmentation Competition [3].

Despite of extensive research, the current algorithms for document structure recognition are far from being perfect, especially if document layout varies. In order to cope with problems of this sort, we described several machine learning approaches that explore large numbers of interrelated and properly weighted features in order to detect document structures. More specifically, we described *Conditional Random Fields* (CRF) as a general probabilistic algorithm to model the uncertain relation between input and output units. It can be adapted to deal with geometrical models of document structure, which may be set up as a linear sequence or a general graph. As an alternative, we outlined discriminative parsing models which are able to cope with tree-structured documents. These advanced models show a better performance than simpler alternatives, but require far more computational resources.

In the future, traditional approaches to document structure recognition will be enhanced by machine learning approaches, which are able to take into account the underlying publication as a whole. This is especially attractive if many different types of layouts have to be processed as, for example, in large digitization projects like G*oogle books* [47], the *Deutsche Digitale Bibliothek* [12] and *Europeana* [16].

Acknowledgement

The work presented here was partly funded by the *German Federal Ministry of Economy and Technology* (BMWi) under the THESEUS project.

References

[1] Akindele, O., Belaid, A.: Page segmentation by segment tracing. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), pp. 341–344 (1993)

[2] Antonacopoulos, A., Gatos, B., Bridson, D.: ICDAR2007 Page Segmentation Competition. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), vol. 2, pp. 1279–1283. IEEE Computer Society, Los Alamitos (2007)

[3] Antonacopoulos, A., Pletschacher, S., Bridson, D., Papadopoulos, C.: Icdar 2009 page segmentation competition. In: 10th International Conference on Document Analysis and Recognition, ICDAR (2009)

[4] Awasthi, P., Gagrani, A., Ravindran, B.: Image modeling using tree structured conditional random fields. In: IJCAI (2007)

[5] Baird, H., Casey, M.: Towards versatile document analysis systems. In: Proc. 7th International Workshop Document Analysis Systems, pp. 280–290 (2006)

[6] Besag, J.: Statistical analysis of non-lattice data. The Statistician 24(3), 179–195 (1975)

[7] van Beusekom, J., Keysers, D., Shafait, F., Breuel, T.: Example-based logical labeling of document title page images. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), vol. 2, pp. 919–923. IEEE Computer Society, Los Alamitos (2007)

[8] Breuel, T.M.: High performance document layout analysis. In: Symposium on Document Image Understanding Technology, Greenbelt, Maryland (2003)

[9] Cattoni, R., Coianiz, T., Messelodi, S., Modena, C.: Geometric layout analysis techniques for document image understanding: a review. Tech. Rep. 9703-09, ITC-irst (1998), http://citeseer.comp.nus.edu.sg/330609.html

[10] Chen, S., Mao, S., Thoma, G.: Simultaneous layout style and logical entity recognition in a heterogeneous collection of documents. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), vol. 1, pp. 118–122. IEEE Computer Society, Los Alamitos (2007)

[11] Chidlovskii, B., Lecerf, L.: Stacked dependency networks for layout document structuring. In: SAC 2008, pp. 424–428 (2008)

[12] DDB, Deutsche Digitale Bibliothek (2010), http://www.deutsche-digitale-bibliothek.de/ (retrieved on December 23, 2010)

[13] Dias, A.P.: Minimum spanning trees for text segmentation. In: Proc. Annual Symposium Document Analysis and Information Retrieval (1996)

[14] Doermann, D.: Page decomposition and related research. In: Proc. Symp. Document Image Understanding Technology, pp. 39–55 (1995)

[15] Esposito, F., Malerba, D., Semeraro, G., Ferilli, S., Altamura, O., Basile, T., Berardi, M., Ceci, M., Di Mauro, N.: Machine learning methods for automatically processing historical documents: from paper acquisition to xml transformation. In: Proc. 1st International Workshop Document Image Analysis for Libraries, pp. 328–335. IEEE Computer Society, Los Alamitos (2004)

[16] Europeana, Europeana portal (2010), http://www.europeana.eu/ (retrieved on December 23, 2010)

[17] Finkel, J., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: ACL (2005)

[18] Furmaniak, R.: Unsupervised newspaper segmentation using language context. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), vol. 2, pp. 619–623. IEEE Computer Society, Los Alamitos (2007)

[19] Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3686, pp. 609–618. Springer, Heidelberg (2005)

[20] Getoor, L., Taskar, B. (eds.): Introduction to Relational Statistical Learning. MIT Press, Cambridge (2007)

[21] Ha, J., Haralick, R., Phillips, I.: Document page decomposition by the bounding-box projection technique. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), pp. 1119–1122 (1995)

[22] Haralick, R.: Document image understanding: Geometric and logical layout. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 385–390 (1994)

[23] Ittner, D.J., Baird, H.S.: Language-free layout analysis. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), pp. 336–340 (1993)

[24] Jain, A., Yu, B.: Document representation and its application to page decomposition. IEEE Trans on Pattern Analysis and Machine Intelligence 20(3), 294–308 (1998)

[25] Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area voronoi diagram. Computer Vision and Image Understanding 70(3), 370–382 (1998)

[26] Konya, I.V., Seibert, C., Eickeler, S.: Fraunhofer newspaper segmenter – a modular document image understanding system. Journal on Document Analysis and Recognition, IJDAR (2011); Ijdar – expected publication in 2010 (accepted for publication)

[27] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on ine Learning, vol. (2001)

[28] Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge (1999)

[29] Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: A literature survey. In: Document Recognition and Retrieval X, SPIE, vol. 5010, pp. 197–207 (2003)

[30] Marinai, S., Fujisawa, H. (eds.): Machine Learning in Document Analysis and Recognition. Springer, Heidelberg (2008)

[31] Neville, J., Jensen, D.: Relational dependency networks. Journal of Machine Learning Research 8, 653–692 (2007)

[32] Niyogi, D., Srihari, S.: Knowledge-based derivation of document logical structure. In: Proc. Int. Conference on Document Analysis and Recognition, Montreal, Canada, pp. 472–475 (1995)

[33] Paaß, G., Reichartz, F.: Exploiting semantic constraints for estimating supersenses with crfs. In: Proc. SDM 2009 (2009)

[34] Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. In: HLT-NAACL 2004, pp. 329–336 (2004)

[35] Rangoni, Y., Belaïd, A.: Document logical structure analysis based on perceptive cycles. In: Proc. 7th International Workshop Document Analysis Systems, pp. 117–128. Springer, Heidelberg (2006)

[36] Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)

[37] Sankar, K.P., Ambati, V., Pratha, L., Jawahal, C.: Digitizing a million books: Challenges for document analysis. In: Proc. 7th International Workshop Document Analysis Systems, pp. 425–436 (2006)

[38] Schneider, K.M.: Information extraction from calls for papers with conditional random fields and layout features. Artif. Intell. Rev. 25, 67–77 (2006)

[39] Shafait, F., Keysers, D., Breuel, T.: Performance comparison of six algorithms for page segmentation. In: 7th IAPR Workshop on Document Analysis Systems (DAS), pp. 368–379 (2006)

[40] Summers, K.: Near-wordless document structure classification. In: Proc. International Conf. on Document Analysis and Recognition (ICDAR), pp. 462–465 (1995)

[41] Sutton, C., McCallum, A.: Collective segmentation and labeling of distant entities in information extraction. In: ICML Workshop on Statistical Relational Learning (2004)

[42] Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Relational Statistical Learning. MIT Press, Cambridge (2007)

[43] Sutton, C.A., Rohanimanesh, K., McCallum, A.: Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In: Proc. ICML 2004 (2004)

[44] Tang, Y., Ma, H., Mao, X., Liu, D., Suen, C.: A new approach to document analysis based on modified fractal signature. In: Proc. International Conf. Document Analysis and Recognition (ICDAR), pp. 567–570 (1995)

[45] Taskar, B., Abbeel, P., Koller, D.: Discriminative probabilistic models for relational data. In: Eighteenth Conference on Uncertainty in Artificial Intelligence, UAI 2002 (2002)

[46] Tsujimoto, S., Asada, H.: Major components of a complete text reading system. Proc. IEEE 80(7), 1133–1149 (1992)

[47] Vincent, L.: Google book search: Document understanding on a massive scale. In: Proc. 9th International Conf. Document Analysis and Recognition, pp. 819–823 (2007)

[48] Viola, P.A., Narasimhan, M.: Learning to extract information from semi-structured text using a discriminative context free grammar. In: SIGIR 2005, pp. 330–337 (2005)

[49] Wahl, F., Wong, K., Casey, R.: Block segmentation and text extraction in mixed text/image documents. Computer Vision, Graphics, and Image Processing 20, 375–390 (1982)

[50] Wisniewski, G., Gallinari, P.: Relaxation labeling for selecting and exploiting efficiently non-local dependencies in sequence labeling. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 312–323. Springer, Heidelberg (2007)

Chapter 13
Corpus-Based Structure Mapping of XML Document Corpora: A Reinforcement Learning Based Model

Francis Maes, Ludovic Denoyer, and Patrick Gallinari

Abstract. We address the problem of learning to map automatically flat and semi-structured documents onto a mediated target XML schema. This problem is motivated by the recent development of applications for searching and mining semi-structured document sources and corpora. Academic research has mainly dealt with homogeneous collections. In practical applications, data come from multiple heterogeneous sources and mining such collections requires defining a mapping or correspondence between the different document formats. Automating the design of such mappings has rapidly become a key issue for these applications. We propose a machine learning approach to this problem where the mapping is learned from pairs of input and corresponding target documents provided by a user. The mapping process is formalized as a Markov Decision Process, and training is performed through a classical machine learning framework known as Reinforcement Learning. The resulting model is able to cope with complex mappings while keeping a linear complexity. We describe a set of experiments on several corpora representative of different mapping tasks and show that the method is able to learn mappings with a high accuracy on different corpora.

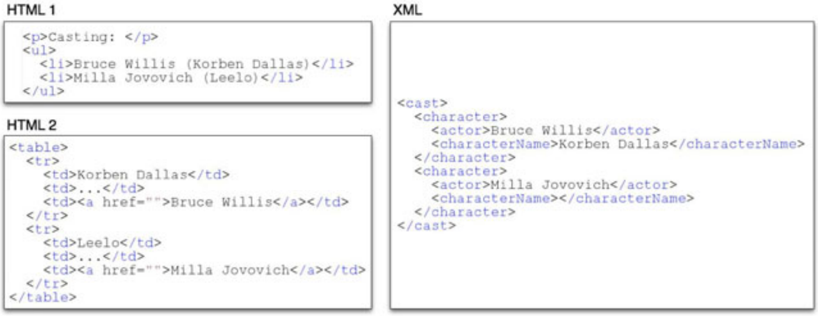
13.1 Introduction and Motivation

The question of heterogeneity is central for semi-structured data: documents often come in many different formats and from heterogeneous sources. Web data sources for example use a large variety of models and syntaxes, as illustrated in Figure 13.1. Although XML has emerged as a standard for encoding semi-structured sources, the syntax and semantic of XML documents following different DTDs or schemas will be different. For managing or accessing an XML collection built from several

Francis Maes · Ludovic Denoyer · Patrick Gallinari
LIP6, 104 Avenue du président Kennedy, 75016, Paris, France
e-mail: {francis.maes,ludovic.denoyer,patrick.gallinari}@lip6.fr

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 249–266.
springerlink.com

sources, a *correspondence* between the different document formats has to be established. Note that in the case of XML collections, the schemas themselves may be known or unknown depending on the source. For HTML data, each site will develop its own presentation and rendering format. Thus even in the case of HTML where the syntax is homogeneous across documents, there is a large variety of formats. Extracting information from different HTML web sites also requires specifying some type of mapping between the specific Web sites formats and the predefined format required by an application.



HTML 1

```
<p>Casting: </p>
<ul>
  <li>Bruce Willis (Korben Dallas)</li>
  <li>Milla Jovovich (Leelo)</li>
</ul>
```

HTML 2

```
<table>
  <tr>
    <td>Korben Dallas</td>
    <td>...</td>
    <td><a href="">Bruce Willis</a></td>
  </tr>
  <tr>
    <td>Leelo</td>
    <td>...</td>
    <td><a href="">Milla Jovovich</a></td>
  </tr>
</table>
```

XML

```
<cast>
  <character>
    <actor>Bruce Willis</actor>
    <characterName>Korben Dallas</characterName>
  </character>
  <character>
    <actor>Milla Jovovich</actor>
    <characterName></characterName>
  </character>
</cast>
```

Fig. 13.1 Example of web page wrapping. This is a case of XHTML to XML Structure Mapping. Nodes can be relabeled, inserted, suppressed and moved.

Designing tree transformations, in order to define correspondences between the different schemas or formats of different sources is thus a key problem to develop applications exploiting and accessing semi-structured sources. This problem has been addressed for some times by the database and to a lesser extent by the document communities for different conversion tasks and settings. Anyway, the real world solution is to perform a manual correspondence between heterogeneous schemas or towards a mediated schema via structured document transformation languages, like XSLT. Although special tools have been developed for helping programmers at this task, the process remains complex; it requires expert skills and the resulting mappings are often very complicated. This is not adapted to situations where document sources are multiple and change frequently. Furthermore, languages such as XSLT are limited by two important properties of real world document collections:

- The schema for large document collections are often very loose and impose only few constraints on valid documents[1]. In this case, the schema itself does not provide enough information and writing an XSLT script for document transformations can be very difficult and time consuming.
- Many sources, especially on the Web, come without schema and the only evidence comes from the document itself. The transformation problem has then to

[1] This is the case for example for the Wikipedia [6] or the IEEE INEX XML collections.

be studied from a *document centric* perspective as opposed to the *data centric view developed for databases*. This means that the semantic of the document is important and that transformations shall take into account both the textual content and the structure of the document. Here, the order of the document leaves and nodes is meaningful and shall be taken into account by the transformation.

Automating the design of these transformations has rapidly become a challenge. Several approaches have been explored ranging from syntactic methods based on grammar transformations or tree transducers to statistical techniques. However, these methods are still hard to use in practical cases. Many of them heavily rely on task specific heuristics. Current approaches to document transformation are usually limited to one transformation task or to one type of data. A majority of techniques only consider the structural (logical or syntactic) document information and do not exploit content nodes. Even this structural information is used in a limited way and most methods exploit only a few structural relationships. Besides, most proposed techniques do not scale to large collections.

We consider here the problem of *Structure Mapping* which consists in automatically learning to map semi structured documents coming from heterogeneous sources onto a predefined mediated schema. This transformation problem is studied from a **document centric** perspective as opposed to the **data centric view developed for databases**. This means that the semantic of the document is important and that mappings shall use both the textual content information and the structure of the document. Let us motivate this mapping task. The structure mapping task encompasses a large variety of real-world applications like:

- **Semantic Web** conversion from raw HTML to semantically enriched XML. Example sources include forums, blogs, wiki-based sites, domain specific sites (music, movies, houses, ...).
- **Wrapping of Web pages** conversion from pages coming from many sources to a unified format.
- **Legacy Document conversion** The document-engineering field were document mapping has been at the heart of many applications like document annotation or document conversion of flat text, loosely structured text, HTML or PDF formats onto a predefined XML schema [4, 3].
- **Heterogeneous Search** Tree transformation is also relevant to the field of Information Retrieval. When searching in XML collections, targets are no more documents but document elements. Queries may address either the content of the document, or both its content and structure. In both cases, the goal will be to retrieve the most specific relevant elements in a document. The INEX initiative [12] launched in 2002 has focused on the development of XML search engines. The initial collections at INEX were homogeneous, all documents sharing a common DTD. Recently a heterogeneous search track has been launched where the goal is to query collections coming from different sources and with different formats.

The paper is organized as follow: We propose a related work in Section 13.2. In part 13.3 we describe the general Structure Mapping task. We introduce the model in part 13.4 and then present experiments on five different corpora in Section 13.5. The related work is presented in Section 13.2.

13.2 Related Work

Three domains have been mainly concerned up to now with the document conversion problem : Information Retrieval, Document Engineering and Databases. We briefly review below related work in these three domains.

Structured document transformation is a relatively new topic in the document community. Automating XML document transformation from one source format onto a target schema is a key issue for document reuse and several approaches have been recently proposed. Work in this area only consider schema transformations and thus requires that the input schema is provided. Content information is completely ignored. This class of methods is thus restricted to collections with well-defined schema and little content like bibliographic data. It does not apply to large document collections. Leinone and al. in [15] for example propose a syntax directed approach based on finite state tree transducers. The system automatically generates mappings when the user specifies a correspondence between document leaves. Su et al. in [21] propose a tree-matching algorithm, which decomposes the mapping into a sequence of basic operation. The algorithm relies on heuristics for exploring the “action” space and on user provided semantic information. Boukotaya et al. in [1] also propose a heuristic algorithm, which combines multiple criteria (semantic relationships between label names, data types compatibility, path similarity, etc). Another interesting topic in the document and web communities is document annotation, which is the transformation of rendering formats like HTML or PDF formats onto a target XML schema. Annotation is a special case of structure mapping where for example the input and output documents have similar leave sequences. Yip Chung et al. in [4] consider an HTML to XML conversion problem. They use unsupervised machine learning and manually defined rules to discover tree patterns in the input document and to produce the output structure. Closest to us is the work by Chidlovskii and colleagues [3] which has been used here as a baseline model for comparison. They also consider the conversion of HTML or PDF documents to a target XML schema. They use classifiers and probabilistic grammars to label input document elements and build output trees. The complexity of these methods however limits their application to small collections (e.g. Shakespeare corpus in the tests presented here).

It is worth mentioning here the work of Collins et al. [5] who recently proposed an incremental method based on machine learning for parsing. This method, which achieved impressive results, also uses a sequence of actions quite similar to ours for building the parse tree from an input sentence but it used for natural language processing and cannot be easily adapted to our structure mapping task.

In the database community automatic or semi-automatic data integration – known as *schema matching* – has been a major concern for many years and there is a profusion of work and topics in this area. Surveys of these techniques can be found in [19] and [20]. We briefly review below three groups of matching models.

- Schema-based models: they transform schemas and ignore the document content [18, 8, 2].
- Instance-level approaches: they consider both the schema and the meaning of the schema elements. They are used when the schema information is limited [9, 16]. Some of these models use basic machine learning techniques like rule learners, neural networks, ...
- Multiple-Matchers approaches: they use several type of matchers in order to compute the matching of two schemas [9, 11]. Some approaches in this field have explored the inference of mapping using machine learning techniques. A remarkable series of work has been developed in this direction by Halevy, Doan and colleagues. In [10], Doan et al. propose a methodology which combines several sources of evidence in the documents (tag names, schema description, data types, local structure, etc), using a regression function learned from a dataset. This method has been developed for different matching problems, and has been used in Section 13.5.4 for comparison on the *RealEstate* corpus.

Past work on schema matching has mostly been done in the context of a particular application domain. As far as we know, there is no benchmark in this domain and no comparison of the different existing approaches.

13.3 Structure Mapping Task

13.3.1 Task Description

We consider here a generic structure mapping task which covers a large range of structure mapping problems like document conversion, document annotation, etc. We define the mapping problem as the task of learning document transformations from heterogeneous formats onto a pre-defined format. Input documents may come from different sources and may take different formats like flat text, wikitext, HTML or XML. Output documents are expressed in a predefined XML schema. The mapping itself may involve several types of elementary actions on the document elements: node labeling, node creation or suppression, merging of nodes, node displacement. The only restriction is that no input leaf will be split during the transformation[2] which is a reasonable assumption for most applications. Of course, the transformation should be consistent so that the output document is a valid XML document according to the target schema.

Figure 13.1 illustrates an example of XHMTL to XML conversion.

[2] This transformation can be handled by our model but it increases its complexity

13.3.2 Notations

Structure mapping consists in transforming an input document $d \in D_{in}$ into a target XML document $d^* \in D_{out}$ where D_{in} is the set of possible input documents and D_{out} is the set of possible output documents. For example in the case of structure mapping on Web sources, D_{in} may be the set of all valid HTML documents. In our setting, the mapping will be performed via a mapping function $f_\theta : D_{in} \rightarrow D_{out}$ with parameters θ that transforms input documents into target documents. There will be two problems to be solved:

- *Inference problem*: given f_θ and an input document d, compute the target document d^*
- *Learning problem*: learn the parameters θ of the mapping function f_θ from a training set of examples.

As training data, a user provides a set of document pairs $\{(d_i, d_i^*)\}_{i \in [1,N]} \in D_{in} \times D_{out}$ where d_i is an input document, d_i^* is the corresponding output document and N is the number of training examples.

13.3.3 Complexity of Inferring and Learning Mappings

Inferring a target structure from an input document is a complex task which involves a search process. The f_θ function will typically evaluate different potential solutions and output the best one according to some scoring function. For a general mapping task, the size of D_{out} – the set of potential solutions – is exponential *w.r.t.* the number of nodes of input documents. In most cases, search amounts at solving a combinatorial problem. This is the reason why most works in this domain only considers limited classes of mappings. For annotation for example the leaf sequence is left unchanged by the mapping which reduces the size of the search space. Even in this limited setting, the inference of document trees rapidly becomes intractable. Existing solutions like [3] rely on dynamic programming for trees and their complexity is:

$$O((\text{number of leaves in the input document})^3 \times C) \tag{13.1}$$

where C is a constant that measures the complexity of the output schema[3]. This complexity is prohibitive even for moderate size documents.

We adopt here a different approach to the inference problem which will allow the complexity to remain linear *w.r.t.* the number of leaves of the input document. The transformed document will be inferred from the input document through a sequence of elementary actions. The global mapping problem is decomposed into a sequence of sub problems which will allow us to break the inherent complexity of the global

[3] The number of leaves of a document is usually much larger than the number of possible output node labels.

problem. Training amounts at learning how to explore the search space in order to perform efficient and reliable inference. Inference is modeled using the *Deterministic Markov Decision Process* framework and training is framed in a classical machine learning framework known as *Reinforcement Learning* [22].

13.4 Reinforcement Learning Based Model

13.4.1 Informal Description

Since this topic might not be familiar to a *Knowledge Management* audience we will provide here a brief introduction to the key ideas of the domain. Given an input document, an output document is build in an incremental process using a sequence of elementary *actions*. Figure 13.2 illustrates the inference step. The leaves of the input document are considered sequentially one after the other and for each leaf, a series of action will be chosen. Initially, we start with a void output document (node labeled HTML on the left of the figure). The first input leaf is then selected and different possible actions for inserting this leaf in the output document are considered. This is illustrated in the second column of the figure. For example, this first leaf may be inserted as a new section (top column box) or as a title (box below), etc. The set of specific actions used in our model will be defined in Section 13.4.2. After this first step, the process is reiterated by selecting the second leaf and by considering potential actions for inserting this leaf in the partial tree built in the first step. The process is then iterated for all leaf nodes. When the last node of the input document has been examined, a valid output documents has been constructed (last column). A partially built output document is called a *state*, transformation from a current state to a new state is called a *state transition*. The graph whose vertices are states and edges transitions is the *state graph*. Since there are usually many valid documents and many valid actions for each state, the graph grows rapidly, hence the combinatorial nature of the problem.

- **Inference** consists in exploring the state space in order to build the best output document. With our model, only a small part of the whole state space will be explored: at each step the best action will be selected according to a decision function called a *policy*. In this way, only one output documents will be selected as the computed output.
- **Training** amounts at learning the policy in order to explore efficiently the search space. This policy will compute a probability for each action in a given state and a subset of actions with high probability will be selected for exploring and growing the state graph. Suppose that the complete state graph was built. We could use a score function to measure the similarity of each output document with the target document. From this score the policy could be learned. Exploring the whole graph is however unfeasible: at each state, one must evaluate the *expectation* of the loss when taking this specific decision. This loss can be seen as a prediction of the quality of the final solution obtained by following this action in the graph. To do that, a *reward* will be associated to each action, and the

Table 13.1 Some examples of $\phi(s,a)$ features which jointly describe the current state s and the action a. These features are usually binary features (in $\{0,1\}$) but real valued features are also possible. The features are generated in a data-driven way: a feature is considered only once it is observed in the learning data.

The second word of the input node is made of lower case letters and we are creating a node with label `LINE`.
The first word of the input node is made of one upper case letter followed by lower case letters and we are creating a node with label `ISBN`.
The first word of the input node is made of one lower case letter followed by digits letters and the node we are creating follows a `SECTION` node.
The first word is a symbol and we are creating a node which follows a `TITLE`.
The input node has tag `H3` and we are skipping it.
...

learning algorithm will learn to optimize the cumulative reward associated to the best path. The algorithms for this kind of sequential learning problem are known as *Reinforcement Learning* [22] and their general formulation is illustrated in Figure 13.3.

We provide in the next section a more formal description of this process, however, the main ideas have been introduced and the reader may skip the formal Section 13.4.2.

13.4.2 Deterministic Markov Decision Process

A classical mathematical setting for modeling sequential decision making is provided by Markov Decision Processes (MDP) [14]. We will consider here Deterministic MDP.

A DMDP is a tuple $(S,A,\delta(.,.),r(.,.))$ where S is the state space, A is the action space, $\delta{:}S\times A\rightarrow S$ is the transition function and $r{:}S\rightarrow\Re$ is a reward function. At any time the process is in a given state $s\in S$. In each state s, several actions in A can be chosen by the model. For a state s and an action a, the transition function δ defines the next state $\delta(s,a)$. The model earns a reward $r(s,a)$ when choosing action a in a visited state s. Decisions are taken according to a *Policy* π_θ where the parameters θ are learned from examples. $\pi_\theta : S\times A\rightarrow[0,1]$ is a function that maps states to action probabilities. $\pi_\theta(s,a)\in[0,1]$ thus denotes the probability to select action a in state s. Learning algorithms attempt to estimate the parameters θ of the policy that maximize a cumulative reward over the course of the problem. The cumulative reward is simply the sum of all rewards over all the states visited from an initial state to a final state.

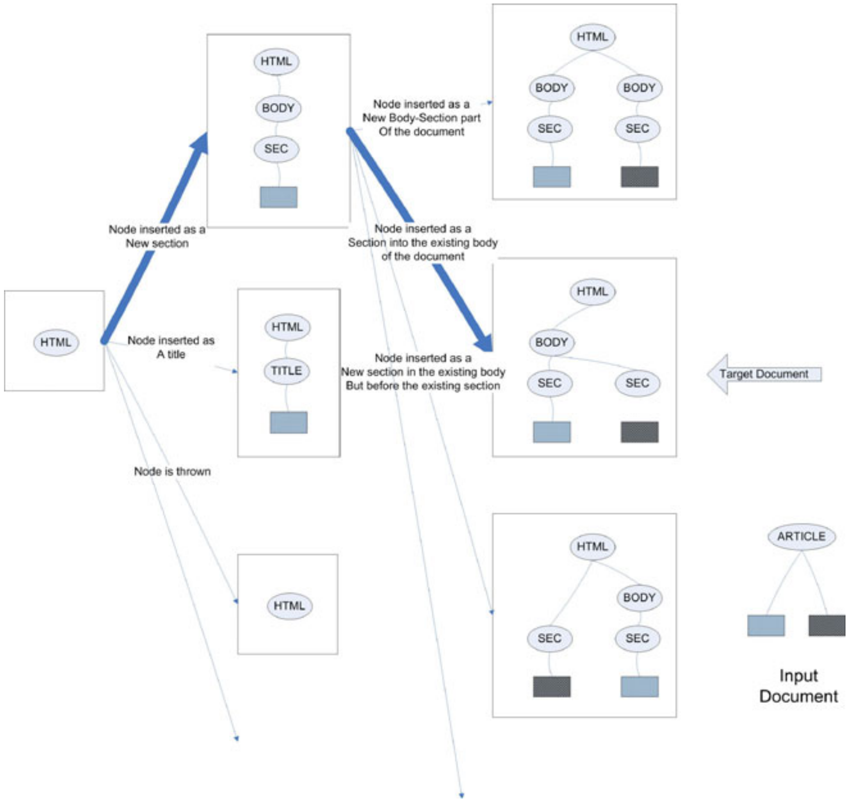

HTML
BODY
SEC
Node inserted as a New section
Node inserted as a New Body-Section part Of the document
HTML
BODY
BODY
SEC
SEC
Node inserted as a Section into the existing body of the document
HTML
Node inserted as A title
HTML
TITLE
Node inserted as a New section in the existing body But before the existing section
HTML
BODY
SEC
SEC
Target Document
Node is thrown
HTML
HTML
BODY
SEC
SEC
ARTICLE
Input Document

Fig. 13.2 This figure is an example of DMDP. The states (squares on the figure) correspond to partial output trees. The initial state (at the left) is the state with the empty document (one HTML node). States at the right of the figure are potential output documents. The edges are the possible actions. In this DMDP, we first choose an action for the first input document node (blue square node), and then choose an action corresponding to the second input document node (black square node).

Let us now instantiate this model for the structure mapping task. We define below the state and action spaces S and A for the problem. The input document is processed sequentially leaf per leaf. n_i will denote the ith leaf node.

At time t, a state s_t contains the input document d and the current output document $\hat{d}_t$. The initial output document $\hat{d}_1$ is the XML root. Each state corresponds to a different partial output document. At each step t, the model has to take a decision concerning the current leaf n_t of the input document d. A final state is reached when all input leaves have been processed *i.e* when $t = \perp_d$ where $\perp_d$ is the number of leaves of d. We denote state $s_{\perp_d}$ a final state and $\hat{d}$ the corresponding predicted output document for input document d.

In a given state s_t, the model has to choose an action for the current input node n_t according to the policy. We consider here two elementary actions: `Skip` and

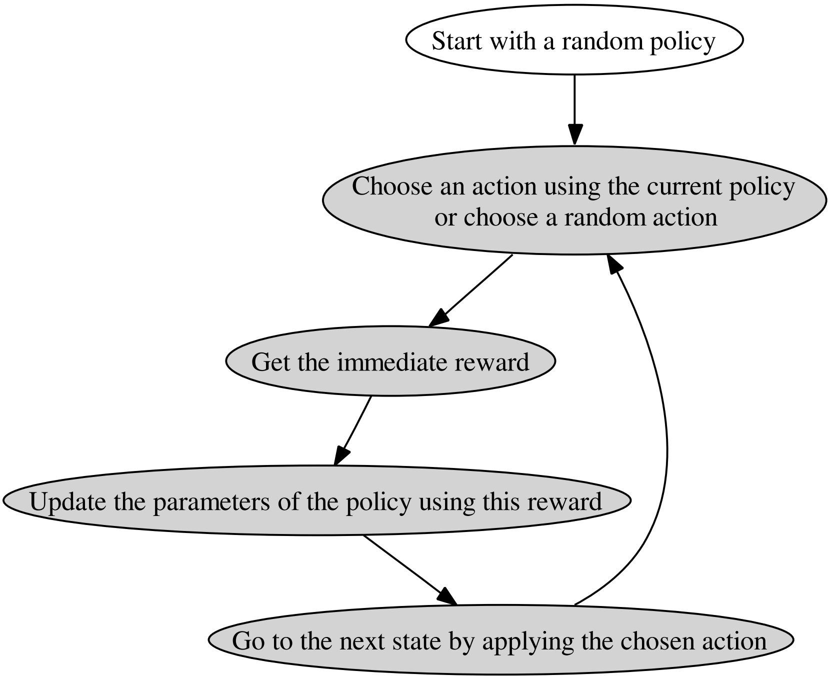

Start with a random policy
Choose an action using the current policy or choose a random action
Get the immediate reward
Update the parameters of the policy using this reward
Go to the next state by applying the chosen action

Fig. 13.3 The learning algorithm is an iterative algorithm which chooses an action and then updates its parameters according to the immediate reward obtained by applying this action at the current state.

`Create(Tags, Position)`. The former skips the input node so that it will not appear in the output document. The latter creates a path whose tag sequence is `Tags` and leaf is n_t. This path is rooted at position `Position` in the current output tree. The addition of a new node must comply with the target DTD or XML schema. Note that the `Create` action allows us to handle complex transformations like leaves reordering, relabeling,etc.

13.4.3 Modeling the Policy

At each state s_t, the model will choose an action a_t according to the policy π_θ. The policy used in our model is deterministic and computed via a scoring function $Q_\theta : S \times A \rightarrow \Re$ which estimates the quality of action a in state s. Given this scoring function, the policy is defined as:

$$\pi_\theta(s,a) = \begin{array}{l} 1 \text{ if } a = argmax_{a'} Q_\theta(s,a') \\ 0 \text{ otherwise.} \end{array} \tag{13.2}$$

This means that for each state s, only the most relevant action a according to function Q is selected. The goal of the scoring function $Q_\theta(s,a)$ is to evaluate, for all possible states and actions, the benefit of choosing action a at state s *w.r.t.* to the target output document. In our experiments, Q is a linear combination of features describing (state, action) pairs:

$$Q_\theta(s,a) = \langle \theta, \phi(s,a) \rangle \tag{13.3}$$

where $\langle .,. \rangle$ is the classical dot product, $\phi(s,a)$ is a vectorial joint description of the state-action pair, and θ is a vector of parameters which will be learned by the algorithm. We describe in Table 13.1 some examples of features which have been used in our experiments. Note that we have used mainly different features that can't be all described in this paper and we just give here the general underlying idea about features generation for structure mapping. You can find more details about the different features in [17].

13.4.4 Evaluating and Learning the Policy

Let d^* and $\hat{d}$ be the target and computed outputs for the input document d. The quality of $\hat{d}$ can be computed via a user supplied loss-function $\Delta : D_{out} \times D_{out} \rightarrow [0,1]$ which measures the dissimilarity between d^* and $\hat{d}$.

The *optimal policy* is the one which given any state s in the state graph will lead to the best approximation $\hat{d}$ of d^* which can be reached from s. Said otherwise, suppose that the process has reached a state s, the optimal policy will always find a sequence of actions which will lead to the best possible output (as measured by $\Delta(.,.)$) which can be computed from s. Formally, it is defined as the policy which minimizes the expectation of the loss computed over all states of the state graph.

One way to learn the corresponding optimal parameters consists in exploring all the states of the state space *i.e* computing all the losses obtained by choosing the different possible actions for all states and then using a classical regression algorithm in order to compute the parameters θ^* of the optimal policy. This is usually intractable due to the size of the state space and we have to use *sampling techniques* that allow us to explore only a small but relevant part of this space.

Reinforcement learning algorithms like SARSA learning or Q-Learning [22] use the immediate reward function $r(s,a)$ in order to find the Q function that maximizes the cumulative reward over all the actions taken. The cumulative reward is the sum of the immediate rewards obtained at each state for each action taken. The immediate reward provides information used to guide the exploration of the state space so as to focus only on the relevant parts of the space.

For the structure mapping problem, we propose to define a reward function based on the loss function provided by the user. We use the following immediate reward function:

$$r(s_t,a_t) = \begin{array}{ll} -\Delta(\hat{d}_{t+1},d^*) & \text{if } t=0, \\ \Delta(\hat{d}_t,d^*) - \Delta(\hat{d}_{t+1},d^*) & \text{otherwise.} \end{array} \quad (13.4)$$

where $\hat{d}_t$ is the partial output document at state s_t and $\hat{d}_{t+1}$ is the partial output document obtained by applying action a_t to state s_t. $\Delta(\hat{d}_t,d^*)$ measures a dissimilarity between partial document $\hat{d}_i$ and target document d^*. The reward thus measures the improvement made on the partial output document $\hat{d}_t$ after the t-*th* action which has led to partial document $\hat{d}_{t+1}$.

The cumulative obtained reward is then:

$$R(s_{\perp_d}) = \sum_{t=0}^{\perp_d} r(s_t,a_t) = -\Delta(\hat{d},d^*) \quad (13.5)$$

which corresponds to the negative loss. So, the parameters θ^* that maximize the cumulative reward also minimize the loss over the predicted output document and the real output document.

Once the model is learned, the complexity for computing a mapping is $O(\perp_d * ||\hat{A}_s||)$ where $||\hat{A}_s||$ corresponds to the mean number of possible actions per state s and $\perp_d*$ is the number of leaves of $d*$ The complexity of the inference is linear *w.r.t.* the number of leaves of d times the number of possible actions and thus the inference is tractable on large document.

13.5 Experiments

We present experiments performed on different corpora. We learn for each of them to map the input documents expressed in one or two homogeneous input format onto a target XML schema. These corpora have very different characteristics and illustrate how the proposed method behaves on a variety of situations and mapping tasks.

13.5.1 Datasets

We used three medium-scale datasets and two large-scale datasets that are described below:

- RealEstate [10]. This corpus, proposed by Anhai Doan[4] is made of 2,367 data-oriented XML documents. The documents are expressed in two different XML formats. The aim is to learn the transformation from one format to the other.
- Mixed-Movie [7]. The second corpus is made of more than 13,000 movie descriptions available in three versions: two mixed different XHTML versions and

[4] `http://www.cs.wisc.edu/~anhai/`

Table 13.2 *Tree transformation corpora properties and statistics:* The top part of the table gives properties of the tree transformation problem. From top to bottom: the name of the corpus, the input format, the target format and two flags that tell if the transformation requires node skipping and if it requires node reordering. The bottom part of the table gives statistics on the target documents of the dataset. From top to bottom: the number of documents, the mean number of internal nodes per tree, the mean number of leaves per tree, the mean tree depth and the number of output labels.

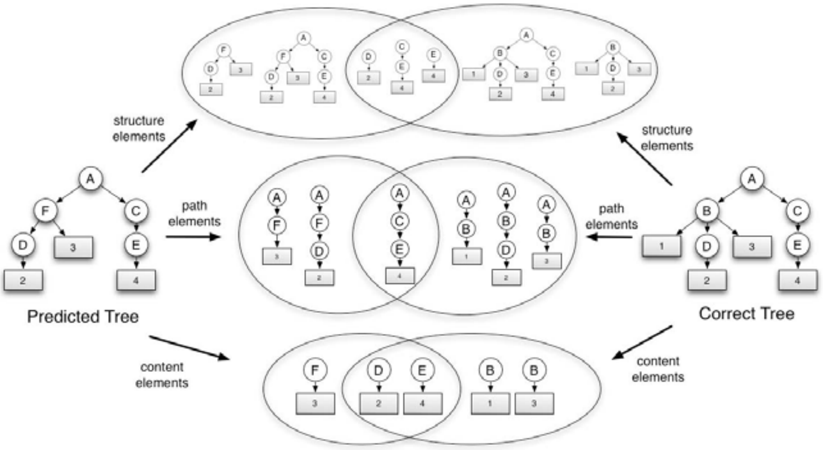
Corpus	RealEstate	Mixed-Movie	Shakespeare	Inex-Ieee	Wikipedia
Input Format	XML	HTML	Segmented Text	Segmented Text	HTML
Target Format	XML	XML	XML	XML	XML
Node skipping	yes	yes	no	no	yes
Node reordering	yes	yes (±10)	no	no	yes (±10)
Num. Documents	2,367	13,048	750	12,107	10,681
Internal Nodes	about 33	about 64	about 236	about 650	about 200
Leaf Nodes	about 19	about 39	about 194	about 670	about 160
Depth	about 6	5	about 4.3	about 9.1	about 7.7
Labels	37	35	7	139	256

one XML version. This corresponds to a scenario where two different websites have to be mapped onto a predefined mediated schema. The transformation includes node suppression and some node displacements.

- Shakespeare [3]. This corpus is composed of 60 Shakespeare scenes[5]. These scenes are small trees, with an average length of 85 leaf nodes and 20 internal nodes over 7 distinct tags. The documents are given in two versions: a flat segmented version and the XML version. The tree transformation task aims at recovering the XML structure using only the text segments as input.
- Inex-Ieee [13]. The Inex-Ieee corpus is composed of 12,017 scientific articles in XML format, coming from 18 different journals. The tree transformation task aims at recovering the XML structure using only the text segments as input.
- Wikipedia [6]. This corpus is composed of 12,000 wikipedia pages. For each page, we have one XML representation dedicated to wiki-text and the HTML code. The aim is to use the layout information available in the HTML version, for predicting the semantical XML representation.

The properties and statistics of our corpora are summarized in Table 13.2. For each corpus, we mention if node skipping and node reordering are required to fulfill the transformation. Note that the node skipping and node reordering properties have effect on the set of possible actions allowed in the model. In the general case (skipping and reordering), the set of actions can be very large and, in the experiments, we have used additional constraints to limit this size. For example, we only consider actions that result in an output document where node paths have been seen in the training set. More details on the constraints used can be found in [17].

[5] http://metalab.unc.edu/bosak/xml/eg/shaks200.zip



$$F_{structure} = \frac{2 \times 3}{5+5} = 60\% \qquad F_{path} = \frac{2 \times 1}{3+4} \simeq 28.57\% \qquad F_{content} = \frac{2 \times 2}{3+4} \simeq 57.14\%$$

Fig. 13.4 *Computation of the $F_{structure}$, F_{path} and $F_{content}$ similarity measures:* The left and right parts of the figure respectively correspond to the predicted tree and the correct tree. For each similarity measure, the trees are decomposed into set of elements (structure elements, path elements or content elements). The bottom part of the figure gives the similarity scores, which are the F_1 score between predicted and correct elements.

13.5.2 Evaluation Measures

Ideally, evaluating the quality of structure mapping should be made in the context of a specific application as, for example, the search on heterogeneous collections, a specific document conversion task, etc. Our model was developed to solve a large variety of tasks and we propose here to evaluate the general performance of the method by measuring the similarity of computed output documents with target documents. We used three measures, each one corresponds to the mean over the documents of the test corpus of a F_1 measure. The measures used are the following:

- F_{leaf}: F_1 score computed between the label of the leaves of the input and the output document. It measures the capacity of the model for computing the correct leaves labels.
- F_{path}: F_1 score computed between the paths of the input and output document. A path corresponds to the set of ordered labels from the root node to a leaf node. This measure evaluates the ability of the model at discovering the structure of the output tree.
- $F_{subtree}$: F_1 score computed between all the subtrees of the input and output document. This measure reflects the percentage of common subtrees. Two subtrees are equals if they share exactly the same nodes in the same order. Note that this measure decreases quickly with only a few errors because each node is involved

in many subtrees. For example, only one error on a medium-sized document gives a $F_{subtree}$ measure of around 80%.

The different measures are illustrated in Figure 13.4.

13.5.3 Comparison to Baseline Models

We compare our model to both a real model and some artificial ones:

- The PCFG+ME model proposed in [3] is the model closest to ours. Due to its complexity (roughly cubic in the number of nodes whereas ours is linear), this baseline model can only be used on small corpora. For the same reason, there is no competing model which can deal with the larger corpora.
- The artificial models are incremental models based on a greedy policy which is computed by using the F measures presented previously.These greedy policies make use of the correct output and select actions whose execution most increase the immediate $F_{structure}$ or F_{path} similarity scores. The scores of $\pi^{greedy}_{structure}$ could be considered as upper bounds for the performance reachable by learning to maximize the immediate reward, *e.g.* with SARSAand a discount factor of 0. We also computed the scores of the random policy π^{random}, which selects actions randomly.

Some methods in the field of *Information Extraction* or *Wrapper Induction* are able to extract some pieces of structure from XML documents (names, locations, etc.) but these methods are not able to transform a whole document to a mediated schema. As far as we know, our approach is the only one able to perform the structure mapping task on large **XML documents** corpora.

13.5.4 Results

We only have one *learning-based* baseline on two of these datasets for the complexity reasons explained in Section 13.3.3. The baseline model PCFG+ME [3] is a global model for *one-to-one* tree transformation. It models the probability of outputs by using probabilistic context free grammars (PCFG) and maximum-entropy (ME) classifiers. Inference is then performed with a dynamic-programming algorithm that has a cubic complexity in the number of input leaves. This model can only operate on the two smallest datasets RealEstate and Shakespeare.

Our experimental results on the three medium-scale datasets are given in Table 13.3. On RealEstate and Mixed-Movie, the reinforcement learning approaches give significantly better results than the greedy policy baselines. This result is very satisfactory and has two major implications. Firstly, it shows that reinforcement-learning algorithms are able to find a *better strategy than greedily moving* toward the correct output. Secondly, it shows that the algorithms perform an *effective learning and generalization* of this strategy. On Shakespeare, the scores of reinforcement learning algorithms is slightly inferior to those of the greedy policies. Since greedy policies

Table 13.3 *Tree transformation with collapsed actions, medium-scale datasets:* For each dataset and each method, we give the three average similarity scores $F_{structure}$, F_{path} and $F_{content}$ between the predicted and correct trees of the test set. The first column corresponds to the SARSA reinforcement learning algorithm. The next three columns are non-learning baselines. The last column is the PCFG+ME [3] baseline. The / symbol denotes results that could not be computed due to the complexity of PCFG+ME.

		RL	Baselines			
Corpus	Score	SARSA	$\pi^{greedy}_{structure}$	π^{greedy}_{path}	π^{random}	PCFG+ME
RealEstate	$F_{structure}$	99.54	87.09	**97.09**	3.27	49.8
	F_{path}	99.87	84.42	**100**	3.91	7
	$F_{content}$	99.88	**100**	**100**	5.10	99.9
Mixed-Movie	$F_{structure}$	86.22	**47.04**	44.15	3.54	/
	F_{path}	91.53	52.02	**52.18**	5.29	/
	$F_{content}$	91.53	52.02	**52.18**	5.67	/
Shakespeare	$F_{structure}$	**96.03**	**98.65**	75.16	11.34	94.7
	F_{path}	**97.88**	98.91	**100**	16.47	97.0
	$F_{content}$	**98.87**	99.83	**100**	18.25	98.7

perform nearly perfectly on this corpus, the main difficulty here is to *generalize* over the whole corpus.

The RealEstate collection corresponds to an *XML database* where we need to label the leaves correctly and to put them in the correct order. The task is easy, and most RL approaches achieve $>$ 99% on the different scores. PCFG+ME only performs 7 % on the F_{path} score and about 50 % on the $F_{structure}$ score because it does not handle node skipping and node reordering, which are required on this collection. On the Shakespeare corpus, PCFG+ME gives slightly lowers results than RL methods.

In order to demonstrate the scalability of our approach, we have performed experiments with SARSA on the two large-scale corpora. These experiments required $\approx$ 5 days training time in order to perform 1000 training iterations, *i.e.* 10^5 SARSAepisodes, for each dataset. This huge amount of time has to be contrasted with the scale of the task: these corpora involve particularly large documents (the biggest documents in these corpora contain up to 10.000 nodes), complex operations (nodes displacements or nodes deletions), highly heterogeneous documents and large number of labels (139 labels for Inex-Ieee and 256 labels for Wikipedia).

The results for large-scale tree transformation are given in Table 13.4. On Wikipedia, SARSA outperforms the scores of the greedy policy baselines, which is very satisfactory, given the large number of labels on this corpus. On Inex-Ieee, SARSA does not reach the level of the greedy policy baselines. However, this corpus contains a huge amount of noise, which could explain this result.

Table 13.4 *Tree transformation with collapsed actions, large scale datasets:* For each method and each dataset, we give the three average similarity scores on the test set. We compare SARSA with the baseline policies, which do not use learning.

		RL	Baselines		
Corpus	Score	SARSA	$\pi^{greedy}_{structure}$	π^{greedy}_{path}	π^{random}
Inex-Ieee	$F_{structure}$	67.5	**76.32**	49.94	2.17
	F_{path}	74.4	39.23	**97.20**	1.00
	$F_{content}$	75.8	82.91	**97.20**	8.62
Wikipedia	$F_{structure}$	**65.6**	57.37	23.53	5.51
	F_{path}	**74.3**	2.28	32.28	0.12
	$F_{content}$	**80.2**	72.92	39.34	12.35

13.6 Conclusion

We have described a general model for mapping heterogeneous document representations onto a target structured format. This problem has recently appeared as a key issue for querying heterogeneous semi-structured document collections. The model learns the transformation from examples of input and corresponding target document pairs. It is based on a new formulation of the structure mapping problem based on Deterministic Markov Decision Processes and Reinforcement Learning. This formulation allows us to deal with a large variety of tasks ranging from the automatic construction of a target structure from flat documents to the mapping of XML and HTML collections onto a target schema. Experiments have been performed for different mapping tasks using four corpora with different characteristics. They have shown that the model scales well with large collections and may achieve high accuracy for different mapping tasks.

References

[1] Boukottaya, A., Vanoirbeek, C.: Schema matching for transforming structured documents. In: ACM DOCENG, pp. 101–110 (2005)

[2] Castano, S., Antonellis, V.D., di Vimercati, S.D.C.: Global viewing of heterogeneous data sources. IEEE Trans. Knowl. Data Eng. 13(2), 277–297 (2001)

[3] Chidlovskii, B., Fuselier, J.: A probabilistic learning method for xml annotation of documents. In: IJCAI (2005)

[4] Chung, C.Y., Gertz, M., Sundaresan, N.: Reverse engineering for web data: From visual to semantic structure. In: ICDE, pp. 53–63 (2002)

[5] Collins, M., Roark, B.: Incremental parsing with the perceptron algorithm. In: ACL 2004, Barcelona, Spain, pp. 111–118 (2004)

[6] Denoyer, L., Gallinari, P.: The wikipedia xml corpus. In: SIGIR Forum (2006)

[7] Denoyer, L., Gallinari, P.: Report on the xml mining track at inex 2005 and inex 2006. In: SIGIR Forum, pp. 79–90 (2007)

[8] Doan, A., Domingos, P., Levy, A.Y.: Learning source description for data integration. In: WebDB (Informal Proceedings), pp. 81–86 (2000)

[9] Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: A machine-learning approach. In: SIGMOD Conference, pp. 509–520 (2001)

[10] Doan, A., Domingos, P., Halevy, A.: Learning to match the schemas of data sources: A multistrategy approach. Maching Learning 50(3), 279–301 (2003), doi: http://dx.doi.org/10.1023/A:1021765902788

[11] Embley, D.W., Jackman, D., Xu, L.: Multifaceted exploitation of metadata for attribute match discovery in information integration. In: Workshop on Information Integration on the Web, pp. 110–117 (2001)

[12] Fuhr, N., Govert, N., Kazai, G., Lalmas, M.: INDEX: Initiative for the Evaluation of XML Retrieval. In: SIGIR 2002 Workshop on XML and IR (2002)

[13] Fuhr, N., Gövert, N., Kazai, G., Lalmas, M. (eds.): Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, Germany, December 9-11 (2002)

[14] Howard, R.A.: Dynamic Programming and Markov Processes. Technology Press-Wiley, Cambridge, Massachusetts (1960)

[15] Leinonen, P.: Automating xml document structure transformations. In: ACM DOCENG, pp. 26–28 (2003)

[16] Li, W.S., Clifton, C.: Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data Knowl. Eng. 33(1), 49–84 (2000)

[17] Maes, F.: Choose-reward algorithms incremental structured prediction, learning for search and learning based programming. PhD in Computer Science, University Pierre and Marie Curie, LIP6 (2009)

[18] Palopoli, L., Saccà, D., Ursino, D.: Semi-automatic semantic discovery of properties from database schemas. In: IDEAS, pp. 244–253 (1998)

[19] Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. 10(4), 334–350 (2001)

[20] Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. J. Data Semantics IV, 146–171 (2005)

[21] Su, H., Kuno, H.A., Rundensteiner, E.A.: Automating the transformation of xml documents. In: WIDM, pp. 68–75 (2001)

[22] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). MIT Press, Cambridge (1998), http:www.amazon.co.ukexecobidosASIN0262193981citeulike-21

Chapter 14
Learning Methods for Graph Models of Document Structure

Peter Geibel, Alexander Mehler, and Kai-Uwe Kühnberger

Abstract. This chapter focuses on the structure-based classification of websites according to their hypertext type or genre. A website usually consists of several web pages. Its structure is given by their hyperlinks resulting in a directed graph. In order to represent the logical structure of a website, the underlying graph structure is represented as a so-called *directed Generalized Tree* (GT), in which a rooted spanning tree represents the logical core structure of the site. The remaining arcs are classified as reflexive, lateral, and vertical up- and downward arcs with respect to this kernel tree.

We consider unsupervised and supervised approaches for learning classifiers from a given web corpus. Quantitative Structure Analysis (QSA) is based on describing GTs using a series of attributes that characterize their structural complexity, and employs feature selection combined with unsupervised learning techniques. Kernel methods – the second class of approaches we consider – focus on typical substructures characterizing the classes. We present a series of tree, graph and GT kernels that are suitable for solving the problem and discuss the problem of scalability. All learning approaches are evaluated using a web corpus containing classified websites.

Peter Geibel
Fak. IV, TU Berlin, Straße des 17. Juni 135, D-10623 Berlin, Germany
e-mail: info@peter-geibel.de

Alexander Mehler
Computer Science and Informatics, Goethe-Universität Frankfurt,
Senckenberganlage 31, D-60325 Frankfurt am Main, Germany
e-mail: Mehler@em.uni-frankfurt.de

Kai-Uwe Kühnberger
Universität Osnabrück, Institute of Cognitive Science, Albrechtstraße 28,
D-49076 Osnabrück, Germany
e-mail: kkuehnbe@uos.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 267–298.
springerlink.com

14.1 Introduction

In this chapter, we consider the classification of websites (i.e., collections of web pages) according to their genre [48]. Examples of such types are conference websites, corporate sites, electronic encyclopedias, online shops, personal and academic home pages, weblogs, etc. [56, 58, 62]. The structure of a website is influenced by its content and function, and can be seen as a source of features for detecting the (main) topic of a site (content-related) or its genre (function-related). In this chapter, we focus on genre by exploring structural features of websites. These structural features might pertain to the structure of single documents and to the structure of the whole collection of pages spanned by their hyperlinks.

We consider the learning of classifiers from a given corpus, which allow to determine the hypertext type of a website. This question, which is of interest in computational linguistics [59], is also highly relevant for information retrieval. Being able to determine the hypertext type of a site returned as a result to a query, we can group the list of hits accordingly and return it to the user in a more structured and organized way [64].

Content-related approaches usually apply the bag-of-words approach to represent single documents by the frequency of their lexical constituents [e.g., 7]. This allows for applying approaches to feature-based learning as, for example, *Support Vector Machine*s (SVM) introduced by Vapnik [66]. This approach frequently delivers classifiers with high accuracy in the domain of text categorization [32]. Structure-based approaches are rarely found in this area [e.g., 69, 27, 45]. One reason is that in order to learn the topic or genre of a (web) document by means of its structure, one has to define and process training sets with labeled trees and graphs that represent the respective DOM trees (Document Object Model) and hyperlink structure, respectively. This task is considerably more complex than learning based on feature vectors.

In previous work [27, 45], we considered the classification of single documents that were represented as DOM trees, that is, labeled ordered trees. We considered two types of approaches:

- In *Quantitative Structure Analysis* (QSA), input graphs are characterized by feature vectors that describe their topological complexity. That is, indices are used or newly invented (according to the specifics of the object area) that model structural characteristics of the micro-, meso- or macro-level of the input graphs.[1] These indices range from features as simple as the average geodesic distance to the entropy of the distribution of the fraction of triggered vertices as introduced by [44]. Since QSA maps graphs onto feature vectors, it allows for supervised learning based on SVMs.
- As an alternative approach, we also developed a series of *tree kernels*, which allow to characterize a pair of input trees in terms of *common substructures* that

[1] The micro-level relates to local vertex or edge-related invariants, while the meso-level refers to graph clustering, graph motifs and other types of subgraphs. Finally, the macro-level relates to graph invariants that represent the topology of the corresponding graph as a whole.

influence their similarity. In particular, we defined the *Soft Tree Kernel* (SoftTK), which is based on the Parse Tree Kernel by [9], which we combined with a kernel for node labels and positions, and a set kernel for the children of a node.

Note that QSA and tree kernels correspond to orthogonal approaches for characterizing structures. QSA relies on structural complexity measures, whereas a typical graph kernel is based on computing common (isomorphic) substructures. In this chapter, we extend QSA and the kernel approach to graphs, which is necessary for the classification of websites consisting of several web pages. We will focus on the hyperlink structure of a website. The inclusion of DOM trees of single web pages will only play a minor role.

It is well-known from the field of structural kernels that tree kernels can be computed relatively efficiently, whereas graph kernels tend to be very challenging in terms of computational complexity. We therefore consider the representation of websites by so-called *Generalized Trees* (GTs) [39], in which the core structure is defined by a so-called kernel tree (not to be confused with the kernels used by the SVM) plus additional types of arcs (lateral, upward, downward, and reflexive) that allow for the representation of the potentially cyclic structure of a graph.

In addition to applying an unsupervised version of QSA to GTs, we present the following types of kernels for supervised learning:

- *Tree kernels for GTs:* It is possible to apply the Soft Tree Kernel (SoftTK) to the kernel trees of a pair of GTs. We use a feature-based approach to include information on reflexive, lateral, upward and downward arcs.
- *GT kernels:* We present the Soft GT kernel (SoftGTK) for generalized trees that is also based on the SoftTK, but treats the graph underlying a GT as a directed acyclic graph (DAG).
- *Graph kernels:* We consider kernels based on the learning system INDIGO [23, 24, 25, 22], which transforms graphs into feature vectors by computing certain types of attributes based on label sequences.

The article is structured as follows. Generalized trees are described in Section 14.2, followed by a description of QSA (Section 14.3). The class of kernel methods are explored in Section 14.4 including the SoftTK, the two INDIGO kernels, and the SoftGTK. In Section 14.5 we present experimental results for a corpus of webgraphs describing conference, personal, and project websites. Our conclusion can be found in Section 14.6.

14.2 Directed Generalized Trees

Let $G' = (V, E')$ be an undirected connected finite graph. Informally speaking, an undirected generalized tree $G = (V, E, r)$ is a graph rooted in r such that this graph is induced by a rooted spanning tree $T = (V, E'', r)$ of G where any edge $e \in E$ is classified as either a *kernel* (if belonging to T), *lateral* or *vertical* edge.[2] That is,

[2] The range of edge types varies with the underlying application area. See, for example, [42] who distinguishes seven types by example of directed generalized trees.

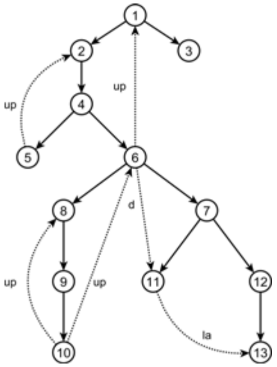


Fig. 14.1 Visual depiction of a directed generalized directed tree with four upward arcs, one downward arc (d) and one lateral arc (la) [39].

generalized trees are graphs whose edges are classified subject to selecting a certain spanning tree together with a certain root. Analogously, directed generalized trees D are digraphs whose arcs are classified as a result of selecting a rooted sub-tree T of D such that the disorientation of T is a spanning tree of the disoriented variant of D. In what follows we concentrate on directed generalized trees and therefore will widely omit the attribute *directed*. This class of graphs is defined as follows [cf. 42] (see Figure 14.1 for a schematic depiction of generalized trees):

Definition 1. Let $T = (V, A', r)$ be a directed tree rooted in $r \in V$. Further, for any vertex $v \in V$ let $P_{rv} = (v_{i_0},\ a_{j_1}, v_{i_1}, \ldots, v_{i_{n-1}}, a_{j_n}, v_{i_n}), v_{i_0} = r, v_{i_n} = v, a_{j_k} \in A', in(a_{j_k}) = v_{i_{k-1}}, out(a_{j_k}) = v_{i_k}$, $1 \leq k \leq n$, be the unique path in T from r to v such that $V(P_{rv}) = \{v_{i_0}, \ldots, v_{i_n}\}$ is the set of all vertices on that path. A *Directed Generalized Tree* $G = (V, A_{1..5}, r)$ based on the kernel tree T is a pseudograph whose arc set is partitioned in the sense that $A_{1..5} = \cup_{i=1}^{5} A_i$, $\forall 1 \leq i < j \leq 5 : A_i \cap A_j = \emptyset$, $a \in A_{1..5}$ iff $a \in \cup_{i=1}^{5} A_i$ and

$$\begin{array}{lr}
A_1 = A' & \textit{(kernel arcs)} \\
A_2 \subseteq \{a \,|\, in(a) = v \in V \wedge out(a) = w \in V(P_{rv}) \setminus \{v\}\} & \textit{(upward arcs)} \\
A_3 \subseteq \{a \,|\, in(a) = w \in V(P_{rv}) \setminus \{v\} \ \wedge \ out(a) = v \in V\} & \textit{(downward arcs)} \\
A_4 \subseteq \{a \,|\, in(a) = out(a) \in V\} & \textit{(reflexive arcs)} \\
A_5 \subseteq V^2 \setminus (A_1 \cup A_2 \cup A_3 \cup A_4) & \textit{(lateral arcs)}
\end{array}$$

G is said to be generalized by its reflexive, lateral, up- and downward arcs.

At this stage, the following question arises: *Why do we care about the notion of generalized trees instead of just speaking about connected graphs?* The reason behind this notion is manifold due to empirical and formal considerations:

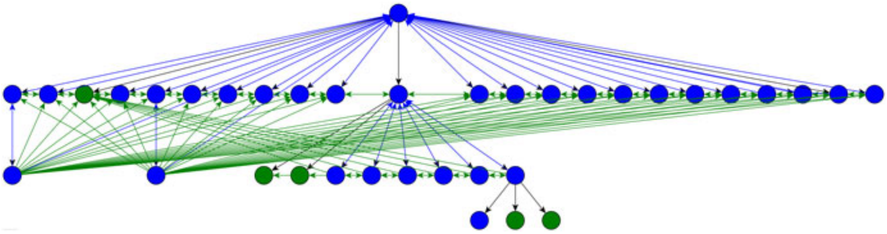


Fig. 14.2 The ACM Multimedia 2005 website represented as a generalized tree [from 42]. Vertices denote HTML pages or resources in the form of, for example, PDF documents.

- Generalized trees are used to model a range of *empirical*, *non-formal* systems like *websites* [15, 16, 42], *social ontologies* [41], *semantic spaces* [39], *multimodal data* [43] or *discourse structures* [46]. Further, [40] introduces a new class of Markovian spanning trees that are motivated in terms of cognitive semantics. This shows that kernel trees of generalized trees are not restricted to well-known classes of spanning trees. Rather, generalized trees are a sort of text-technological data structure that captures a certain range of empirical systems. This also means that the selection of the kernel tree T of a generalized tree G is restricted by the underlying empirical system. In the case of natural language texts, for example, T may correspond to their logical document structure [53] or to their RST-tree [38], while in the case of websites, T may be induced by the logical hypertext document structure of the corresponding website [42].
- In formal terms, generalized trees are digraphs whose arcs are typed *non arbitrarily*. That is, generalized trees provide more information than digraphs without typed arcs. This typing is immediately of help when considering the task of graph similarity measuring or graph matching, since corresponding algorithms can profit from the inherent tree-like structure of generalized trees. This kernel provides a way to reduce the complexity of measurements especially if being determined by the underlying application area in a predictable manner. Natural language texts, for example, are characterized by their logical document structure that provides information about their text type [27, 45, 54].
- Finally, [39] shows that by typing arcs in generalized trees, we get a basis for a functional interpretation of these arcs in terms of information retrieval. In this sense, the notion of generalized trees serves as a bridge between a purely formal data structure and its semantic interpretation according to structure models in cognitive science [cf. 35].

In this chapter, we utilize the notion of directed generalized trees to learn web-genres. As instances of certain genres, we represent websites as directed generalized trees to get graph representations as input to structure learning (see Figure 14.2 for the depiction of a website represented as a generalized tree). In this sense, we introduce an approach to learning a class of graphs in the range of trees and unrestricted graphs.

14.3 Quantitative Structure Analysis by Example of Websites

Generally speaking, *Quantitative Structure Analysis* (QSA) is an approach to learning classes of (discourse) structures based on their salient, distinctive features while disregarding their content. This relates to the micro-level of vertices and their arcs, the meso-level of subgraphs and the macro-level of graphs as a whole as models of certain empirical structures. Starting from a classification of a corpus of instances of a certain application area, QSA seeks structural features that best distinguish them according to this classification. In terms of discourse analysis, QSA is based on the idea that the structure of these instances reveal their class membership (either according to some predominant function or content) [2, 17]. Thus, QSA goes along with developing structural models of the underlying application area as based, for example, on their entropy [14], distance structure [3], clustering [5] or modularity [49]. QSA involves three steps of modeling:

1. *Quantitative graph modeling*: firstly, each target object (i.e., graph) is represented by a vector of topological indices that model its micro, meso or macro level structure. Ideally, these indices are developed based on a thorough analysis of the distribution of structures in the corresponding object area.
2. *Feature selection:* in the next step, a genetic search is performed to find salient features within the vectors that best separate the target classes (in each round of the search, a fraction of features is reselected at random while all other features are selected according to the list of highest ranked features of the preceding round; further, in each round, several such feature populations compete with each other). This step implements a sort of supervised learning as the search utilizes the F-measure[3] to select feature subsets. Currently, this feature selection activates or deactivates features. However, one may also consider genetic searches that result in weighting their contribution. Note that the process of feature selection may stop at a local maximum as it does not necessarily find an optimal feature subset.
3. *Classification:* Based on the appropriately projected feature vectors, a hierarchical agglomerative clustering is performed together with a subsequent partitioning that is informed about the number of target classes. Within the current implementation of QSA, classification and feature selection occur in a single step in that the feature selection includes a search for the best performing linkage methods and distance measures. We use *weighted linkage* together with the *Mahalanobis distance* and the *correlation coefficient* to perform this step.[4] Note that the Mahalanobis distance handles correlations between different features.

To sum up, QSA takes the set of input objects together with the parameter space of linkage methods and distance measures to find out the feature subset that best

[3] The F-measure is the harmonic mean of precision and recall; see Section 14.5.

[4] The correlation coefficient is transformed into a distance function. Note that we use MATLAB to make any of these computations.

separates the data according to the underlying classification. In the present study, we model directed generalized trees according to [41]. That is, we utilize a subset of indices of complex network theory together with a subset of indices that were invented to model nearly acyclic graphs [60] to capture the structural specifics of websites. Additionally, a subset of measures of graph entropy is used as described in [41]. See [47] for a summary of this quantitative graph model. Finally, the fractions of upward, downward and lateral arcs were considered as additional topological indices. All in all, 77 different topological indices were computed per input website to model its structural characteristics. Frequency oriented indices (e.g., the number of vertices or arcs) were excluded.

Note that QSA explores structural features. This does not mean to rule out content-related features in principle, but to measure the classificatory power of structural features.

14.4 Kernel Methods

The Support Vector Machine (SVM) introduced by [66] is known for its high classification accuracy and can be applied to non-vectorial data like sequences, trees, and graphs by defining an appropriate kernel for the structural class of the data. A kernel is a similarity measure that is required to be positive-semidefinite [see, e.g., 61, 11]. Given an appropriate feature mapping for the structures considered, each kernel also corresponds to the scalar product in the vector space defined by this mapping.

A well-known example of a kernel for structures is the so-called Parse Tree Kernel [9, 50, 51, 33], which operates on parse trees of sentences given a fixed underlying grammar. The Parse Tree Kernel allows the classification of sentences, or parts of them. In the following section, we present the Soft Tree Kernel, which is an extension of the Parse Tree Kernel for general labeled ordered trees. A slight extension allows the application of the Soft Tree Kernel to GTs. In section 14.4.2, we present an additional kernel, SoftGTK, that is also based on the Soft Tree Kernel, but incorporates more elaborate extensions specifically tailored for generalized trees.

In addition to trees, the SVM can be applied to labeled graphs by defining appropriate graph kernels [26, 29, 13, 63, 34, 20, 18, 36, 21, 52, 37]. In Section 14.4.3, we describe two graph kernels derived from the feature extraction algorithm employed in the learning system INDIGO: Based on the INDIGO context kernel, we explore the INDIGO label sequence kernel and discuss its relationship to the random walk kernel [19, 34]. Both the context kernel and the label sequence kernel can be used with a training set of generalized trees, although they are also applicable to arbitrary labeled directed graphs.

14.4.1 The Soft Tree Kernel for GTs

The well-known Parse Tree Kernel can be used for solving learning and classification problems for parse trees generated from a given grammar. For a parse tree, the grammar rule applied to a non-terminal determines the number, type, and sequence of the children. Our goal will be to describe a tree kernel that is applied to the kernel tree of a GT, and additionally takes into account some of the information given by other types of arcs (i.e., reflexive, lateral, upward, and downward). To this end, we first describe the Soft Tree Kernel (SoftTK) introduced by [27], followed by the extensions for GTs.

In the following, we consider node-labeled ordered trees. The node labels will later on be used to encode the information given by reflexive, lateral, upward and downward arcs. We assume arcs (i.e., directed edges) $E \subseteq V \times V$ where V is the set of nodes of tree T. For a node $v \in V$, we denote the ordered set of children as $\{v_1, \ldots, v_{n(v)}\}$, where $n(v)$ denotes its (out-)degree which is defined as 0 for tree leaves. The root of the tree, r, has only outbound arcs.

We consider ordered trees whose nodes are labeled by a function $\alpha : V \longrightarrow \Sigma$, where Σ is a set of node labels, which will later on be used for defining indicator functions for the different types of GT arcs. Since Σ is a general set, it is also possible to label nodes with specific attributes like tags, words, etc. In this case, each element of Σ corresponds to a tuple, list, or set of (elementary) labels. For the task at hand, however, it is most natural to label a node (representing a single web page) with its DOM tree. In order to work with node labels inside the SoftTK, we have to provide an appropriate node kernel that defines the overall similarity of node labels in an appropriate manner, see below.

In what follows we assume that the input trees are *ordered* because the children of each node are considered a sequence and not a set. Since the ordering information might not be relevant in the case of generalized trees, the Soft Tree Kernel has a parameter γ that determines how order-sensitive the kernel is.

As a last definition, two trees T and T' are called *isomorphic* if there is a bijective mapping of the nodes that preserves the structure given by the arcs, the ordering of the children and the labellings specified by α and α'.

Kernels in general, and the Parse Tree Kernel in particular, can be defined by specifying a sequence of features t. This way each input structure can be described by a feature vector. Based on this projection, the kernel is defined to correspond to the dot product in a feature space. In the case of the Parse Tree Kernel, the features or pattern trees t correspond to incomplete parse trees, in which some productions have not yet been applied, yielding leaves labeled with non-terminals. In the following, we will use the general graph-theoretic term *subtree* for referring to these trees. Note, however, that other authors distinguish between subtrees, subset trees, and partial trees [e.g., 51], which correspond to specific kinds of subgraphs of parse trees.

For a tree T and a feature or pattern tree t in the tree sequence, we define $\phi_t(T)$ as the number of subtrees of T being *isomorphic* to the feature tree t. Let $\phi(T)$ denote the arbitrarily ordered sequence of all feature values, i.e., numbers, obtained by this procedure. Based on this sequence, the Parse Tree Kernel is defined as

$$k(T,T') = \langle \phi(T), \phi(T') \rangle = \sum_t \phi_t(T) \cdot \phi_t(T') \,. \tag{14.1}$$

The index t ranges over all pattern trees in the sequence. Since there is only a limited number of subtrees shared by two input trees T and T', $k(T,T')$ results in a finite number despite the infinite set of potential subtrees. Note that the term $\phi_t(T) \cdot \phi_t(T')$ corresponds to the number of possible isomorphic mappings of subtrees (isomorphic to t) between T and T'.

Collins and Duffy showed that k can be computed efficiently by directly determining the number of possible mappings of isomorphic subtrees instead of explicitly generating all possible pattern trees. Let $v \in V$ and $v' \in V'$. The function $\Delta((v,T),(v',T'))$ is defined as the number of isomorphic mappings of subtrees rooted in v and v', respectively. It holds that

$$k(T,T') = \sum_{v \in V, v' \in V'} \Delta((v,T),(v',T')) \,. \tag{14.2}$$

Note that Δ is a tree kernel as well, in which v and v' serve as "pointers" to the respective subtrees. That is, (v,T) corresponds to the subtree of T that is rooted in v. In contrast to k, however, the pattern trees t underlying Δ are required to be subtrees restricted to such parse trees that start in the root. The next trick of the kernel computation is to compute Δ in a recursive manner.

14.4.1.1 The Soft Tree Kernel (SoftTK)

We will leave out the technical details of the Parse Tree Kernel and refer the reader to the work by [9, 10]. In the following, we describe a tree kernel introduced by [27], which is called the *Soft Tree Kernel (SoftTK)*. This tree kernel additionally incorporates kernel values $k_\Sigma(\alpha(v), \alpha'(v'))$ for the node labels, which comes in handy for generalized trees. For two node labels $\alpha(v)$ and $\alpha'(v')$ from the set of all labels, Σ, the value of $k_\Sigma(\alpha(v), \alpha'(v'))$ correspond to their similarity. The simplest choice for k_Σ is the identity function. Dependent on the type of the elements in Σ (i.e., tuples, vectors, sets, lists, trees etc.) more elaborate kernels can be used, which better reflect the semantics of the domain. For the problem at hand, in which a node represents a single web page, a natural choice for k_Σ would be a tree kernel operating on DOM trees. Later on, we will extend any given k_Σ with information specific for GTs.

The SoftTK takes the position of a node in the child sequence into account. The position i of some child v_i of a node v is basically used as an additional attribute of the respective node. When comparing two positions, their similarity can be based on their numerical distance. This suggests to use the RBF kernel defined as

$$k_\gamma(i,i') = e^{-\gamma(i-i')^2} \tag{14.3}$$

for node positions i and i'. The maximum value is attained for $i = i'$. The value of the parameter γ is to be set by the user. It determines how much a difference in position diminishes the value of the kernel. Note that the comparison of positions can be considered soft or "fuzzy", which lead to the name of the kernel. Note also that setting $\gamma = 0$ results in the Set Tree Kernel also defined by [27]. [12], [33], and [4] also described kernels allowing for the comparison of node labels. However, they do not pertain to a "soft" comparison of node positions, which is the crucial point in defining the SoftTK.

The Δ-function of the Soft Tree Kernel is now defined recursively as

$$\Delta_S((v,T),(v',T')) = k_\Sigma(\alpha(v),\alpha'(v')) + \lambda \Psi_S((v,T),(v',T')) \tag{14.4}$$

with a "convolution part" defined as

$$\Psi_S((v,T),(v',T')) = \sum_{i=1}^{n(v)} \sum_{i'=1}^{n'(v')} k_\gamma(i,i') \cdot \Delta_S((v_i,T),(v'_{i'},T')) . \tag{14.5}$$

Ψ_S is used for comparing the child tree sequences and can be shown to take the form of a convolution kernel, see below. $\lambda \in [0.0, 1.0]$ is a parameter that controls the influence of the matches between child trees. Basically, we consider every pair of child trees, but the contribution of the kernel value for this pair is discounted by $k_\gamma(i,i')$.

Specifying a feature space interpretation in terms of pattern trees for this kernel is difficult for $\gamma > 0$, because we have to take k_Σ and k_γ into account. However, if k_Σ is the identity kernel and if $\gamma = 0$ and $\lambda = 1$ holds, it can be shown that the features correspond to paths in the tree. In order to be able to prove that SoftTK is a kernel also in the general case, it can be shown that it is a *convolution kernel* - a class of kernels which was introduced by [29]. A convolution kernel operating on structures X and X' looks at possible mappings of their "parts" which are specified using a relation R. The general definition of the convolution kernel is given by

$$k_c(X,X') = \sum_{R(x_1,\ldots,x_n,X),R(x'_1,\ldots,x'_n,X')} \prod_{j=1}^{n} k_j(x_j,x'_j) . \tag{14.6}$$

The relation R relates the part $x = (x_1,\ldots,x_n)$ to X, and the part $x' = (x'_1,\ldots,x'_n)$ to X'. Note that the parts can basically be anything: nodes, arcs, label sequences, degrees, positions, etc. In particular, the parts are allowed to belong to overlapping structures of X and they do not have to cover all of X. Each such part is characterized by n attributes.

In the case of the SoftTK, we set $X = (v,T)$, $X' = (v',T')$, and $n = 2$, and define

$$R(x,X) = \left\{ \Big((i,(v_i,T)),(v,T)\Big) \mid 1 \le i \le n(v) \right\} \tag{14.7}$$

and

$$R(x',X') = \left\{ \Big((i', (v'_{i'}, T')), (v', T') \Big) \mid 1 \le i' \le n'(v') \right\} . \tag{14.8}$$

$n(v)$ and $n'(v')$ denote the number of children of v and v', respectively.

For the proof, we choose $k_1(x_1, x'_1) = k_\gamma(i, i')$, $k_2(x_2, x'_2) = \Delta_S((v_i, T), (v'_{i'}, T'))$. With these definitions, $\Psi_S((v,T),(v',T')) = k_c(X, X')$ takes the form of a convolution kernel (with a grain of salt, because the definition is recursive). According to the general rules for constructing kernels from kernels [e.g., 11], Δ_S and finally k_S are also kernels if Ψ_S is one. Note, however, that in order to complete the proof, we actually have to use induction on the maximum depth of trees considered, because Ψ_S and Δ_S depend on each other. Based on such an induction over tree depth, we get the proof of the following theorem.

Theorem 1. *k_S, Ψ_S and Δ_S are proper kernels.*

Note that we can use multiplication $\cdot$ instead of $+$ for defining Δ_S resulting in $\Delta_S((v,T),(v',T')) = c + k_\Sigma(\alpha(v), \alpha'(v')) \cdot \lambda \Psi_S((v,T),(v',T'))$. The constant $c \ge 0$ allows to reward node mappings independent of how well their actual labels fit. A further variant can be obtained by replacing k_γ with the identity kernel $\iota(i,i')$, which equals to 1 when $i = i'$, and to 0 for $i \ne i'$. This has the effect of reducing the complexity of comparing the child tree sequences from quadratic to linear. This approach will be called *Simple Tree Kernel* (SimTK).

14.4.1.2 Applying the SoftTK to GTs

We now define two indicator functions ρ^-_R and ρ^+_R for each type of relation (reflexive, lateral, vertical upward and downward arcs). According to the definition of GTs (Def. 1), the index R is in the set $\{2,3,4,5\}$. For a node $v \in V$, $\rho^-_R(v)$ denotes the number of outgoing arcs belonging to the respective type, whereas $\rho^+_R(v)$ denotes the respective number of incoming arcs. The original node labeling function α is now extended to

$$\bar{\alpha}(v) = \Big(\alpha(v), \rho^-_2(v), \ldots, \rho^-_5(v), \rho^+_2(v), \ldots, \rho^+_5(v) \Big) . \tag{14.9}$$

Let $\bar{\Sigma}$ be the set of possible labels for the new labeling function $\bar{\alpha}$. The extended kernel function $k_{\bar{\Sigma}}$ is then defined as

$$k_{\bar{\Sigma}}(\bar{\alpha}(v), \bar{\alpha}'(v')) = k_\Sigma(\alpha(v), \alpha'(v')) + \sum_{R=2}^{5} \rho^-_R(v)\rho^-_R(v') + \sum_{R=2}^{5} \rho^+_{R_j}(v)\rho^+_R(v') . \tag{14.10}$$

That is, the new node kernel is a combination of the original one, k_Σ, and the scalar product of the vectors representing the numbers of incoming and outgoing edges of the different types. Note that it is possible to use other (positive) weights for combining the two functions.

Given a pair of GTs, we are now able to apply the SoftTK to the pair of kernel trees. Part of the information given by the other types of arcs is now incorporated into the new node kernel $k_{\bar{\Sigma}}$ used by SoftTK. In the following section, we describe a more elaborate variant of the SoftTK, which is called SoftGTK and incorporates more structural information.

14.4.2 The Soft GT Kernel

Generalized trees correspond to general, labeled graphs for which a rooted spanning tree has been specified. This means that in addition to the original graph structure, one node has been designated root of the so-called kernel tree, and the arcs of the GT have been labeled as kernel arcs (k), downward arcs (d), upward arcs (u), reflexive arcs (r), and lateral arcs (l). In the following, we describe the Soft GT Kernel (SoftGTK) which is an extension of the SoftTK described in the last section.

In order to devise a GT kernel, we first note that SoftTK defined in Section 14.4.1 can also be applied to DAGs (*directed acyclic graphs*) without modification. From the perspective of a tree kernel, the DAG is (implicitly) transformed into a tree by "duplicating" those nodes and their sub-DAGs, for which more than one inbound arc exists. A paper describing the "reverse" idea was published by [1].

The vertical d-arcs in a GT form such DAG arcs, and so do *inverse* u-arcs. This means that the tree kernel may follow downward arcs as well as the inverse of upward arcs when computing the respective Δ function. This leaves us reflexive r-arcs and lateral a-arcs to deal with.

Reflexive arcs can be accounted for by just labeling the respective node using an additional binary boolean attribute, which is true whenever the node has an reflexive arc. This feature will be denoted as $\rho_4(v,T)$ in the following (similar to the one used by SoftTK). Lateral arcs, however, are much harder to handle because just following them might lead the algorithm into a cycle. In order to prevent such cycles, we might "deactivate" such an arc after following it by deleting it from the copy of the generalized tree used in the kernel computation. If (v,v') is such an l-arc, we denote the tree without that arc as $T_{(v,v')}$.

Let us now consider a generalized tree T and a node v in T. In order to define the kernel, we consider the following node sets, which comprise the direct descendants of v with respect to the different types of relations.

- A list of descendants $\mathbf{K} = \{(k_1,T),\ldots,(k_K,T)\}$ with respect to the kernel arcs. The generalized tree T occurs for technical reasons that will become clear later on.
- A list of descendants $\mathbf{D} = \{(d_1,T)\ldots,(d_D,T)\}$ with respect to the downward arcs. In order to define this set we have to assume an ordering of the d-neighbors of v which, for instance, might be given by a natural ordering on the set of nodes. However, this means that for an isomorphism class of generalized trees the value of the kernel might depend on the specific representative unless we chose a kernel that neglects ordering information, which is the case for the set tree kernel.
- A list of descendants $\mathbf{I} = \{(i_1,T),\ldots,(i_I,T)\}$ with respect to inverse upward arcs.

- A list of neighbors $\mathbf{L} = \{(l_1, T_{(v,l_1)}), \dots, (l_L, T_{(v,l_L)})\}$ with respect to lateral arcs. In this case, we also need the reduced tree structures in order to prevent computational cycles.

For a node v' in T' we consider the corresponding "primed" definitions.

The generalized tree kernel is defined in the usual manner via

$$k_G(T,T') = \sum_{v \in V, v' \in V'} \Delta_G((v,T),(v',T')) \,. \tag{14.11}$$

Now consider two nodes v and v' in T and T' respectively. Let $\mathbf{V}$ and $\mathbf{V}'$ be two sets that contain the N resp. N' descendants for one of the relations together with the potentially modified GTs. We now define the convolution part of the Δ-function as

$$\Psi_\gamma(\mathbf{V},\mathbf{V}') = \sum_{i=1}^{N} \sum_{i'=1}^{N'} k_\gamma(i,i') \Delta_G\big((v_i,T_i),(v'_{i'},T'_{i'})\big) \,, \tag{14.12}$$

which is a slight generalization of the Soft Tree Kernel such that every descendant v_i might come with its own, possibly reduced, structure T_i. Note that in the case $\mathbf{V} = \mathbf{L}$ and $\mathbf{V}' = \mathbf{L}'$, the edge leading to the respective child was removed in T_i and $T'_{i'}$ in order to prevent cycles. Note that we obtain the Ψ of the set tree kernel for $\gamma = 0$.

The Δ-function for the generalized tree kernel is defined by

$$\begin{aligned} \Delta_G((v,T),(v',T')) = 1 + k_\Sigma(\alpha(v),\alpha'(v')) + \rho_4(v,T) \cdot \rho_4(v',T') \\ + \big(\lambda_k \Psi_\gamma(\mathbf{K},\mathbf{K}') + \lambda_d \Psi_0(\mathbf{D},\mathbf{D}') + \lambda_i \Psi_0(\mathbf{I},\mathbf{I}') + \lambda_a \Psi_0(\mathbf{L},\mathbf{L}')\big) \end{aligned} \tag{14.13}$$

λ_k, λ_d,λ_i, and λ_l are weights that can be set by the user. γ has to be optimized by the learning algorithm, e.g., by grid search.

Theorem 2. *k_G is a kernel for isomorphism classes of ordered generalized trees. Setting $\gamma = 0$ results in a kernel for unordered generalized trees.*

Proof: If T and T' were trees, then $\lambda_k \Psi_\gamma(\mathbf{K},\mathbf{K}') + \lambda_d \Psi_0(\mathbf{D},\mathbf{D}') + \lambda_i \Psi_0(\mathbf{I},\mathbf{I}') + \lambda_a \Psi_0(\mathbf{L},\mathbf{L}')$ is a kernel because it is just a sum of convolution kernels in which each sub-kernel has a positive weight. Although T and T' are no trees in general, we can construct trees $\tau(v,T)$ and $\tau(v,T')$ such that

$$\Delta_G((v,T),(v,T')) = \Delta_G(\tau(v,T),\tau(v,T')) \,. \tag{14.14}$$

These trees $\tau(v,T)$ and $\tau(v,T')$ can be obtained from the recursive computation of Δ_G applied to (v,T) and (v,T'): we obtain a tree of function calls from it. In this call tree, we can for instance prefix each node name with the tree position of the call to Δ_G in which the node occurs (positions can be represented as sequences of numbers) resulting in the GTs expanded into two larger trees. Infinite growth is prevented by (locally) deactivating lateral arcs after having used them once (see definition of $\mathbf{L}$). This way, we can extract a suitable pair of trees $\tau(v,T)$ and $\tau(v',T')$ from the call tree. The construction of the trees is quite cumbersome yet straightforward and we will leave out the technical details.

Since we have just shown that $\Delta_G((v,T),(v,T')) = \Delta_G(\tau(v,T),\tau(v,T'))$ holds and also that Δ_G is a kernel for trees, we can conclude that $k_G(T,T') = \sum_{v\in V,v'\in V'} \Delta_G(\tau(v,T),\tau(v',T'))$ is also a (convolution) kernel, according to the general rules for constructing kernels.

It even holds that $k_G([T],[T']) := k_G(T,T')$ is a kernel for isomorphism classes of ordered, generalized trees, since we chose the γ-values such that ordering information is only taken into account for the k-arcs. □

Note that a lateral arc originating from a certain node might be followed several times if the respective node is reached via another lateral arc in addition to its incoming kernel arc. Modified strategies for dealing with lateral arcs are possible: complexity can be reduced substantially if we only allow one lateral arc in each recursion. This strategy further reduces the underlying tree used in the proof of Theorem 2 and was used in the experiments.

14.4.3 The INDIGO Context and Label Sequence Kernels

In the following, we will define two kernels for labeled graphs, starting with a feature mapping that was employed in the learning system INDIGO [23, 24, 25, 22]. The advantage of the INDIGO kernels defined in section 14.4.3.1 over most other graph kernels lies in the fact that the expensive computations are done in the feature mapping step, whereas the computation of the kernel itself just corresponds to the standard dot product of vectors (or rather sparse representations). For transforming the input graphs, INDIGO employed two related sets of attributes:

- *Context attributes* describe nodes and arcs in their relational context. The context depth is increased using an iterative re-labeling algorithm to be described later.
- *Label sequence attributes* describe the label sequences occurring in a graph, and can be computed from context attributes.

We start by presenting the context kernel.

14.4.3.1 The INDIGO Context Kernel

INDIGO operates on a training set of classified graphs and outputs a graph classifier. INDIGO was successfully applied to a couple of domains including the prediction of the mutagenicity of chemical compounds. The output of INDIGO is a so-called decision tree [31, 6, 55] whose inner decision nodes are labeled with patterns for graphs–an idea first described by [68, 65]. These patterns are generated by a procedure that will be described in the following. The computed patterns or features form the basis of the two INDIGO kernels to be described later in this section.

INDIGO's procedure for constructing features is based on an algorithm for deciding isomorphism between graphs. This algorithm was described by B. Weisfeiler in the 1970s [67]. The procedure is based on simultaneously *re-labeling* nodes and arcs based on the *context* of the respective node or arc. In each step of the algorithm, the new node labels are based on the original node labels and the labels of adjacent arcs. The new label of an arc is determined by its original label, plus the labels of the two adjacent nodes, plus the labels of all "triangular" graph structures, which the respective arc forms a part of (see below).

Note that the new arc and node labels are based on their direct context. By iterating this procedure, the contexts expand exponentially which results in an efficient procedure for checking non-isomorphism. The procedure fails, for instance, for non-isomorphic strongly regular graphs, which are unlikely to occur in practice. In the following, we will described a variant of the procedure, in which the depth of context grows only linearly in each iteration, which is more suitable for learning.

In the following, we will consider artificial web graphs using the node labels p ("is a web page") and r ("has reflexive edge"). The arc label k is used for representing kernel arcs, and l, u, d denote specific types of non-kernel arcs (lateral, upward, and downward arcs). Note that these labels can be considered strings of characters "p", "r", "k", "l", "u", and "d". However, in what follows we left out the quotation marks. So please distinguish a node u from the arc label u, and a kernel denoted by k from the label k of kernel edges.

The (symbolic) adjacency matrix of a graph comprises the information of the graph in compact form, using ε for denoting "no relation". Let the adjacency matrix of an example graph be given as

$$A(G_1) = \begin{pmatrix} p & l & \varepsilon \\ l & p & \varepsilon \\ k+d & k+d & p \end{pmatrix} \tag{14.15}$$

This graph represents a website consisting of three webpages 1, 2, and 3, corresponding to three HTML files. All webpages are of type p. Webpage 3 corresponds to the index page of the website, and is linked to pages 1 and 2 via a kernel arc and an additional downward arc. Pages 1 and 2 are connected by lateral edges. There are no upward arcs present in the graph (which would be unusual for an actual website).

The $+$ operator is only used in a formal way. That is, $k+d$ denotes a set consisting of two elements. This notation allows to represent multi-sets as, for instance, $k++d+d = k+2d$. This means that using adjacency matrices allows the representation of labeled multidigraphs. In order to arrive at a unique representation, we assume that the elements have been sorted lexicographically. The re-labeling procedure described in the following operates on this symbolic adjacency matrix by taking formal powers of it.

14.4.3.2 The Weisfeiler Iteration

We consider an arbitrary graph G together with its adjacency matrix $A(G)$. The following iteration produces a sequence of formal matrices corresponding to formal powers of the original adjacency matrix $A(G)$. The addition is assumed to be commutative, whereas the multiplication is non-commutative, but the usual distributive laws hold. The "+"-operator is used for constructing multi-sets, whereas the multiplication corresponds to string concatenation.

Definition 1 (Weisfeiler Iteration). *We define the Weisfeiler sequence of matrices* $\left(W^i(G)\right)_{i\geq 0}$ *as*

$$W^0(G) = A(G), W^{i+1}(G) = W^i(G) \cdot A(G).$$

For the matrix in Equation (14.15), we get $W^1(G_1) =$

$$\begin{pmatrix} (p)(p)+(\varepsilon)(k+d)+(l)(l) & (p)(l)+(l)(p)+(\varepsilon)(k+d) & (p)(\varepsilon)+(\varepsilon)(p)+(l)(\varepsilon) \\ (p)(l)+(l)(p)+(\varepsilon)(k+d) & (p)(p)+(\varepsilon)(k+d)+(l)(l) & (p)(\varepsilon)+(\varepsilon)(p)+(l)(\varepsilon) \\ (p)(k+d)+(k+d)(p)+(k+d)(l) & (p)(k+d)+(k+d)(p)+(k+d)(l) & (p)(p)+(k+d)(\varepsilon)+(k+d)(\varepsilon) \end{pmatrix}$$

In order to compute the sequence efficiently (i.e. in polynomial time), it is crucial to *not expand* the generated polynomials. The polynomials we get for iteration i are considered atomic in the next step. This can be achieved by storing the (sorted) polynomials in a dictionary and using their code for the next iteration. Note that it is also possible to treat ε as a kind of zero of the formal addition and multiplication, this way allowing the context to expand only via existing arcs and not via disconnected nodes. In this case, it is also reasonable to introduce inverse relations like k'. This is the approach we used in the experiments.

It can be shown that $W^i(G)[u,v]$ is a compact representation of the label sequences of all paths[5] in graph G that start at node u, terminate at node v and have a length of exactly i. Each such path results in a specific label sequence. For the example graph, we get the length 2 label sequences

$$(p)(l) + (l)(p) + (\varepsilon)(k+d) = pl + lp + \varepsilon k + \varepsilon d \qquad (14.16)$$

leading from node 1 to node 2. The path underlying εk is $(1,3,\varepsilon),(3,2,k)$, the one underlying εd is $(1,3,\varepsilon),(3,2,d)$, the one of pl is $(1,1,p),(1,2,l)$, and lp corresponds to $(1,2,l),(2,2,p)$. Using the Weisfeiler iteration, we arrive at compact representations of sets of label sequences that were used as patterns by INDIGO. These attributes will be called *context attributes* in what follows.

In the following definition, img $W^i(G)$ is used to denote the different entries (complex labels) of the i-th power of the adjacency matrix of G. In general, img denotes the set of values of a function. The adjacency matrix is hence considered a function of $\{1,\ldots,n\} \times \{1,\ldots,n\}$ to the set of complex labels of nodes and edges.

Definition 2 (Context Attributes). *Given a set of graphs, T, we define a sequence of attribute/feature sets*

[5] Here and in the following, we use the term *path* for connected sequences of arcs. Node labels are treated as labels of loops, i.e., particular arcs. Paths are allowed to contain cycles.

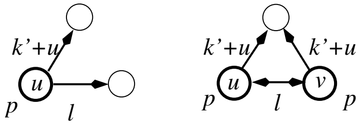


Fig. 14.3 The node context f_2 and the arc context f_4. Note that $(k'+u)$ and $(k+d)$ are inverse to each other and therefore only represented once. Note that u is used as a label for upward arcs and also for naming a node.

$$F_i(T) = \bigcup_{G \in T} img\, W^i(G)$$

with the corresponding feature value for $f \in F_i(T)$ defined as

$$\phi_f(G) = \left\| \left\{ (u,v) \mid 1 \leq u,v \leq n \wedge W^i(G)[u,v] = f \right\} \right\| .$$

For a context f from the set of all depth i contexts $F_i(T)$, the value of $\phi_f(G)$ corresponds to the number of nodes and edges, respectively, that have the label f.

In the example, after the insertion of inverse relations k' and $d' = u$,[6] the features $f_1 \in F_0(T)$ and $f_2 \in F_1(T)$ are context attributes characterizing nodes:

$$f_1 = p \tag{14.17}$$
$$f_2 = (p)(p) + (k'+u)(k+d) + (l)(l) \tag{14.18}$$

Among others, we have the arc attributes

$$f_3 = u \tag{14.19}$$
$$f_4 = (p)(l) + (l)(p) + (k'+u)(k+d). \tag{14.20}$$

f_2 and f_4 are depicted in Figure 14.3 in a graphical manner. In the edge attribute f_4, the term $(k'+u)(k+d)$ originates from the arc sequences $(1,3)(3,2)$. The triangular configuration results from the matrix multiplication in the Weisfeiler iteration, in which the entries are computed as $W^{i+1}(G)[u,v] = \sum_w W^i(G)[u,w] \cdot A(G)[w,v]$.

Definition 3 (Context Kernel). *Given a set of graphs, T, two graph G and G', and a sequence of non-negative weights w_i, the context kernel is defined as*

$$k_I(G,G') = \sum_{i=0}^{\infty} w_i \sum_{f \in F_i(T)} \phi_f(G)\phi_f(G') \, . \tag{14.21}$$

For computing the kernel k_I, we can employ the following weighting schemes:

1. $w_i = \lambda^i$ for some $0 \leq \lambda < 1$. In order to be able to compute this kernel, we need to set an upper limit D for the maximum context depth to be considered.

[6] Note that u is used as a symbolic arc label here. This is not to be confused with a node u.

2. $w_i = 1$ for $D_1 \leq i < D_2$ and $w_i = 0$ for and $i < D_1$ and $i \geq D_2$. This results in a mere concatenation of the feature vectors belonging to the specified range of context depths given by the integers D_1 and D_2.

Of course, other choices for $w_i \geq 0$ are possible as long the infinite sums used in the definition of k result in finite values. The following theorem holds.

Theorem 3. *k_I is a kernel.*

Proof: k_I is a sum of scalar products. □

14.4.3.3 The INDIGO Label Sequence Kernel

In order to derive the second INDIGO kernel, recall that $W^i(G)[u,v]$ corresponds to the set of label sequences of length i from paths from u to v. In order to define the label sequence kernel, we do not use whole context attributes as patterns, but rather the label sequences constituting them. The label sequences can be obtained by expanding the expressions in $W^i(G)$ by applying the usual laws of non-commutative multiplication and commutative summation, e.g., $((p)(l) + (l)(p) + (k{+}u)(\varepsilon)) = pl + lp + k\varepsilon + u\varepsilon$. This can be achieved by using a modified Weisfeiler iteration, in which the function e denotes the expansion process.

Definition 4 (Modified Weisfeiler Iteration). *We define the sequence of matrices $(M^i(G))_{i\geq 0}$ as*

$$M^0(G) = A(G), \qquad M^{i+1}(G) = e(M^i(G) \cdot A(G)).$$

Note that the entries in $M^{i+1}(G)$ represent multi-sets, in which each label sequence might occur more than once. For a multi-set c let $\#(l,c)$ denote the multiplicity of l in c. $\#(l,c) = 0$ holds exactly when l is not a member of c.

Definition 5 (Label Sequence Attributes). *Given a set of graphs, T, we define a sequence of attribute/feature sets*

$$L_i(T) = \bigcup_{G\in T} \left\{ \, l \mid \#(l,c) > 0 \wedge c \in img\, M^i(G) \right\}.$$

with the corresponding feature value for $l \in L_i(T)$ defined as

$$\phi_l(G) = \sum_{1\leq u,v\leq n} \#\big(l, M^i(G)[u,v]\big).$$

The expression $\#(l,c) > 0 \wedge c \in \mathrm{img} W^i(G)$ states that the label sequence l occurs in some level i context c of graph G. The expression $\#(l, M^i(G)[u,v])$ is the number with which l occurs in the respective entry of the expanded i-th power of $A(G)$.

Definition 6 (Label Sequence Kernel). *Given a set of graphs, T, two graphs G and G', and a sequence of weights $w_i \geq 0$, the label sequence kernel is defined as*

$$k_L(G,G') = \sum_{i=0}^{\infty} w_i \sum_{l \in L_i(T)} \phi_l(G)\phi_l(G') .$$

While the computation of $k_L(G,G')$ is quite straightforward, the number of label sequences l grows exponentially with i. This means that the computation of k_I is computationally intractable for large i. Using a small context depth, however, this approach could still be applied successfully for classifying chemical compounds with INDIGO [22].

Note that we only need the mappings of the *common* label sequences in order to compute $k_L(G,G')$. Moreover, we do not need the label sequences themselves but only the number of those shared between the graphs. These numbers can be obtained from the so-called *product graph* $X(G,G')$ [34, 18]. The nodes of the product graph correspond to pairs of nodes (u,u') from the original graphs G and G'. In the following definition, we directly specify its adjacency matrix. To this end, each node (u,u') is mapped to a natural number $(u-1)\cdot m+u'$, where m is the number of nodes of G'.

Definition 7 (Product Graph). *Let G and G' be graphs of degree n and m, respectively, and $A(G)$ and $A(G')$ their adjacency matrices. Then the (numerical) adjacency matrix X of the product graph is defined in the following manner:*

$$A(G)[u,v] = A(G')[u',v'] \rightarrow X[(u-1)\cdot m+u',(v-1)\cdot m+v'] = 1$$

and

$$A(G)[u,v] \neq A(G')[u',v'] \rightarrow X[(u-1)\cdot m+u',(v-1)\cdot m+v'] = 0 .$$

Note that the product has a size of nm. The index $(u-1)\cdot m+u'$ represents node (u,u') in the product graph, whereas $(v-1)\cdot m+v'$ stands for node (v,v'). X has a value of 1 just when the (arc or node) label of (u,v) in G corresponds to that of (u',v') in G'. The 1-norm of $X(G,G')$ thus corresponds to the number of mappings of identical label sequences (length 1) in the two graphs.

It can be shown that the 1-norm of the i-th power, $|X^i(G_1,G_2)|_1$, corresponds to the number of mappings of length i paths with identical label sequences in G and G' [34, 18] (note that the underlying paths do not have to be isomorphic). We thus have the following theorem.

Theorem 4. *The INDIGO label sequence kernel can be computed as*

$$k_L(G,G') = \sum_{i=0}^{\infty} w_i \, |X^i(G,G')|_1 .$$

With an appropriate choice of the weights w_i, the kernel k_L can now be computed in polynomial time. However, the adjacency matrix of the product graph might still become quite large in practice.

Although the formulation via the product graph is quite elegant, we have to store the adjacency matrix and its powers in order to compute the kernel. For the corpus at hand, we were not able to store the adjacency matrix of the product graph, for instance, when computing the norm of a graph with more than 1000 nodes, which leads to 1 Million nodes in the product graph, and an adjacency matrix with 10^{12} entries. We also tried sparse matrix representations, which slowed down the multiplication process a lot. Moreover, for some pairs of graphs, the powers of $A(G)$ become less and less sparse (mainly due to the lateral arcs present in a GT). In the end it turned out that the INDIGO label sequence kernel is much more efficient to use and it is also more easy to apply optimizations that discard label sequences, which are not considered useful as features (e.g., repetitions of the same node label in a sequence).

14.5 Experiments

In this section we present the learning results for a corpus of websites, which is described in the next section.

14.5.1 The Corpus

We exemplify the learning of graph structures by means of webgenres [48]. More specifically, we analyze a corpus of three genres, that is, *personal academic homepages*, *conference websites* and *project websites* (see Table 14.1). Each website is represented as a directed generalized tree whose vertices denote pages and whose arcs denote hyperlinks between these pages [57]. The vertices are uniquely labeled by the URL of the corresponding page, while the arcs are typified in terms of directed generalized trees (see Definition 1).

Our central classification hypothesis says that webgenres can be distinguished by the hypertextual structure of their instances. In this sense, we expect that the membership of a website to a certain hypertext type is predictable by the topology of its constituent pages and their hyperlinks. In the present chapter, we consider two classification hypotheses based on this general assumption.

1. Firstly, we hypothesize that this structure-oriented classification can be performed by a certain class of graph kernels as introduced in Section 14.4.
2. Secondly, we hypothesize that this structural classification can be performed by means of *Quantitative Structure Analysis* (QSA) as described in Section 14.3.

Both of these classification hypotheses are motivated by [44] who successfully classify the webgenres considered here by means of a two-level *bag-of-structures* model. This approach integrates two levels of learning as it restricts the genre-related classification of webpages by the genre-related classification of their constituents where pages are represented as bags of subgeneric constituents – in this sense, it is called a bag-of-structures model. While this approach shows that structure-based classifications are in principle possible even with an F-score above .9, it does so without additionally exploring hyperlinks. In this chapter, we focus on

Table 14.1 The webgenre corpus of websites of three hypertext types. The minimum, median and maximum value of the corresponding variable are displayed.

hypertext type	number	order	size
conference websites	1,086	(10; 28; 1,130)	(9; 102; 7,547)
academic homepages	177	(10; 35; 1,258)	(9; 52; 2,145)
project websites	35	(10; 31; 242)	(23; 133; 1,876)

this specific informational resource. That is, we explore hypertext structure as the sole resource of classification.

As we deal with three target categories, any instance to be classified needs to be structured in a way that is sufficient to attribute its category membership. Thus, a website of less than 10 pages hardly provides structural information in order to decide on its webgenre. Suppose, for example, that we deal with a set of unlabeled digraphs each of four vertices and 3 arcs. In this case, we could hardly distinguish three classes as there are likely equally structured graphs that belong to different classes. Thus, a lower bound of the order of graphs $\gg 3$ is indispensable. In the present study, we process websites of at least 10 pages. All these websites were downloaded and managed by means of the HyGraph system [28, 57].

14.5.2 Quantitative Structure Analysis

In this Section, we experiment with classifying websites per webgenre, where the sites are represented as generalized digraphs (see Section 14.5.1). Starting from the corpus described in Section 14.5.1, we evaluate our twofold cluster hypothesis about the effectiveness of graph-kernel- and QSA-based classifiers.

In order to evaluate our candidate classifiers, we compute the well-known **F**-measure, that is, the harmonic mean of recall and precision: let $\mathbb{C} = \{C_1, \ldots, C_n\}$ be the set of target categories to be learnt and $X = \{x_1, \ldots, x_m\}$ be a non-empty training corpus of objects each of which belongs to exactly one category in $\mathbb{C}$. Further, let $\mathbb{L} = \{L_1, \ldots, L_n\}$ be a classification of X according to $\mathbb{C}$ such that $L_i \neq \emptyset$, $i \in \{1, \ldots, n\}$, is the non-empty subset of X of true instances of $C_i \in \mathbb{C}$ where $|\mathbb{C}| = |\mathbb{L}|$ and $X = L_1 \cup \ldots \cup L_n$. Now, let $\mathbb{P} = \{P_1, \ldots, P_n\}$ be a computed partitioning of X with respect to $\mathbb{C}$. Without loss of generality, we call the elements of $\mathbb{P}$ *clusters*. Then, we compute the recall, precision and F-score of any $P_i \in \mathbb{P}$ in relation to a corresponding $L_j \in \mathbb{L}$ and finally the F-measure of $\mathbb{P}$ with respect to $\mathbb{L}$ as follows [30]:

$$\mathrm{P}(P_i,L_j) = \mathrm{Precision}(P_i,L_j) = \begin{cases} \frac{|P_i \cap L_j|}{|P_i|} & : \quad |P_i| > 0 \\ 0 & : \quad \mathit{else} \end{cases} \tag{14.22}$$

$$\mathrm{R}(P_i,L_j) = \mathrm{Recall}(P_i,L_j) = \mathrm{Precision}(L_j,P_i) \tag{14.23}$$

$$F(P_i,L_j) = \begin{cases} \frac{2\mathrm{P}(P_i,L_j)\mathrm{R}(P_i,L_j)}{\mathrm{P}(P_i,L_j)+\mathrm{R}(P_i,L_j)} : \mathrm{P}(P_i,L_j)+\mathrm{R}(P_i,L_j)>0 \\ 0 : \mathit{else} \end{cases} \tag{14.24}$$

$$F(\mathbb{P},\mathbb{L}) = \sum_{j=1}^{n} \frac{|L_j|}{|X|} \max_{P_i \in \mathbb{P}} F(P_i,L_j) \tag{14.25}$$

Note that by applying Equation 14.25, it may occur that different categories L_i, L_j are mapped onto the same cluster P_k. Suppose, for example, three categories C_1, C_2, C_3 such that $L_1 \subset P_1$, $L_2 \subset P_1$, $P_2 \cup P_3 = L_3$, and $|P_3| \geq |P_2|$. In this case, $P_1 = \arg\max_{P_i \in \mathbb{P}} F(P_i, L_1)$, $P_1 = \arg\max_{P_i \in \mathbb{P}} F(P_i, L_2)$, and $P_3 = \arg\max_{P_i \in \mathbb{P}} F(P_i, L_3)$. Thus, we do *not* get a one-to-one mapping of categories and clusters.

From the point of view of categorization, this scenario is problematic: as we assume that any object in X belongs to exactly one category in $\mathbb{C}$, we demand that any candidate classifier unambiguously decides the category membership of any object in X. We call this scenario the *decider scenario*. It sets the F-measure of a classification $\mathbb{P}$ to zero in the case that it does not result in a one-to-one mapping of clusters and categories in the sense of Equation 14.25. Obviously, the decider scenario is more demanding as it constrains the set of candidate classifiers in the latter sense. In general, it should be more difficult to perform under the decider scenario than without this restriction.

As a basis of comparison, we compute four baselines. Three of them randomly assign objects to the target categories according to the decider scenario: the random baseline called *known-partition* has information about the number of training instances of the target categories. By knowing that there are 1,086 conference websites, 177 homepages and 35 project websites, it randomly generates three subsets of the same cardinalities in order to compute the F-measure based on Equation 14.25. Under this regime, it may occur that the target categories are not mapped to their random analogue of equal cardinality (see above). Therefore, we additionally compute the random baseline called *named partition*, which presupposes this one-to-one-mapping for computing the F-measure. As shown in Table 14.2 and 14.3, the values of the latter baseline correspond to the values of the former one. This results from the dominance of conference websites compared to academic homepages and project websites. Each random baseline is computed 1,000 times so that 14.2 and 14.3 show the corresponding average F-measures.

Obviously, it is a source of favoring that both baselines are informed about the cardinalities of the target categories. Therefore, we compute a third random baseline, called *equi-partition*, that assumes equally sized target categories. As expected by the skewed distribution of category size, this baseline decays enormously – thus,

Table 14.2 Results of learning three webgenres subject to the decider scenario.

model	Setting	F(CW)	F(PA)	F(PW)	F-measure
QSA[correlation]	11 / 77	.93715	.55285	.18605	**.86449**
QSA[correlation]	12 / 77	.93715	.55285	.18605	**.86449**
QSA[correlation]	10 / 77	.93715	.54918	.17778	.86377
QSA[Mahalanobis]	8 / 77	.93483	.525	.051282	.85512
QSA[Mahalanobis]	9 / 77	.93276	.5	.05	.84994
QSA[Mahalanobis]	9 / 77	.93236	.49362	.05	.84874
catchAll(CW)		$\boxed{.911074}$	0	0	$\boxed{.76227}$
catchAll(PA)		0	$\boxed{.24}$	0	.032727
catchAll(PW)		0	0	$\boxed{.052513}$	.001416
baseline *known part.*					.71918
baseline *named part.*					.71911
baseline *equi-part.*					.442872

Table 14.3 Results of classifying personal academic homepages and conference websites according to the decider scenario.

model	Setting	F(CW)	F(PA)	F-measure
QSA[correlation]	12 / 77	.95175	.55285	.89585
QSA[correlation]	15 / 77	.9518	.54918	.89537
QSA[Mahalanobis]	11 / 77	.94954	.53441	.89136
QSA[Mahalanobis]	10 / 77	.94903	.536	.89115
catchAll(cw)		$\boxed{.924649}$	0	$\boxed{.795066}$
catchAll(PA)		0	$\boxed{.245833}$	.034452
baseline known partition				.75931
baseline equi-partition				.584609

we report it for information only.[7] This skewness is the starting point for computing a fourth baseline, which, in the present study, generates the highest baseline values. According to the dominance of conference websites, it simply assigns each object to this webgenre. We denote this baseline by the functional catchAll(X), which takes the "catching" webgenre as its single parameter, that is, CW (*conference websites*), PA (*personal academic homepages*), or PW (*project websites*). catchAll(CW) results in the highest baseline F-measure of .76227, which even

[7] Note that the baseline that randomly guesses the cardinalities of the target classes in order to randomly select their members basically converges to the equi-partition scenario if being repeated infinitely.

outperforms the known-partition baseline (see Table 14.2).[8] Due to their small training set cardinalities, `catchAll(PA)` and `catchAll(PW)` result in very small F-measure values (see Table 14.2). In any event, the values of `catchAll(CW)`, `catchAll(PA)` and `catchAll(PW)` can be used as category-specific baselines. Thus, it seems natural to demand that a candidate classifier performs above these baselines of about .91 (in the case of conference websites), .24 (in the case of academic homepages), and .053 (in the case of project websites). Otherwise it can hardly be called a classifier of these webgenres when considered in isolation.

14.5.2.1 Discussion

By looking at Table 14.2, the first impression is that QSA solves the classification task considered here, though not very well. Apparently, .86 is not a very high F-measure value since we only deal with three target classes. However, if we study the F-measure values more closely, it is evident that QSA has a classificatory power in the area of structure-based webgenre learning. The reason is that on the one hand, the micro-averaged F-measure outperforms the best performing baseline by at least 10%. If we look at the second best performing baseline, this surplus is even around 13%. This assessment is confirmed when looking at the category-specific F-measure values: for each of the target categories considered here, the best performing instance of QSA outperforms the corresponding baseline value of the F-measure: $0.937 > 0.91$ (CW), $0.55 > 0.24$ (PA) and $0.186 > 0.053$ (PW). Thus, there is a classificatory gain of applying QSA, which cannot be reconstructed by any of the random baselines considered here. However, the absolute value of the F-measure is rather low for the category of personal academic homepages and very low for the category of project websites. Thus, one should not overestimate our findings as they do not show a good classifier but simply the possibility of structural classifications of websites. In other words: there is an impact of structure-oriented webgenre classification as measured by QSA, but this impact is not very high. Note that using the Mahalanobis distance (instead of the correlation coefficient) produces similar results.

One reason for the smaller impact being observed is certainly given by the highly skewed distribution of the number of instances of the target categories. In order to check this, Table 14.3 shows the result of classifying conference websites and personal academic homepages only. At a first glance, we see that the F-measure values go up to ca. 90% thereby retaining the difference to the best performing baseline. In any event, the general picture is retained that the classification could be better. Thus, we conclude that QSA proves a potential of structural classification, however to a minor degree.

One reason for our findings is certainly the comparatively small set of training samples of personal academic homepages and project websites in relation to conference websites. Another reason might be the skewed distribution of the size of the

[8] Note that in this scenario, the F-score of PA and CW is 0. The reason is that for these categories, recall is 0. Thus, according to Equation 14.24, the F-score of these categories is 0.

Table 14.4 Results for the different kernel approaches: we highlighted the best performance values in each CV-column.

	$F(CW)$		$F(PA)$		$F(PW)$		Acc		F_{macro}		F_{avg}	
	train	CV	train	CV	train	CV	train	CV	train	CV	train	CV
SimTree	0.998	0.911	0.989	**0.422**	0.986	0.154	0.997	0.837	0.991	0.435	0.997	0.782
SoftTree	0.975	0.912	0.823	0.408	0.959	0.194	0.956	0.839	0.919	0.437	0.954	0.779
SimGT	0.998	0.911	0.991	0.416	0.986	0.182	0.997	0.837	0.991	0.437	0.997	0.787
SoftGT	0.994	0.911	0.963	0.418	0.986	0.160	0.989	0.837	0.978	0.435	0.989	0.787
Context	0.998	0.912	0.989	0.407	0.986	**0.207**	0.996	**0.841**	0.990	**0.464**	0.997	0.785
Label S.	0.997	**0.913**	0.989	0.401	0.986	0.190	0.996	**0.841**	0.990	0.456	0.996	**0.794**

websites (i.e., the order of the corresponding graphs), which ranges from websites of only 10 pages to websites that contain more than 1,000 different pages (as in the case of conference website and personal academic homepages). In this sense, it might be useful to partition the training data so that the different bins are more balanced in terms of the size of their elements.

A third reason for the observed results is certainly that the data model used here to represent websites needs to be complemented by more elaborated feature models in order to capture the characteristics of webgenres. As shown by [44], it is not necessary that these additional features are content-based. The reason is that websites as considered here consist of multiple webpages whose internal structure is informative about the membership to webgenres. Thus, a combined mode of page-*internal* structures (as mapped by Mehler and Waltinger) and page-*external* structures (as analyzed here) could be a way to get better classifiers. In any event, the combination of content- and structure-related features remains an option to be considered by future work.

14.5.3 Kernel Methods

QSA and the kernel approaches follow different philosophies. The kernel approaches are used for classical supervised learning, in which a SVM is determined, which maximizes a certain measure of performance, e.g., the F-measure. In our experiments, we measured both the performance for the training set and that for the test set by employing 10-fold cross validation. QSA uses the class information in the feature selection step and for evaluating the quality of the learned categories by utilizing unsupervised techniques for finding them. On the one hand, this means that QSA can be considered "handicapped" during training. On the other hand, the evaluation is done on the complete set of input objects as there is no distinction of a training set and test set. Therefore, the performance of QSA can be expected to lie between the usually better training set performance and the usually worse cross validation performance of the kernel methods. This turned out to be the case in our experiments, and we therefore did not undertake a direct comparison.

For the experiments, we used the LibSVM [8], which is able to handle multi-class problems. Table 14.4 shows the results obtained for the webgraph corpus.

The values $F(CV)$, $F(PE)$, $F(PR)$ correspond to the F-measures for the single classes. Acc is the accuracy, i.e., the proportion of correctly classified examples. F_{macro} corresponds to the standard macro-averaged F-measures, in which each class is weighted with $\frac{1}{3}$. The value of F_{avg} corresponds to the F-measure according to Equation (14.25), but with the sets determined by the actual and predicted classes in the usual manner.

14.5.3.1 Discussion

The main problem for learning is that the data set is highly imbalanced. This means that it is always a good guess that a website is a "conference website" yielding high performance values for this class. We also tried weighting the classes according to their prior probabilities with some, but limited success: A larger weight for class "project webpages" can only partly compensate for the lack of information about this class. We also applied kernel normalization and input scaling with no consistent behavior in terms of performance.

Looking at Table 14.4, we find that the context kernel performs best with respect to the macro-averaged F-measure (cross validation), whereas the INDIGO label sequence kernel performs best with respect to the averaged F-measure (cross validation). The differences to the other approaches are small, though. As a general observation, the values for the training set were excellent, while the CV-estimated performance values are substantially worse than the training set performance. For class CW, the estimated F-measures are consistently high, but barely exceed the baseline given by $catch\_all(CW)$ in Table 14.2. Learning mainly took place for the two smaller classes, with a relatively hight improvement compared to the respective baselines.

Although most of the differences are not statistically significant, there is a tendency that the INDIGO kernels outperform the tree and GT kernels. We used grid search for parameter optimization. Since the "Soft"-versions of the tree and GT-kernel are much more expensive to compute, we could only cover a smaller range of parameter combinations. In our view, this explains the worse performance of the "Soft"-version with respect to the "Sim"-Versions.

Since the dataset is so difficult, it is not yet possible to say anything conclusive about the relative performance of the different kernels. Regarding computational complexity, however, there is a clear distinction. The INDIGO context kernel is the most efficient, and it is also possible to attain a high depth of context (see the Weisfeiler iteration in Def. 1). However, since higher level context attributes tend to comprise a lot of structural information, they become very graph-specific and have only little generalization power. Label sequence attributes, in contrast, tend to be shared among different graphs, but their number tends to grow exponentially with the level of the Weisfeiler iteration. In our experiments, we therefore had to rule out certain label sequences in order to limit the length of the feature vectors.

For SimTree, SoftTree, SimGT, and SoftGT, we use the “pre-computed kernel” kernel feature of the LibSVM. Still the kernel computations tend to be more relatively complex for the “Soft” variants. SimGT and SoftGT were the most efficient to compute.

Note that all classes of kernels result in a loss of structural information, because they only approximate the trees, graphs, and GTs (respectively) in the training set. A kernel based on a full structural characterization is possible, but computationally intractable [see, e.g., 18].

14.6 Conclusion

In this chapter, we presented a series of kernel approaches suitable for the classification of trees, graphs, and generalized trees. As one of the simplest approaches, we used the Soft Tree Kernel, which was originally developed for the classification of trees. It comprises a variant of the well-known parse tree kernel combined with a modified set kernel for the children of a node. The structural information encoded in the reflexive, lateral, downward and upward arcs is taken into account via node attributes indicating the presence of such arcs.

Replacing the RBF kernel for node positions with the identity kernel results in the Simple Tree Kernel, which changes the complexity of comparing child sequences from quadratic to linear. Being given graphs with more than 1000 nodes, this turned out to be crucial for learning classifiers efficiently.

The INDIGO context and label sequence kernels both fall in the class of graph kernels that use a transformational approach. Node and arc contexts characterize the relational context of the respective object. The context depth is increased based on the so-called Weisfeiler iteration. The attribute value corresponds to the number of nodes and arcs, respectively, of the corresponding type.

We demonstrated that it is possible to compute the label sequences in a graph from these contexts, albeit changing the complexity (and the number of attributes) from polynomial to exponential. The resulting label sequence kernel is related to a version of the random walk kernel, which can be computed from the product graph of the input graph. For the corpus at hand, however, it was not possible to use this approach at all, for the size of the product graph just gets too large. Surprisingly, enumerating (and filtering) the label sequences turned out to be more practical, partly because the input structures are tree-like.

The Soft GT kernel, which is based on the Soft Tree Kernel, also implicitly transforms GTs into trees. Reflexive arcs are handled using a simple node kernel, which is plugged into the SoftGTK. In contrast to the SoftTK, downward links and upward links are treated like kernel arcs. The children with respect to these four types of arcs are handled separately during kernel computation. Lateral arcs are also included into the computation of the kernel, but the algorithm follows each lateral arc only once in each recursion.

Concept learning was possible for the web corpus containing three different classes of web graphs. Due to the extremely imbalanced nature of the training set, the learning problem turned out to be very challenging. We hope to be able to better clarify their performance in future experiments. A clear result could be obtained in terms of computational complexity: only the "simple" versions of the kernels, as well as the INDIGO context and label sequences kernels really scale to larger graphs.

Comparable to the tree and graph kernels considered here, *Quantitative Structure Analysis* (QSA) generates classification results that are in need of improvement: on the one hand, we observe F-scores above the corresponding baselines. On the other hand, these F-scores leave plenty room for optimization. Thus, our findings are in support of a structure-based classification of websites into webgenres. At least in part, we have shown that one can identify the genre of a website by examining its hyperlink-based structure. However, our F-scores question the reliability of such *purely* hyperlink-based classifiers.

With regard to this finding, both approaches QSA and kernel methods are not really different. Their difference is rather a methodological one: while the kernel methods studied here explore similar substructures to assess class membership of websites, QSA is based on quantitative graph models that focus on local (e.g., vertex clustering-related) or global (e.g., graph centrality-based) characteristics in order to derive feature vectors of websites that are input to classifying them. In other words, kernel methods assess graph similarity more directly, while QSA builds on numerical representations of the topology of graphs to asses their similarity. Note that using SVMs in the case of kernel methods in contrast to cluster analysis in the case of QSA does not manifest a basic difference of these methods. The reason is that QSA can be combined with SVMs as well [cf. 45]. Thus, our comparison of QSA and kernel methods hints at the fact that the classification problem studied here is equally treated by supervised and unsupervised methods.

As a consequence of these findings, we are convinced that additional topological features will not basically improve the classification of websites when solely exploring their hyperlink-based structure. Rather, we have to think of additional resources of hypertext formation. In this sense, we plan to build structural classifiers, which integrate document internal, that is, DOM-based structures with document external, that is, hyperlink-based structures. In this way, we conceive an integrated model of structure formation in the web that includes page internal and page external features. Finally, we plan to explore content- and structure-related features in order to arrive at an integrated classifier of the topic and function of websites. This will be part of future work for which the present study has paved the way in regard to hyperlink structure.

References

[1] Aiolli, F., Martino, G.D.S., Sperduti, A., Moschitti, A.: Efficient kernel-based learning for trees. In: CIDM, pp. 308–315. IEEE, New York (2007)

[2] Biber, D.: Dimensions of Register Variation: A Cross-Linguistic Comparison. Cambridge University Press, Cambridge (1995)

[3] Blanchard, P., Volchenkov, D.: Mathematical Analysis of Urban Spatial Networks. Springer, Berlin (2009)

[4] Bloehdorn, S., Moschitti, A.: Combined syntactic and semanitc kernels for text classification. In: Proceedings of the 29th European Conference on Information Retrieval, Rome, Italy (2007)

[5] Bollobás, B., Riordan, O.M.: Mathematical results on scale-free random graphs. In: Bornholdt, S., Schuster, H.G. (eds.) Handbook of Graphs and Networks. From the Genome to the Internet, pp. 1–34. Wiley-VCH, Weinheim (2003)

[6] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group (1984)

[7] Chakrabarti, S.: Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, San Francisco (2002), http://www.cse.iitb.ac.in/~soumen/mining-the-web/

[8] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/cjlin/libsvm

[9] Collins, M., Duffy, N.: Convolution kernels for natural language. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NIPS, pp. 625–632. MIT Press, Cambridge (2001)

[10] Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In: ACL, pp. 263–270 (2002)

[11] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods). Cambridge University Press, Cambridge (2000)

[12] Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: ACL 2004: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, p. 423 (2004), http://www.cs.umass.edu/˜culotta/pubs/culotta04dependency.pdf, doi:http://dx.doi.org/10.3115/1218955.1219009

[13] Cumby, C., Roth, D.: On kernel methods for relational learning. In: Fawcett, T., Mishra, N. (eds.) Proceedings of the Twentieth International Conference on Machine Learning, pp. 107–115. AAAI Press, Menlo Park (2003)

[14] Dehmer, M.: Information processing in complex networks: Graph entropy and information functionals. Applied Mathematics and Computation 201, 82–94 (2008)

[15] Dehmer, M., Emmert-Streib, F., Mehler, A., Kilian, J.: Measuring the structural similarity of web-based documents: A novel approach. International Journal of Computational Intelligence 3(1), 1–7 (2006)

[16] Dehmer, M., Mehler, A., Emmert-Streib, F.: Graph-theoretical characterizations of generalized trees. In: Proceedings of the 2007 International Conference on Machine Learning: Models, Technologies & Applications (MLMTA 2007), Las Vegas, June 25-28, pp. 113–117 (2007)

[17] Foscarini, F., Kim, Y., Lee, C.A., Mehler, A., Oliver, G., Ross, S.: On the notion of genre in digital preservation. In: Chanod, J.P., Dobreva, M., Rauber, A., Ross, S. (eds.) Proceedings of the Dagstuhl Seminar 10291 on Automation in Digital Preservation, July 18–23, Dagstuhl Seminar Proceedings. Leibniz Center for Informatics, Schloss Dagstuhl (2010)

[18] Gärtner, T.: A survey of kernels for structured data. SIGKDD Explorations 5(2), 49–58 (2003)

[19] Gärtner, T.: A survey of kernels for structured data. SIGKDD Explor. Newsl. 5(1), 49–58 (2003), doi: http://doi.acm.org/10.1145/959242.959248

[20] Gärtner, T., Flach, P.A., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Proceedings of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop (2003)

[21] Gärtner, T., Lloyd, J.W., Flach, P.A.: Kernels and distances for structured data. Machine Learning 57(3), 205–232 (2004)

[22] Geibel, P.: Induktion von merkmalsbasierten und logische Klassifikatoren für relationale Strukturen. Infix-Verlag (1999)

[23] Geibel, P., Wysotzki, F.: Induction of Context Dependent Concepts. In: De Raedt, L. (ed.) Proceedings of the 5th International Workshop on Inductive Logic Programming, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, pp. 323–336 (1995)

[24] Geibel, P., Wysotzki, F.: Learning relational concepts with decision trees. In: Saitta, L. (ed.) Machine Learning: Proceedings of the Thirteenth International Conference, pp. 166–174. Morgan Kaufmann Publishers, San Francisco (1996)

[25] Geibel, P., Wysotzki, F.: Relational learning with decision trees. In: Wahlster, W. (ed.) Proceedings of the 12th European Conference on Artificial Intelligence, pp. 428–432. J. Wiley and Sons, Ltd, Chichester (1996)

[26] Geibel, P., Jain, B.J., Wysotzki, F.: Combining recurrent neural networks and support vector machines for structural pattern recognition. Neurocomputing 64, 63–105 (2005)

[27] Geibel, P., Pustylnikov, O., Mehler, A., Gust, H., Kühnberger, K.-U.: Classification of documents based on the structure of their DOM trees. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 779–788. Springer, Heidelberg (2008)

[28] Gleim, R.: HyGraph: Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertexte. In: Fisseni, B., Schmitz, H.C., Schröder, B., Wagner, P. (eds) Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Frühjahrstagung 2005, 10. März – 01, Universität Bonn, Lang, Frankfurt a. M., pp. 42–53 (April 2005)

[29] Haussler, D.: Convolution Kernels on Discrete Structure. Tech. Rep. UCSC-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA (1999)

[30] Hotho, A., Nürnberger, A., Paaß, G.: A Brief Survey of Text Mining. Journal for Language Technology and Computational Linguistics (JLCL) 20(1), 19–62 (2005)

[31] Hunt, E.B., Marin, J., Stone, P.J.: Experiments in Induction. Academic Press, London (1966)

[32] Joachims, T.: Learning to classify text using support vector machines. Kluwer, Boston (2002)

[33] Kashima, H., Koyanagi, T.: Kernels for semi-structured data. In: Sammut, C., Hoffmann, A.G. (eds.) ICML, pp. 291–298. Morgan Kaufmann, San Francisco (2002)

[34] Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Fawcett, T., Mishra, N. (eds.) Proceedings of the Twentieth International Conference on Machine Learning, pp. 321–328. AAAI Press, Menlo Park (2003)

[35] Kemp, C., Tenenbaum, J.B.: The discovery of structural form. Proceedings of the National Academy of Sciences 105(31), 10,687–10,692 (2008)

[36] Kersting, K., Gärtner, T.: Fisher kernels for logical sequences. In: ECML, pp. 205–216 (2004)

[37] Kondor, R.I., Shervashidze, N., Borgwardt, K.M.: The graphlet spectrum. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML, ACM International Conference Proceeding Series, vol. 382, p. 67. ACM, New York (2009)

[38] Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge (2000)

[39] Mehler, A.: Generalized shortest paths trees: A novel graph class applied to semiotic networks. In: Dehmer, M., Emmert-Streib, F. (eds.) Analysis of Complex Networks: From Biology to Linguistics, pp. 175–220. Wiley-VCH, Weinheim (2009)

[40] Mehler, A.: Minimum spanning Markovian trees: Introducing context-sensitivity into the generation of spanning trees. In: Dehmer, M. (ed.) Structural Analysis of Complex Networks. Birkhäuser Publishing, Basel (2009)

[41] Mehler, A.: A quantitative graph model of social ontologies by example of Wikipedia. In: Dehmer, M., Emmert-Streib, F., Mehler, A. (eds.) Towards an Information Theory of Complex Networks: Statistical Methods and Applications. Birkhäuser, Basel (2010)

[42] Mehler, A.: Structure formation in the web. A graph-theoretical model of hypertext types. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, Text, Speech and Language Technology, pp. 225–247. Springer, Dordrecht (2010)

[43] Mehler, A., Lücking, A.: A structural model of semiotic alignment: The classification of multimodal ensembles as a novel machine learning task. In: Proceedings of IEEE Africon 2009, September 23-25. IEEE, Nairobi (2009)

[44] Mehler, A., Waltinger, U.: Integrating content and structure learning: A model of hypertext zoning and sounding. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modeling, Learning and Processing of Text Technological Data Structures. SCI. Springer, Berlin (2010)

[45] Mehler, A., Geibel, P., Pustylnikov, O.: Structural classifiers of text types: Towards a novel model of text representation. Journal for Language Technology and Computational Linguistics (JLCL) 22(2), 51–66 (2007)

[46] Mehler, A., Waltinger, U., Wegner, A.: A formal text representation model based on lexical chaining. In: Proceedings of the KI 2007 Workshop on Learning from Non-Vectorial Data (LNVD 2007), September 10, Osnabrück, Universität Osnabrück, Osnabrück, pp. 17–26 (2007)

[47] Mehler, A., Pustylnikov, O., Diewald, N.: Geography of social ontologies: Testing a variant of the Sapir-Whorf Hypothesis in the context of Wikipedia. Computer Speech and Language (2010), doi:10.1016/j.csl.2010.05.006

[48] Mehler, A., Sharoff, S., Santini, M. (eds.): Genres on the Web: Computational Models and Empirical Studies. Springer, Dordrecht (2010)

[49] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Alon, D.C.U.: Network motifs: simple building blocks of complex networks. Science 298(5594), 824–827 (2002)

[50] Moschitti, A.: A study on convolution kernels for shallow statistic parsing. In: ACL, pp. 335–342 (2004)

[51] Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 318–329. Springer, Heidelberg (2006)

[52] Muggleton, S., Lodhi, H., Amini, A., Sternberg, M.J.E.: Support vector inductive logic programming. In: Hoffmann, A., Motoda, H., Scheffer, T. (eds.) DS 2005. LNCS (LNAI), vol. 3735, pp. 163–175. Springer, Heidelberg (2005)

[53] Power, R., Scott, D., Bouayad-Agha, N.: Document structure. Computational Linguistics 29(2), 211–260 (2003)

[54] Pustylnikov, O., Mehler, A.: Structural differentiae of text types. A quantitative model. In: Proceedings of the 31st Annual Conference of the German Classification Society on Data Analysis, Machine Learning, and Applications (GfKl), pp. 655–662 (2007)

[55] Quinlan, J.: Induction of Decision Trees. Machine Learning 1(1), 82–106 (1986)

[56] Rehm, G.: Towards automatic web genre identification – a corpus-based approach in the domain of academia by example of the academic's personal homepage. In: Proc. of the Hawaii Internat. Conf. on System Sciences (2002)

[57] Rehm, G., Santini, M., Mehler, A., Braslavski, P., Gleim, R., Stubbe, A., Symonenko, S., Tavosanis, M., Vidulin, V.: Towards a reference corpus of web genres for the evaluation of genre identification systems. In: Proceedings of LREC 2008, Marrakech, Morocco (2008)

[58] Santini, M.: Cross-testing a genre classification model for the web. In: [48] (2010)

[59] Santini, M., Mehler, A., Sharoff, S.: Riding the rough waves of genre on the web: Concepts and research questions. In: [48], pp. 3–32 (2010)

[60] Saunders, S.: Improved shortest path algorithms for nearly acyclic graphs. PhD thesis, University of Canterbury, Computer Science (2004)

[61] Schoelkopf, B., Smola, A.J.: Learning with Kernels. The MIT Press, Cambridge (2002)

[62] Sharoff, S.: In the garden and in the jungle. Comparing genres in the BNC and Internet. In: [48] (2010)

[63] Smola, A., Kondor, R.: Kernels and regularization on graphs. In: Schölkopf, B., Warmuth, M. (eds.) Proceedings of the Annual Conference on Computational Learning Theory and Kernel Workshop. LNCS. Springer, Heidelberg (2003)

[64] Stubbe, A., Ringlstetter, C., Goebel, R.: Elements of a learning interface for genre qualified search. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 791–797. Springer, Heidelberg (2007)

[65] Unger, S., Wysotzki, F.: Lernfähige Klassifizierungssysteme (Classifier Systems that are able to Learn). Akademie-Verlag, Berlin (1981)

[66] Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

[67] Weisfeiler, B.: On Construction and Identification of Graphs. No. 558 in Lecture Notes in Mathematics. Springer, Berlin (1976)

[68] Wysotzki, F., Kolbe, W., Selbig, J.: Concept Learning by Structured Examples - An Algebraic Approach. In: Proceedings of the Seventh IJCAI (1981)

[69] Zhang, D., Mao, R.: Extracting community structure features for hypertext classification. In: Pichappan, P., Abraham, A. (eds.) ICDIM, pp. 436–441. IEEE, Los Alamitos (2008)

Chapter 15
Integrating Content and Structure Learning: A Model of Hypertext Zoning and Sounding

Alexander Mehler and Ulli Waltinger

Abstract. The *bag-of-words* model is accepted as the first choice when it comes to representing the content of web documents. It benefits from a low time complexity, but this comes at the cost of ignoring document structure. Obviously, there is a trade-off between the range of document modeling and its computational complexity. In this chapter, we present a model of content and structure learning that tackles this trade-off with a focus on delimiting documents as instances of webgenres. We present and evaluate a two-level algorithm of *hypertext zoning* that integrates the genre-related classification of web documents with their segmentation. In addition, we present an algorithm of *hypertext sounding* with respect to the thematic demarcation of web documents.

15.1 Introduction

With the advent of the web, a range of linguistic concepts became significant in research on hypertexts. This includes the multifaceted notion of *genre* that supports the classification of textual units with regard to their (pragmatic) function in addition to their (semantic) content [17, 8, 25]. Webgenre modeling applies this notion to web documents (e.g., webpages or websites) [40, 43, 48] by distinguishing a range of functionally demarcated document types that are ideally orthogonal to topic categories (for a recent overview of webgenre modeling see Mehler et al 34).

Alexander Mehler
Computer Science and Informatics, Goethe-Universität Frankfurt, Senckenberganlage 31,
D-60325 Frankfurt am Main, Germany
e-mail: Mehler@em.uni-frankfurt.de

Ulli Waltinger
Faculty of Technology, Bielefeld University, Universitätsstraße 25,
D-33615 Bielefeld, Germany
e-mail: Ulli_Marc.Waltinger@uni-bielefeld.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 299–329.
springerlink.com

The machine learning of such document types, that is, *webgenre learning*, is primarily conceived as the task of automatically mapping web *pages* onto genre *labels* [cf. 43]. In this line of research, webgenre learning is performed similarly to *text* categorization [47]. That is, instead of textual units, webpages are processed as target units by means of a bag-of-features model [42], which may but does not necessarily include *hyper*textual features such as the number of hyperlinks in a page. Following this line of thinking, several approaches have been developed that model content-related structures of webpages. This includes the exploration of so-called broad topics [11], or thematic clusters of webpages [37] as well as thematic aspects of page linkage [35]. These approaches – whether topic- or genre-oriented – have in common that they use the bag-of-words model and related methods.

This chapter presents an integrated model of genre- and topic-based structures of web documents. The background of this research is reviewed in Section 15.1.1. In this introduction, we also set the stage for integrating genre with topic modeling from the point of view of structure formation. This is done in terms of thematic-generic hypertext zoning, tracking and sounding as illustrated in Section 15.1.2.

15.1.1 Webgenre Learning in a Two-Level Perspective

Webgenre modeling gains from an extensive research on text categorization with its focus on single texts as the target units of learning [47, 19]. However, by utilizing this research, webgenre modeling abstracts information from a central characteristic of web documents, that is, their *hypertextual structure* [14]. On the one hand, webpages are preferably tagged as correspondents of textual units based on predefined sets of genre labels. This approach disregards both the build-up of websites by interlinked pages (page-external structure) and their segmentation (page-internal structure) [9]. That is, webpages are mainly referred to as instances of webgenres, which are explored for lexical and markup-related features. Obviously, this mono-level approach combines a structural indifference on the side of the input of webgenre learning (webpages) with a structural indifference on the side of its input (genre labels).

The mono-level approach is problematic for several reasons. [32] show that due to effects of structural uncertainty, webpages are non-reliable manifestation units of webgenres. Obviously, pages are layout units that do not necessarily coincide with the boundaries of webgenre instances, which range from units as complex as websites to units as small as segments of single pages. Thus, the genre-related segmentation of pages together with the linkage of the resulting segments is an indispensable ingredient of a *multi-level* perspective on webgenre learning. The multi-level approach is shown in Figure 15.1 in relation to its mono-level counterpart.

As the set of webgenre instances does not coincide with the set of webpages, two scenarios of generic delimitation have to be distinguished. Firstly, in order to demarcate instances of genres that are distributed over several pages we need to know the genres' internal structure since, in this case, pages no longer manifest units on the genre-, but on the subgenre level. Secondly, these subgenre units may also

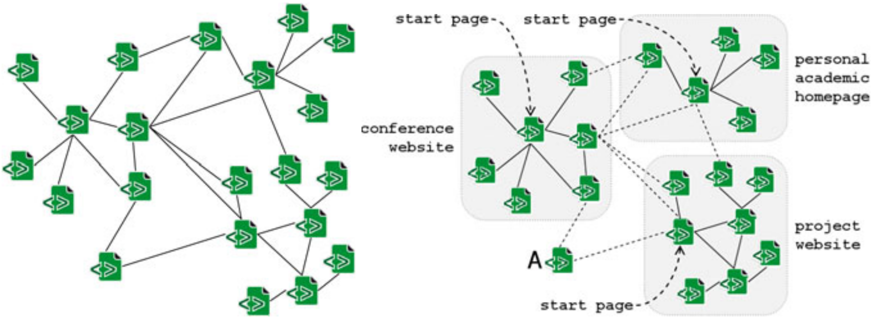


Fig. 15.1 The same outline of the web seen from the point of view of a mono-level approach (left) and its two-level counterpart (right). Vertices denote webpages while edges stand for hyperlinks whose direction is omitted. From the point of view of a two-level approach, Page *A* raises the question whether it belongs to the project website, to the conference website or to any other site. Hypertext zoning aims at answering questions of this sort.

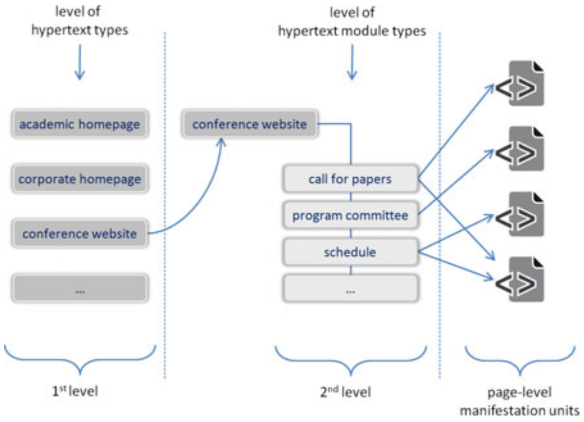


Fig. 15.2 Schematic representation of a two-level model of hypertext types or webgenres: an instance of the webgenre *conference website* is subdivided into several modules or stages (e.g., *call for papers*, *programm committee* and *schedule*) each of which may be manifested by one or more pages.

occur as segments of single pages so that a generic segmentation of pages comes to the fore. In both cases, webgenre learning goes along the demarcation of subgenre units or, analogously, the segmentation of subgenre units within single pages. The present chapter is about this interplay of generic categorization and segmentation, that is, about a *two-level* approach. We present a model of hypertext types, and explore structural modules in order to classify instances of these types. As a result, a segmentation of web documents is conducted to get generic modules as input to classifying the documents by their genre. Accordingly, we distinguish two levels of webgenre modeling (as shown in Figure 15.2):

1. On level 1, web documents are distinguished as instances of webgenres (e.g., as conference websites, personal homepages or city websites).
2. On level 2, constituents of these documents are demarcated above, below, or on the level of pages by mapping them onto genre-related functions that are obligatorily or optionally met to manifest the corresponding webgenre [cf. 17].

To implement this approach, we make the classification of documents on level 1 a function of the segmentation and classification of their constituents on level 2. That is, we build a classifier that explores sets of structural units as a representation model of webgenre instances. In this sense, we present a *bag of structures* approach as an alternative to the *bag of features* approach. As we classify documents subject to their internal structures this means that we integrate the task of hypertext categorization into the framework of document grammars.

At this stage, one may object that there is already a range of approaches that explore bags of *structural* features as, for example, the number of outgoing links in a page (cf. Lim et al 23, Lindemann and Littig 24, Kanaris and Stamatatos 21). It would seem, from this perspective, that what we are doing is tantamount to a structure-based classification. However, we respond from a sign-theoretical perspective that distinguishes between the (explicit) layout structure, the (implicit) logical structure, the (hidden) thematic and the (hidden) functional structure of a document [8, 38]. To clarify this distinction, take the example of URLs as candidate resources of structural features [45]. Just as we resist to calling the page structure of a book a reliable indicator of the logical document structure of the underlying text, we do not assume that URLs provide reliable indicators of hypertext structure. Rather, one has to assume – as is clarified in the *Alternative Document Model* of [49] (cf. Lindemann and Littig 24) – an additional reference point of web document analysis, that is, *physical storage* that includes file format and directory structures. It is important to keep these reference points apart as they refer to different resources of hypertext categorization. This can be exemplified in the framework of Amitay et al's (2003) approach who introduce the notion of a *side link* that exists between pages located in the same directory. Apparently, side links are structural units, but only conditionally. A side link may manifest as, e.g., a *hypotactic* down link in terms of the logical document structure, while at the same time manifesting as a *paratactic* link according to its physical storage [29]. Any approach that explores structural features should clarify their origin according to thematic, generic, layout or physical web document structure before speaking unspecifically of a structure-based approach.

From this point of view, mono-level approaches are vulnerable to criticism for their insufficient sign theoretical underpinning as they blur the source of structuring of webgenre instances. In contrast to this, a sign theoretical notion conceives such instances as complex signs that have a characteristic structure due to their membership in a certain genre. As a sign, a web document has a content plane and an expression plane whose structuring may origin in thematically or functionally based structures. From the point of view of webgenre modeling, we are interested in those structures that have a functional provenance. To map these structures, we transfer the notion of logical document structure of *textual* units [38] onto the level of

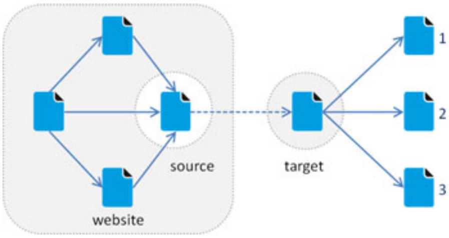


Fig. 15.3 A typical scenario of hypertext zoning that answers the question whether the source page is continued by the target page generically or thematically.

hypertexts. These structures are *logical* as they are based neither upon thematic units, nor on layout units as, for example, webpages. As a consequence, a constituent of the logical *web* document structure may simultaneously cut across the borders of thematic units and layout units. We show that with the *logical web document structure* we gain a further reference point of webgenre learning.

15.1.2 Thematic-Generic Tracking, Zoning and Sounding

Our approach would fall short of the complexity of web documents if we ignored their content. Obviously, document structures may also reflect thematic structures. The aim of including content modeling is to get a further reference point of hypertext zoning, that is, the task of automatic hypertext delimitation. This is exemplified by Figure 15.1 (right), where page *A* raises the question about its generic-thematic identity:

- *Does page A manifest a constituent of a website as an instance of a webgenre?*
- *Is page A thematically related to its neighboring pages (with whom together it forms a thematic cluster) or does it, by being unrelated, change their topics?*

In order to answer questions of this sort in face of the thematic openness of the web, we are in need of an open topic model [52] that grows with the underlying universe of topics. Together with our two-level webgenre model, such a topic model allows for the zoning of hypertexts according to their generic *and* thematic structure. It starts from the idea that hyperlinks can be characterized by the thematic or generic relatedness of the pages linked by them. From the point of view of a source page of a hyperlink in relation to a target page, this scenario can be exemplified as shown in Figure 15.3. It poses three related questions:

1. *Generic continuation:* Is the genre of the target page of a hyperlink functionally related to the genre of the source page? Does the former continue the genre of the latter, is it similar to it or does it change it?
2. *Thematic continuation:* Is the topic of the target page semantically related to the topic of the source page? Does the former continue the topic of the latter, is it similar to it or does it change it?

3. *Thematic-generic continuation:* Is the target both generically and thematically related to the source?

By answering questions of this sort, we get information about the membership of the target to the thematic cluster to which the source belongs (*thematic delimitation*) or to the webgenre that possibly includes both the source and the target (*generic delimitation*). Answering these and related questions is the task of *hypertext zoning* including the delimitation of instances of webgenres, thematic clusters and broad topics in the web [11]. Hypertext zoning can also be conceived as the task of automatically delimiting *websites* as instances of webgenres based on an integrated functional-thematic model irrespective of the pages' physical storage. Generally speaking, hypertext zoning explores the sign character of web documents in order to delimit websites in terms of their meaning and function.

Now, one might object that zoning is more than identifying interlinked pages or clustering them thematically. There is structure formation beyond pairwise linkage in terms of thematic or generic progressions manifested by paths of interlinked pages. This is also the starting point of our approach, which integrates the notion of (thematic/generic) zoning with that of (thematic/generic) tracking and sounding.

1. *Thematic/generic tracking* means that for a stream of documents it is decided to which given topic/genre category a newly encountered document belongs. This is a variant of topic tracking [2], where for a stream of news stories one has to decide whether they are about a given topic or not.
2. *Thematic/generic sounding* is the task of exploring the thematic/generic continuation or extrapolation starting from a given document. By the thematic sounding of a document x we get information about how long we keep track with its topic when following a chain of hyperlinks starting with x. Sounding may also inform us about topic changes or explorations of x. Other than tracking, sounding *looks ahead* by selecting the topic/genre of x and asking how it is developed by documents (mediately) connected to x. From a user perspective, tracking takes place *after* the user has activated a hyperlink starting from x, while sounding is performed *before* deciding where to go next. In this sense, both tasks are related so that methods should be sharable for tackling them. However, other than in topic tracking, sounding does not assume a linearly ordered stream of documents, but a network of interlinked documents. Note that sounding may also inform us about what [9] calls genre/topic drift, that is, the specific discontinuation of genre/topic by interlinked pages.
3. *Thematic/generic zoning* is the most demanding task. It aims at delimiting documents in terms of their thematic/functional structure. Other than sounding, zoning explores the structure of the website to which the focal page x belongs. As exemplified in Figure 15.4, zoning does not only ask what to expect when selecting among the links of x. Rather it asks for the structure and, thus, the delimitation of the whole website to which x belongs. Thus, zoning explores both the direction of pages connected to x and of the pages to which x is connects.

Take the example of an arbitrary article in Wikipedia, say, about *nanotechnology*. In this case, thematic tracking means to answer the question whether this article is

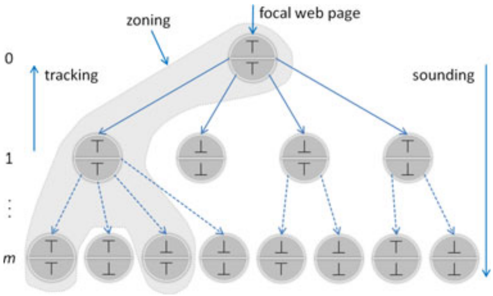


Fig. 15.4 Thematic-generic tracking, zoning and sounding from the point of view of the vertex tagged as *focus*. Vertices are divided into two parts where the upper part denotes thematic continuity by $\top$, while $\perp$ is used to denote thematic discontinuity. Analogously, the lower part denotes generic continuity and discontinuity. Note that zoning generally explores arcs in both directions, that is, outgoing as well as incoming links when using a randomly chosen unit as a starting point to delimit the web document to which it belongs. Further, zoning may also include concurrent paths of units that form "siblings" as indicated by the shaded area.

about the topic that we decided to track. Thematic sounding asks how far we get when following links that focus on nanotechnology, while thematic zoning means to delimit the cluster of all Wikipedia articles about nanotechnology. From the point of view of an application, we may envision a *thematic sounder* as an add-on to a browser that allows you to fix the topic of the present article, masks all hyperlinks that link to thematically unrelated pages and displays for all remaining links the sounded depth of possible continuations. Obviously, such an add-on may help to master the problem of getting lost by the plethora of hyperlinks in Wikipedia. Below, we present such an approach by example of thematic-generic soundings of webpages.

Facing the dual task of hypertext zoning and sounding, the chapter is organized as follows: Section 15.2 presents our two-stage model of genre-oriented hypertext zoning. The model is evaluated by means of a corpus of three webgenres. Section 15.3 presents a model of thematic-generic sounding by focusing on directly linked webpages. Finally, Section 15.4 estimates the effort of thematic sounding by additionally regarding indirectly connected webpages. This is done using Wikipedia as an example.

15.2 A Two-Level Model of Logical Web Document Structures

In this section, we present a two-level model of genre-related hypertext zoning. Other than mono-level approaches that focus on pages as manifestation units of web genres, we integrate a genre-related segmentation in order to explore page segments as manifestation units. Take the example of a *personal academic homepage* that usually manifests a number of modules on the subgenre-level such as *publication*, *teaching*, or *contact* [40]. Obviously, the contact module is also present in

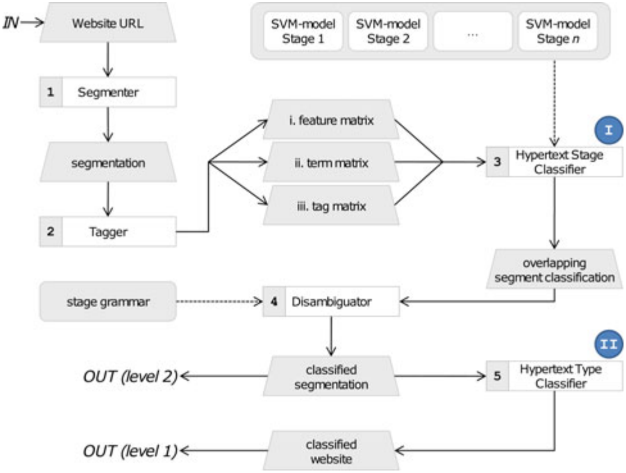


Fig. 15.5 Overview of the two-level model of logical web document structures.

many conference and project websites. So if we focus on single pages as the target units of classification, these web genres would hardly be distinguished correctly. In this example, a unit on the subgenre level (the contact module) is manifested by a *monomorphic* page (that does not manifest any other module of the same or any other genre). Obviously, there are two additional scenarios of manifesting the contact module: on the one hand, it may be distributed over several pages. On the other hand, it may occur as a segment of a single page that additionally manifests different modules of the same or any other genre. In the latter case we speak of a *polymorphic* webpage.

The chapter is about a two-level model of genre that integrates a segmentation of polymorphic webpages in terms of subgenre modules or *stages*.[1] We present and evaluate a two-level algorithm of *hypertext zoning* that integrates the genre-related classification of websites with their segmentation. The idea is to segment pages in order to classify subgeneric modules (*hypertext stage classifier*) that, in turn, are input to classifying sites or pages by their genre (*hypertext type classifier*).

15.2.1 Hypertext Stage Classifier

The architecture of our two-level classifier is shown in Figure 15.5. It includes four steps of computing classified segmentations of webpages as input to our hypertext-type- or webgenre-classifier: in a first step, a layout-based segmentation of webpages is conducted by means of the *Segmenter* module, which decomposes polymorphic pages into monomorphic segments. Secondly, the *Tagger* module generates three representations per segment: a *tfidf*-biased *term vector* of lexical

[1] This term is reminiscent of the notion of *staging* of Halliday & Hasan's (1989) genre model.

features, a *structure vector* of structural features in terms of quantitative structure analysis [31] and a so called *tag vector* that comprises a quantitative model of the usage of HTML-Tags within the segments. This feature model is input to the hypertext-stage-classifier, which uses Support Vector Machines (SVM) for processing the feature vectors. As any such classification may result in ambiguities we, fourthly, integrate a *Disambiguation* module. Based on a Markov-model of stage transitions, that is, based on a probabilistic grammar of subgeneric hypertext stages, this module predicts the most probable stages for ambiguous segments that are finally input to the hypertext type classifier. Subsequently, we describe these four working steps of our stage-classifier in detail.

Hypertext Segmentation

Normally, humans have no problem identifying functionally demarcated segments of webpages such as navigation bars or contact information by exploring their visual depiction. Our segmenter simulates this behavior in that it decomposes pages by exploring their visual depiction in terms of structure- and layout-oriented features (such as whitespace, font-size or differences in color) [39, 53]. The idea is to decompose pages into their *Logical Document Structure* (LDS) by analogy to texts [38]. Thus, we assume that the logical document structure of a page correlates with its visually separable segments. This separability is indicated, for example, by highlighting (e.g., headers with an increased font-size or changing colors), image separators (e.g., logo or animations), or empty sections with no textual or graphical content. At first glance, this approach seems to be superfluous as we may simply explore structure-related HTML tags such as paragraph markers. However, we face the *tag abuse problem* [5], which results from the notorious tendency of writers to overload the functionality of any HTML tag to explicate any kind of document structure. Take the example of highlighting headlines. On the one hand, we may use structure-related tags for annotating them (e.g., `<h1>headline</h1>`). However, we may also utilize document-external stylesheet (e.g., `.myHeader { font-size: 20px;}`) or internal stylesheet information (`<span style='font-size:15px;'>`). As these cases can be mixed, we face a huge range of alternatives that can nevertheless be disambiguated by their visual depiction.

Thus, in order to circumvent the tag abuse problem, we extract the LDS of pages by additionally exploring their cascaded style sheets (CSS), whether internally or externally embedded. To do this, we have selected a set of frequent indicators of segment boundaries (e.g. `<div>`, `<h1>`, `<h2>`, `<a>`, font-size that exceeds a certain threshold and font-color information). This set is denoted by *SF* (see Algorithm 1). For each detected candidate of a segment boundary, a document decomposition is performed that results in a set of consecutive segments denoted by *SV* (see *Segment Cutting* in Algorithm 1). In a second step, we perform a *Segment Re-Connection* that amalgamates consecutive segments, which, because of their size, are unlikely real segments. This step is needed since the initial segmentation often extracts segments that are too small such as navigational items, single headings

```
Require: String $H$ as the input website
  String $C$ as the stylesheet information
  $SF$ as the set of predefined segment features
  $SV$ as the set of output segments
  $min_l$ as the minimum threshold (string length)

  Parse website $H$ and stylesheet information $C$;
  // Segment Cutting
  p:=0; m:=0;
  for each occurrence of $f \in SF$ in H at p do
      add substring $H[m,p]$ to $SV$;
      m:=p;
  end for
  // Segment Re-Connection
  for each entry in $SV$ as $i$ do
      if $i_{length} < \min_l$ then
          connect $SV[i]$ with $SV[i+1]$;
      end if
  end for
  return $SV$
```

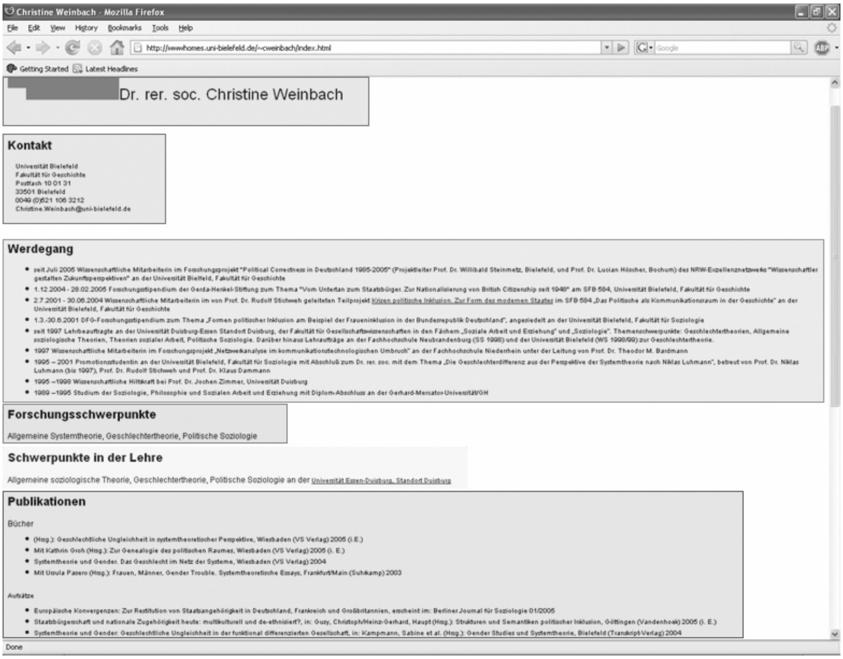
Algorithm 1. Segmenting webpages by means of their visual depiction.

or images. Thus, we use a threshold $\min_l$ to indicate the minimal allowable size of a segment. This threshold corresponds to the number of characters within the candidate segment, where HTML tags and scripts are excluded. As an output of page segmentation, we get a set of possibly monomorphic stages on the subgenre level, which are further used for feature extraction and classification. Figure 15.6 exemplifies a page segmentation as generated by Algorithm 1.

Hypertext Stage Representation

In a typical scenario of text classification, feature selection focuses on lexical units [47]. In our approach, we additionally integrate two levels of feature extraction. This includes structural features and HTML-related features.

In a first step, we build a term vector per input segment based on lexical features by processing the textual content of the segments. All textual data are passed through a linguistic preprocessing module [33, 51] that includes a tokenizer, lemmatizer and stemmer as well as modules for language identification, sentence boundary detection, *Named Entity Recognition* (NER) and *Part-of-Speech* (PoS) tagging. Note that the detection of named entities is of great importance in webgenre analysis as certain stages of academic webgenres (e.g., the publication stage) are easily detected by frequent occurrences of proper nouns. The resulting term vectors are based on nouns, verbs, adjectives, adverbs, numerals, punctation marks and named entities. In order to arrive at a second segment-related feature vector, we explore the frequencies of 91 HTML tags (e.g. `<div>`, `<p>`, `<img>`, `<span>`, ...). This


Christine Weinbach - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://wwwhomes.uni-bielefeld.de/~cweinbach/index.html
Google
Getting Started Latest Headlines
Done

Dr. rer. soc. Christine Weinbach

Kontakt

Universität Bielefeld
Fakultät für Geschichte
Postfach 10 01 31
33501 Bielefeld
0049 (0)521 106 3212
Christine.Weinbach@uni-bielefeld.de

Werdegang

- seit Juli 2006 Wissenschaftliche Mitarbeiterin im Forschungsprojekt "Political Correctness in Deutschland 1985-2006" (Projektleiter Prof. Dr. Willibald Steinmetz, Bielefeld, und Prof. Dr. Lucian Hölscher, Bochum) des NRW-Exzellenznetzwerks "Wissenschaftler gestalten Zukunftsperspektiven" an der Universität Bielfeld, Fakultät für Geschichte
- 1.12.2004 - 28.02.2006 Forschungsstipendium der Gerda-Henkel-Stiftung zum Thema "Vom Untertan zum Staatsbürger. Zur Nationalisierung von British Citizenship seit 1948" am SFB 584, Universität Bielefeld, Fakultät für Geschichte
- 2.7.2001 - 30.06.2004 Wissenschaftliche Mitarbeiterin im von Prof. Dr. Rudolf Stichweh geleiteten Teilprojekt Krisen politische Inklusion. Zur Form des modernen Staates im SFB 584 „Das Politische als Kommunikationsraum in der Geschichte" an der Universität Bielefeld, Fakultät für Geschichte
- 1.3.-30.6.2001 DFG-Forschungsstipendium zum Thema „Formen politischer Inklusion am Beispiel der Fraueninklusion in der Bundesrepublik Deutschland", angesiedelt an der Universität Bielefeld, Fakultät für Soziologie
- seit 1997 Lehrbeauftragte an der Universität Duisburg-Essen Standort Duisburg, der Fakultät für Gesellschaftswissenschaften in den Fächern „Soziale Arbeit und Erziehung" und „Soziologie". Themenschwerpunkte: Geschlechtertheorien, Allgemeine soziologische Theorien, Theorien sozialer Arbeit, Politische Soziologie. Darüber hinaus Lehraufträge an der Fachhochschule Neubrandenburg (SS 1998) und der Universität Bielefeld (WS 1998/99) zur Geschlechtertheorie.
- 1997 Wissenschaftliche Mitarbeiterin im Forschungsprojekt „Netzwerkanalyse im kommunikationstechnologischen Umbruch" an der Fachhochschule Niederrhein unter der Leitung von Prof. Dr. Theodor M. Bardmann
- 1995 – 2001 Promotionsstudentin an der Universität Bielefeld, Fakultät für Soziologie mit Abschluß zum Dr. rer. soc. mit dem Thema „Die Geschlechterdifferenz aus der Perspektive der Systemtheorie nach Niklas Luhmann", betreut von Prof. Dr. Niklas Luhmann (bis 1997), Prof. Dr. Rudolf Stichweh und Prof. Dr. Klaus Dammann
- 1995 –1998 Wissenschaftliche Hilfskraft bei Prof. Dr. Jochen Zimmer, Universität Duisburg
- 1989 –1995 Studium der Soziologie, Philosophie und Sozialen Arbeit und Erziehung mit Diplom-Abschluss an der Gerhard-Mercator-Universität/GH

Forschungsschwerpunkte

Allgemeine Systemtheorie, Geschlechtertheorie, Politische Soziologie

Schwerpunkte in der Lehre

Allgemeine soziologische Theorie, Geschlechtertheorie, Politische Soziologie an der Universität Essen-Duisburg, Standort Duisburg

Publikationen

Bücher

- (Hrsg.): Geschlechtliche Ungleichheit in systemtheoretischer Perspektive, Wiesbaden (VS Verlag) 2006 (i.E.)
- Mit Kathrin Groh (Hrsg.): Zur Genealogie des politischen Raumes, Wiesbaden (VS Verlag) 2006 (i. E.)
- Systemtheorie und Gender. Das Geschlecht im Netz der Systeme, Wiesbaden (VS Verlag) 2004
- Mit Ursula Pasero (Hrsg.): Frauen, Männer, Gender Trouble. Systemtheoretische Essays, Frankfurt/Main (Suhrkamp) 2003

Aufsätze

- Europäische Konvergenzen: Zur Restitution von Staatsangehörigkeit in Deutschland, Frankreich und Großbritannien, erscheint im: Berliner Journal für Soziologie 01/2006
- Staatsbürgerschaft und nationale Zugehörigkeit heute: multikulturell und de-ethnisiert?, in: Gusy, Christoph/Heinz-Gerhard, Haupt (Hrsg.): Strukturen und Semantiken politischer Inklusion, Göttingen (Vandenhoek) 2005 (i. E.)
- Systemtheorie und Gender: Geschlechtliche Ungleichheit in der funktional differenzierten Gesellschaft, in: Kampmann, Sabine et al. (Hrsg.): Gender Studies und Systemtheorie, Bielefeld (Transkript-Verlag) 2004

Fig. 15.6 A sample segmentation of a webpage as part of a personal academic website.

follows the idea that the usage of certain tags correlates with the instantiation of certain generic stages. For example, list-items (e.g. `<li>`, `<ul>`) tend to occur more often in *FAQ*- or *publication*-sections than in *about*- or *contact*-sections. As a third feature vector, we explore quantitative characteristics of the logical document structure of the textual content of the input segments. We compute μ and σ of the length of sections, paragraphs and sentences. The idea is that, e.g., the average sentence and segment length of an *about*- or *publication*-section differs significantly from a *contact*- or *link*-section. The three-part stage representations (see Table 15.2) are finally input to classifying stages as described subsequently.

Hypertext Stage Classification

The stage classifier is based on support vector machines [50], which are efficient in classifying texts [19] as well as web documents [20, 44]. The classification of stages operates in two phases. In the first phase, we train separate SVMs for each subgeneric stage of each targeted webgenre on the basis of manually annotated, pre-classified training data. In the case of personal academic homepages, we distinguish, for example, the stages *publications*, *contact*, *events*, and *objectives*. This is described in Section 15.2.3, which includes an evaluation of the stage classifier. The second phase consists of disambiguating multiply categorized stages (see

Section 15.2.2). Note that the stage classifier combines the three-part stage representation vectors into single vectors as input to the SVMs. Note also that the stage classifier does not provide information about the webgenre of the input page, but classifies its segments in terms of generic modules.

15.2.2 Hypertext Stage Grammar and Type Classifier

The stage classifier labels each segment of an input page by one or more stage labels. This classifier has access to any stage of the underlying set of webgenres to be learnt. Thus, it may assign the labels of stages of different webgenres to the same segment. It may also occur that ambiguous labels are assigned as, for example, *publication* that names the corresponding stage in project sites and in personal academic homepages. Thus, we need to resolve such ambiguities. This is done in two steps: in the training step, the transition probabilities of the stages are predicted separately per targeted webgenre. In this sense, we calculate, for example, the probability by which a publication segment is followed by a contact segment in the context of personal academic homepages. Thus, stage sequences are explored subject to the underlying webgenre so that we arrive at a *probabilistic grammar of stages*. The disambiguation is performed by computing the accumulated transition probability of this first-order Markov model [6], where the stage labels are obtained from the stage classifier. Finally, the most probable sequence of stage labels is used to categorize the segments of the input page.

The hypertext type classifier, that uses the output of the Disambiguation module, is implemented in two variants: on the one hand, we implement a classical bag-of-features model based on the tripartite feature vectors of the Preprocessor (see Figure 15.5) as input to three webgenre-related SVMs. Henceforth, we call this approach the *bag-of-features-classifier*. As an alternative, we build a classifier that explores the set of stage labels of input pages. In this sense, we speak of a *bag-of-structures-classifier* as it analyzes multisets of stage labels instead of the tripartite feature vectors to represent input pages. The bags-of-structures are once more input to SVMs that explore the frequencies of the stages as an additional feature. That is, while the *bag-of-features*-classifier is directly based on the HTML, structure-related and lexical features (see Table 15.2), the *bag-of-structures*-model is a two-level classifier, which uses the latter features to classify subgeneric stages that in the next step define the focal feature units of the webgenre-related classification. In this sense, we speak of a two-level approach.

15.2.3 Experiments

As explained above, we expect that differences between webgenres will correlate with differences in their staging on the subgenre level. That is, we expect that bags of sub-generic modules of a page will reveal its genre. In this sense, we classify webgenres by segmenting and classifying their modules first.

Previous studies [36, 44, 21] of webgenre learning focus on classifying webpages as a whole by disregarding the latter type of structuring. This is reflected by a focus on functionally as well as thematically well-separated genres such as *web shops* in contrast to *FAQs* or *search pages*. Obviously, such genres are separable on the level of lexical features. In contrast to this, we focus on a palette of rather closely related webgenres whose instances may intersect in terms of their content: *conference websites*, *personal academic homepages*, and *academic project websites*. Because of overlaps we expect that these webgenres are more difficult to learn than related palettes reflected in the literature.

The evaluation of our two-level webgenre model is threefold: firstly, we evaluate the performance of the page segmenter in order to show how well it endows the subsequent classifiers with segmentations. Secondly, we analyze the stage classifier in order to measure how well we perform on the subgenre level. Finally, we comparatively evaluate the webgenre classifier based on the *bag-of-features*- and the *bag-of-structures*-model.

Corpus

Although there are few reference corpora [36, 44] for webgenre learning [41], a stage segmentation as needed in the present context has not been done so far. Therefore, we compiled a new reference corpus of instances of conference websites, personal academic homepages, and academic project websites as typical webgenres in the academic domain. The preparation of this training corpus, which is henceforth called the WebgenreStageCorpus (WSC)[2], has been done by selecting and downloading 50 German websites for each of these webgenre. Each of the pages within the WSC has been segmented manually in terms of its genre-related staging (see Table 15.1). The aim of this annotation is to segment monomorphic stages within the pages regarding the stage labels of Table 15.1. The WSC includes 3 webgenres each of 50 websites for which 150 webpages were annotated so that we finally got 1,339 annotations of subgeneric stages.

Regarding the representation model of page segments, we evaluated our feature selection by means of the *GSS coefficient* [47]. Interestingly, with the *GSS*-based feature selection, the categorization neither improved nor deteriorated. In order to implement the SVM-based stage classifier, we used SVM^{light} [19]. For each stage, a separate SVM was trained in a *one-against-all* setting. Based on a leave-one-out cross-validation, we report the F_1-measure, that is, the harmonic mean of precision and recall. Regarding the evaluation of the page segmenter, we removed all manually annotated identifiers and used the entire page as an input to segmentation. The evaluation is based on a correlation analysis that compares the manual annotation with its computed counterpart. Each segment identifier had to be set at exactly the same (character) position in a page in order to be counted as a *true positive*. As a baseline scenario, we computed a segmentation based on tags of headers

[2] The WebgenreStageCorpus can be accessed at `http://www.webgenrewiki.org/`.

Table 15.1 Overview of the WebgenreStageCorpus by webgenre (type) and subgeneric stage (segment).

Type	**Conference**	**Personal**	**Project**
Segment	about accommodation call committees contact disclaimer organizer program registration sightseeing sponsors	contact personal publications research teaching	contact events framework links news objectives project publications staff
#Pages	50	50	50

Table 15.2 Overview of the number of structure-related (Str), tag-related (Htm), and token-related (Tok) features by webgenre. The last column shows the number of stage-related features (Seg) that were used by the *bag-of-features*- and the *bag-of-structures*-model.

Webgenre	Str	Htm	Tok	Seg
Project	29	91	11,734	435
Conference	29	91	56,994	292
Personal	29	91	10,260	612

(`<h1>,<h2>,<h3>,<h4>,<h5>`), paragraphs (`<p>`), divisions (`<div>`) and tables (`<table>`).

Table 15.3 Results of evaluating the Segmenter module (see Figure 15.5).

Model	**Recall**	**Precision**	**F-score**
Segmenter	.936	.625	.745
Baseline	.263	.446	.331

Table 15.4 Results of evaluating the stage classifier for *personal academic website*.

Classes	**Recall**	**Precision**	**F-score**
contact	.947	.857	.899
links	.583	.636	.608
personal	.661	.709	.684
publications	.795	.720	.756
research	.485	.800	.604
teaching	.581	.643	.610
Average	.675	.728	.694
Baseline			.280

15.2.3.1 Results

Table 15.3 shows the results of evaluating the page segmenter. It can be observed that with an F_1-score of .745 we clearly outperform the baseline scenario of .331. This is a promising result, since we used a very strict evaluation setup. Note that the computed identifiers of segment boundaries had to be placed at the same position as their counterparts tagged by the human annotators. In any event, the segmentation of pages into possibly monomorphic units is in about three-quarters of all cases successful.

Table 15.5 Results of evaluating the stage classifier for *conference websites*.

Classes	Recall	Precision	F-score
about	.578	.703	.634
accommodation	.680	.700	.690
call	.350	.389	.368
committees	.609	.609	.609
contact	.581	.720	.643
disclaimer	.706	.667	.686
organizer	.455	.417	.435
program	.692	.838	.758
registration	.729	.771	.749
sightseeing	.708	.739	.723
sponsors	.542	.650	.591
Average	.603	.655	.626
Baseline			.200

Table 15.6 Results of evaluating the stage classifier for *project websites*.

Classes	Recall	Precision	F-score
contact	.823	.869	.849
events	.525	.636	.575
framework	.447	.568	.500
links	.471	.421	.444
news	.539	.560	.549
objectives	.603	.734	.662
project	.799	.789	.794
publications	.761	.761	.761
staff	.500	.807	.617
Average	.608	.683	.639
Baseline			.240

Table 15.7 Results of evaluating the *bag-of-features*-model.

Classes	Recall	Precision	F-score
conference	.640	.640	.640
personal	.618	.627	.622
project	.620	.608	.614
Average	.626	.625	.625
Baseline			.428

Table 15.8 Results of evaluating the *bag-of-structures*-model.

Classes	Recall	Precision	F-score
conference	.894	.919	.906
personal	.917	.941	.929
project	.930	.923	.927
Average	.914	.928	.920
Baseline			.428

Tables 15.4–15.6 show the results of evaluating the stage classifier. Using the tripartite stage representation model (see Table 15.2) as input to the SVM-based classifier, we achieve an F_1-score of .653 on average. As a baseline scenario, we computed a random clustering to get a lower-bound of evaluation. It randomly assigned each segment to one of the stage labels of Table 15.1. Although we clearly outperform this baseline, our results indicate that the classification of subgeneric stages is a challenge. The average F_1-scores range between .626 and .694 for the webgenres. This is certainly not yet an efficient classifier. One reason may be the small number of training examples. In any event, we also observe several stages that are classified with an F_1-score of more than 70%. For example, publication sections within personal academic homepages, program sections of conference websites or contact sections of project websites are segmented efficiently.

Table 15.7 shows the results of evaluating the webgenre classifier based on the *bag-of-features*-model. With an F_1-score of .625 we outperform once more the corresponding baseline, however to a minor degree. Obviously, by inferring the webgenre directly from classical feature vectors, the classifier does not perform very well. Now, look at Table 15.8, which shows the results of the *bag-of-structures*-model. It outperforms the baseline scenario with an average F_1-score of .92 per webgenre. This is a very promising result in terms of our two-level model of webgenre learning. It shows that the threefold feature vectors do not reliably indicate genre membership – at least in the case of the genres considered here. What is more important for categorizing pages correctly is the kind of stages and their frequency as reflected by the *bag-of-structures*-model. In this sense, our findings support a structure-oriented approach. Note that our experiment considers overlapping webgenres and, therefore, goes beyond classical approaches to hypertext categorization with their focus on categories that hardly overlap.

15.3 Thematic-Generic Sounding in the Web

In this section we address the task of thematic-generic sounding by means of open topic models [52]. The basic idea of the algorithm is that hyperlinks can be characterized by the thematic and generic relatedness of the pages linked by them. We focus on a dynamic, content-related categorization and topic labeling of webpages by using a topic-oriented ontology as a reference point. This task is challenging for two reasons:

1. The topic universe is in a state of permanent flux so we cannot presuppose a fixed set of topic labels as demanded by supervised learning. Thus, we need access to a dynamically growing ontology as a knowledge resource for topic labeling [30].
2. Document-centered tasks such as topic labeling or text categorization suffer from the *bottleneck of knowledge acquisition* [7]. This means that documents that are thematically related may nevertheless differ in their vocabulary. Thus, overlapping lexical features do not sufficiently indicate the thematic relatedness of documents.

There are several approaches that try to overcome this problem by means of latent semantic analysis, SVM-based semantic spaces or WordNet and related resources [18, 22, 10]. In contrast to this, we utilize *Explicit Semantic Analysis* (ESA) [15] by mapping any input text onto a given concept hierarchy as an explicit knowledge resource [12, 16, cf.]. More specifically, we utilize the concept hierarchy to represent the content of input texts by categories that are not directly manifested within them, that is, categories as generalizations of concepts, which are directly manifested within the document.

One resource that meets both requirements (hierarchical explicitness and conceptual openness) is given by social ontologies as instantiated by Wikipedia's category system [52]. Based on the collaboration of a multitude of users, who are constantly creating, editing and classifying Wikipedia's article collection, this resource ensures the dynamic aspect in topic labeling in that it provides a category system in terms of a generalized tree [28].

Our algorithm of thematic sounding of web documents is based on two stages: each input document is mapped onto an article-based semantic space in order to assess the similarity of Wikipedia articles and input documents. Then we use the category information within articles that are similar to the input to access Wikipedia's category system. This allows us to retrieve thematically related categories as topic labels for the input documents. Both stages are explained subsequently.

Wikipedia-Based Semantic Spaces

We utilize ESA [15] to explore Wikipedia as a knowledge resource. More specifically, we explore Wikipedia's article network in conjunction with its category system to map input documents onto two semantic spaces: the one being spanned by means of Wikipedia's article network (network A), the other being spanned by means of Wikipedia's category system (network C).

We start from the link-content conjecture [35] according to which a web document shares content with those documents to which it is linked. In terms of Wikipedia, this means that if we link an input document x into network A, we can explore the categories c of the neighboring articles of x within A as entry points to network C (i.e., category system of Wikipedia). The idea is that the categories c would likely have been used to categorize x, if this text were part of Wikipedia's article network. Obviously, this is a way to solve the knowledge bottleneck problem as, now, we have access to categories that are not necessarily manifested within x, but categorize its content. Because of this feature, we speak of network A and C as a two-level semantic space. Note that both concept spaces A and C are minimized in terms of their size. Basically, all articles are deleted with an in-degree of less than 5 links, while all categories are deleted that categorize less than 10 articles.

In order to introduce this model, we need several definitions: we define C_{art} as the set of all titles of Wikipedia articles (network A) and C_{cat} as the set of all category labels within network C. Each Wikipedia article x is represented by a vector $\mathbf{c}_{art}(x)$ of all properly selected lexical features that occur in x and are additionally biased according to the *tfidf* scheme of [42]. Any input document y, whose content has to be computed, is represented the same way by a feature vector $\mathbf{c}_{art}(y)$. Finally, each category $c \in C_{cat}$ is represented by a vector $\mathbf{c}_{cat}(c) \in \{0,1\}^{|C_{art}|}$ whose dimensions define whether the corresponding article is categorized by c or not.

Now, a given input document x is mapped onto the semantic space A by the function

$$f_{art} : \{\mathbf{c}_{art}(x) \,|\, x \in \mathbb{C}\} \rightarrow [C_{art}]^{10} \tag{15.1}$$

which retrieves the top 10 titles of those articles y that by their vector representations $\mathbf{c}_{art}(y)$ are most similar (i.e., least distant) to $\mathbf{c}_{art}(x)$ in terms of the cosine measure.[3] $\mathbb{C}$ is the (open) corpus of input documents. Next, x is mapped onto the semantic space C by the vector $\mathbf{c}_{result}(x) \in \mathbb{R}^{|C_{art}|}$ whose dimensions accord to the cosine similarity of the vector representations of x and the corresponding article y, supposing that the title of y belongs to $f_{art}(\mathbf{c}_{art}(x))$. Otherwise, if the corresponding

[3] Note that for any set X, $[X]^k$ is the set of all subsets of X of exactly 10 elements.

Table 15.9 Initial and generalized topic labels by means of the German Wikipedia taxonomy.

Input Text: "*Das Grösste Kursplus seit 1985 wurde an den acht hiesigen Börsen im vergangenen Jahr erzielt. Beispielsweise zog der Deutsche Aktienindex um 47 Prozent an (vgl. SZ Nr. 302). Trotz Rezession und Hiobsbotschaften von der Unternehmensfront hatten sich zunächst britische und amerikanische Fondsverwalter [...]*"
Output:
Related Article: Anlageklasse / Bundesanleihe / Nebenwert / Bullen- und Bärenmarkt / Börsensegment
Initial Topics: Unternehmen nach Wirtschaftszweig / Unternehmen / Unternehmensart / Deutscher Finanzmarkt / Investmentgesellschaft
Generalized Topics: Finanzierung / Finanzmarkt / Ökonomischer Markt / Wirtschaft / Rechnungswesen

article does not belong to $f_{art}(\mathbf{c}_{art}(x))$, the dimension's value is zero. This allows us to apply the function

$$f_{cat} : \{\mathbf{c}_{result}(x) \,|\, x \in \mathbb{C}\} \rightarrow [C_{cat}]^{10} \tag{15.2}$$

which retrieves the top 10 labels of those categories y that by their vector representations $\mathbf{c}_{cat}(y)$ are most similar (i.e., least distant) to $\mathbf{c}_{result}(x)$ – once more in terms of the cosine measure.[4]

Category-Based Topic Labeling

The next step is to process Wikipedia's category system to retrieve node labels (of a certain scale of generalization) as topic labels for the input document x. We call this procedure *concept generalization*. Since Wikipedia's category system spans a sort of generalized tree rather than a directed tree [28], it has to be preprocessed. That is, starting from the root category *Category:Contents*, we explore the category system by a breadth-first search to transform it into a directed tree. Next, we use the category labels of $f_{result}(\mathbf{c}_{cat}(x))$ as starting points of a hill-climbing search. That is, we move upwards through the tree structure of the taxonomy to enter more general categories. Since the category taxonomy is structured from general categories to specific ones, we derive a topic generalization on the way up. We use a threshold L, that is, the number of edges to be moved upwards, to restrict the scope of generalization. The higher L, the more general the retrieved categories. In the present study, we set $L = 4$.

See the tables 15.9–15.10 for a sample output of this algorithm. So called *initial topics* denote categories that are closely related to the content of the input document. This holds, for example, for *olympic athlete* or *basketball player*. In contrast to this, so called *generalized topics* label the topic of the input document in a more generalized manner. This holds, for example, for the categories *sport*, *Germany*,

[4] For more details on this approach and its evaluation see [51].

Table 15.10 Initial and generalized topic labels by means of the English Wikipedia taxonomy.

Input Text: "*Nowitzki's performance this year has vaulted him past the late Petrovic as the NBA's best-ever European import. But the Bavarian Bomber is fast becoming one of the NBA's best players, period. [...]*"
Output:
Related Article: Dirk Nowitzki / Dallas Mavericks / Avery Johnson / Jerry Stackhouse / Antawn Jamison
Initial Topics: basketball player / basketball / athlete / olympic athlete / basketball league
Generalized Topics: sport / United States / basketball / Germany / sport by country

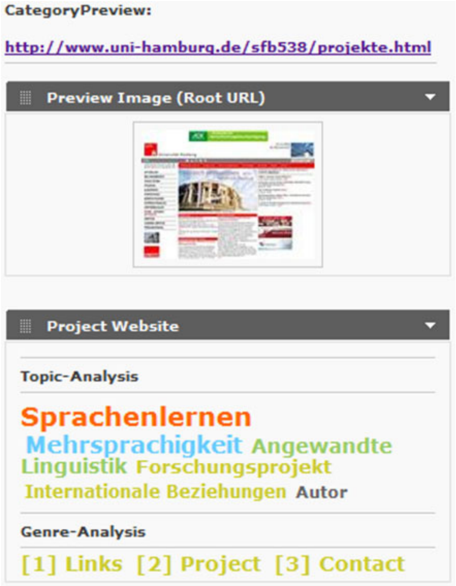
and *sport by country*, which are closer to the root node of Wikipedia's category system (note that [Dirk] Nowitzki is a German basketball player).

Thematic-Generic Sounding by Means of the ThematicGenericLinker

Now we are in a position to utilize our Wikipedia-based semantic space together with our two-level model of webgenres to implement a thematic-generic sounder, the so called `ThematicGenericLinker`, for directly linked web pages. Based on the idea that hyperlinks target at pages that are likely related to their source in terms of genre or topic, we automatically process the target of a page in two ways: firstly, we map its content on the category system of Wikipedia in order to label its topic. This gives us information about whether the source of the link is thematically similar to its target (see Figure 15.3 and Section 15.1.2). Secondly, we segment and classify the target by means of our two-level model of webgenres. This gives us information about the subgeneric staging of the target page and about the genre manifested by it. Currently, the genre-related sounder is implemented only for German and here only for the webgenres analyzed above, while its topic-related counterpart works for German pages as well as for English pages. Figure 15.7 exemplifies the `ThematicGenericLinker` by an explicitly entered URL. In its working mode, the `ThematicGenericLinker` uses a context menu to provide a look ahead on the genre and topic of the page targeted by the corresponding link.[5]

The `ThematicGenericLinker` may be seen as a proof of concept of how to integrate genre with topic modeling. By this approach we pave a way to go beyond the narrow focus of webgenre modeling on the one side and text or web categorization on the other. The reason is that by the zoning and sounding of web documents, we establish new tasks in web mining.

[5] See `http://api.hucompute.org/semantic-linker/` for a download of the `ThematicGenericLinker`.



CategoryPreview:

http://www.uni-hamburg.de/sfb538/projekte.html

Topic-Analysis

Sprachenlernen Mehrsprachigkeit Angewandte Linguistik Forschungsprojekt Internationale Beziehungen Autor

Genre-Analysis

[1] Links [2] Project [3] Contact

Fig. 15.7 Thematic-generic sounding by means of the `ThematicGenericLinker` by example of a project website.

15.4 Bounds of Thematic Sounding in Wikipedia

In the previous section, we presented an algorithm for thematic-generic sounding that focuses on directly linked pages. In this section, we overcome this limitation by studying the thematic sounding of immediately *and* mediately connected pages. More specifically, we ask for an upper bound of the depth and breadth of thematic sounding by starting from any randomly chosen start page. The reason to do this is that without such knowledge, algorithms of thematic sounding are doomed to fail when facing the sheer amount of interlinked pages in the web. Therefore, we need to know the number of pages that have to be processed "on average" by a sounding algorithm. Such a bound tells the application where to stop even in cases of complex document networks. In this section, we present an approach for calculating such a bound.

Generally speaking, the web is a scale-free small world [1, 4, 54]. This also holds for embedded web networks such as Wikipedia and special wikis [55, 26]. Formally speaking, from the point of view of thematic sounding, this characteristic results in too many vertices that belong to the cumulative j-sphere of the start vertex of sounding. By the *j-sphere* [13] $S_j(v)$ of a vertex $v \in V$ of a digraph $D = (V,A)$ we denote the set $S_j(v) = \{w \in V \mid \delta(v,w) = j\}$ of all vertices $w \in V$ whose geodesic distance $\delta(v,w)$ from v to w is j. By the *cumulative j-sphere* of a vertex v we denote the set

$$\hat{S}_j(v) = \cup_{i=0}^{j} S_i(v) \tag{15.3}$$

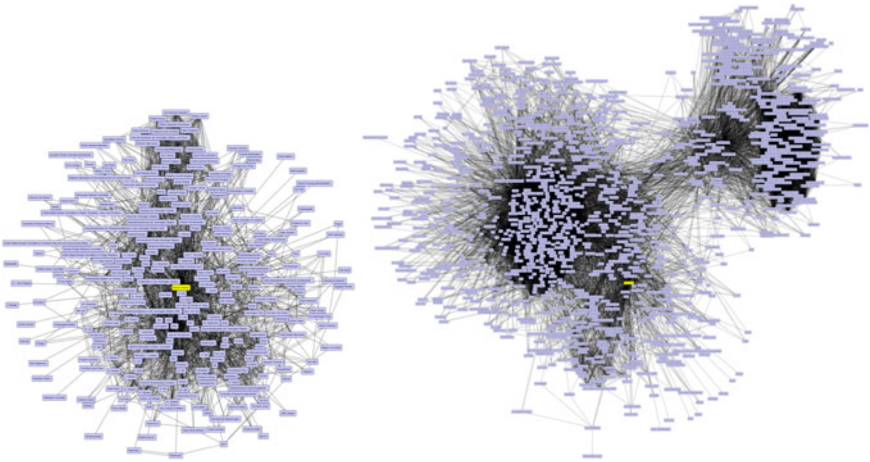


Fig. 15.8 Left: the article subgraph of the German Wikipedia that is induced by the set of 306 articles that are directly linked by the article *Barack Obama*. Right: selection of 1,000 articles of the article subgraph of the German Wikipedia that is induced by the set of 32,244 articles that are either linked by one or two links to the entry article *Barack Obama* (downloaded by February, 2009).

The problem with scale-free small worlds is that even for small values of j (e.g., $j = 3 \pm 1$) $\hat{S}_j$ gets too large such that $|\hat{S}_j| \approx |D|$. In Wikipedia, for example, $|\hat{S}_3| \approx |D|$ if the article graph is represented as an undirected graph – there are just too many neighbors in the 3-sphere of a Wikipedia article (see Figure 15.8 for an example).

This characteristic is a direct consequence of the short average geodesic distance in small-world networks. One reason for this distance is the existence of hubs that link to a multitude of vertices of the same network [4]. Obviously, it is bad advice to disregard such hubs with a high degree of centrality when computing thematic soundings, as the network gets disconnected by deleting them. Thus, we need to follow another approach when performing thematic sounding in scale-free document networks. That is, we need to focus on some appropriately selected subset of vertices that are connected with the chosen start vertex of sounding. In order to provide a model of subset selection that is well-founded in terms of semiotics, we utilize the notion of so-called 1-*dominator sets* of [46].

Generally speaking, for a start vertex $v \in V$ of a digraph (V,A), [46] selects a subset $D_v \subset V$ of vertices $w \in D_v$ that are solely reachable from vertices on some path between v and w. In terms of our application scenario, D_v subsumes all articles that are thematically related to v as the entry point of *all* paths by which these articles can be reached. It will be shown that this model is too restrictive to be used for thematic sounding. However, Saunders' approach can be extended to get a first model of thematic sounding in document graphs. This can be done by restricting the process of sounding to those vertices that are *triggered* by v in a sense to be defined subsequently, while excluding thematically polymorphic vertices, which are reachable from vertices of different thematic provenance.

In order to implement this concept, we complement Saunders' model by the notion of *successor sets*. Thereafter, we introduce the notion of *trigger sets* as a sort of subset in a certain range, using Saunders' 1-dominator sets as a lower bound and successor sets as an upper bound of subset selection. Finally, we compute statistical moments of the distribution of trigger sets in Wikipedia to get insights into the bounds of thematic sounding in complex document networks.

15.4.1 Dominator, Successor and Trigger Sets

[46] defines the 1-*dominator set* $\mathrm{D}(D)$ of a digraph $D = (V,A)$ as a family of subgraphs of D:

$$\mathrm{D}(D)=\{D_v=(V_{\mathrm{D}_v},A_{\mathrm{D}_v})|v\in V\wedge\forall w\in V: D_v\neq D_w=(V_{\mathrm{D}_w},A_{\mathrm{D}_w})\Rightarrow V_{\mathrm{D}_v}\nsubseteq V_{\mathrm{D}_w}\} \quad (15.4)$$

where for each $v \in V$ the subdigraph D_v of D is recursively computed as follows:

$$D_v^{(0)} = \left(\{v\},A_v^{(0)}\right), \ldots, D_v^{(i+1)} = \left(V_v^{(i+1)},A_v^{(i+1)}\right) \quad (15.5)$$

such that for $IN(v) = \{w \in V \mid \exists a \in A : in(a) = w \wedge out(a) = v\}$ we define:

$$V_v^{(i+1)} = \{w \in V \mid IN(w) \cap D_v^{(i)} \neq \emptyset \wedge IN(w) \subseteq D_v^{(0,\ldots,i)} = \cup_{k=0}^{i} D_v^{(k)}\} \quad (15.6)$$

$A_v^{(i+1)} \subseteq A$ is the arc set of the subgraph of D induced by $V_v^{(i+1)}$. Finally, we set

$$D_v \leftarrow D_v^{(|D|)} \quad (15.7)$$

D_v is called *acyclic* in the sense that the subdigraph $D_v - v$ induced by $V_{\mathrm{D}_v} \setminus \{v\}$ is acyclic [46]. Each vertex v for which there exists an acyclic structure $D_v \in \mathrm{D}(D)$ is called the *trigger vertex* of D as it is the root of a maximally acyclic component of D. In this sense, $\mathrm{D}(D)$ is a decomposition of D into acyclic components of the digraph D [46]. In this paper, we call any graph D_v defined according to Formula 15.7 the *dominator graph induced by v* and its vertex set the corresponding *dominator set*. Further, we compute this set for any vertex $v \in V$, whether it is maximal in the latter sense or not. Finally, we compute the set

$$\hat{\mathrm{D}}(D) = \{D_v \mid v \in V\} \supseteq \mathrm{D}(D) \quad (15.8)$$

and call it the *dominator set* of the digraph D.

The vertex set of any dominator graph D_v is restricted to vertices w of the digraph D that are incident to arcs whose tail lies on a path from v to w. In this way, D_v excludes every other vertex even if it is reachable from v. In order to capture the set of all vertices that are reachable from the "trigger vertex" v, we introduce the notion of a successor set. Formally speaking,

$$\mathrm{S}(D)=\{S_v=(V_{\mathrm{S}_v},A_{\mathrm{S}_v})\mid v\in V \wedge \forall w\in V: S_v\neq S_w=(V_{\mathrm{S}_w},A_{\mathrm{S}_w}) \Rightarrow V_{\mathrm{S}_v} \nsubseteq V_{\mathrm{S}_w}\} \quad (15.9)$$

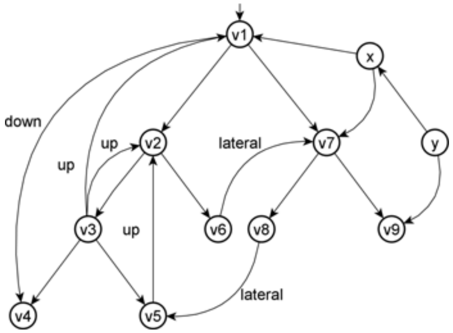


Fig. 15.9 A schematic example of a generalized tree triggered by vertex v_1. By spanning a dominator graph starting from v_1, vertices $v_2,\ldots,v_9$ are excluded though being (mediately) connected to v_1. If we alternatively span a trigger graph, vertices $v_2,\ldots,v_6$ are included. Further, if we allow that vertices are linked from the outside of the successor set of v_1 supposed that these outside vertices are directly connected to v_1, then vertices $v_2,\ldots,v_8$ are also included. Thus, only vertex v_9 is excluded since it is dominated by vertex y, which is connected by a path of length 2 to v_1.

is the set of all *maximum successor graphs* of vertices $v \in V$, where the *successor set* V_{S_v} is the set of all vertices that lie on a path starting from v, and $A_{\mathrm{S}_v} \subseteq A$ is the arc set of the subgraph of D induced by V_{S_v}. Finally, we define

$$\hat{\mathrm{S}}(D) = \{S_v = (V_{\mathrm{S}_v}, A_{\mathrm{S}_v}) \,|\, v \in V\} \tag{15.10}$$

by analogy to $\hat{\mathrm{D}}(D)$ as the *successor set* of the digraph D (whether it is maximal or not). Note that for any vertex $v \in V$, D_v is a subgraph of the successor graph S_v. Note also that $S_v = C_D(v)$ (the component of v in D, that is, the subgraph of D induced by the set of all vertices that are reachable from v).

Based on these preliminaries we can now introduce the notion of (generalized) *trigger sets* of a digraph D that we use to estimate some bounds of thematic sounding in document networks. From the point of view of thematic sounding, successor sets induce oversized subgraphs, while dominator sets generate undersized component graphs. In the former case, vertices may be selected that are thematically dissimilar to the trigger vertex v, while in the latter case vertices may be excluded that actually fit the topic of v. Thus, we need a notion of a trigger set in the range of these extreme cases. This can be exemplified as follows.

1. In Figure 15.9, v_1 is used as a trigger vertex. In this scenario, vertex x is directly connected to v_1 from the outside of the successor graph S_{v_1} induced by v_1. According to the link-content conjecture [35], which states that the content of a page is similar to the content of the pages that link to it, we can assume a thematic relation between x and v_1. However, x may also link to v_1 by serving an organizational function as in, for example, the main page of Wikipedia. Both cases (of a thematic and organizational tie) apply if x is the entry page of a portal

to which v_1 is thematically related. Note that a geodesic distance of 1 (as covered by the link-content conjecture) is significantly below the average geodesic distance in Wikipedia, that is, $3+x$, $0 < x < 1$, for the undirected case [55]. In any of these scenarios, a link from x to v_7 does not interfere with the claim that v_7 is thematically triggered by v_1 independently from any vertex outside of S_{v_1}. Thus, we can disregard x when evaluating v_7 as an element of the trigger graph induced by v_1. This does not hold for vertex y, which is two arcs away from v_1. A distance of 2 is close to the expected value of the geodesic distance in a scale-free graph so we assume that v_9 is thematically more related to y than to v_1. By this consideration, we get a rule to include vertices in the trigger set of a vertex even if they are triggered from outside its successor set.

2. Now, look at vertex v_2 in Figure 15.9. This vertex is excluded from the dominator graph D_{v_1} as it is incident to an arc that starts from v_3. However, v_3 belongs to the successor set S_{v_1} in a way that all tails of arcs that end in v_2 also belong to S_{v_1}. In this case, v_2 can be seen to be thematically triggered by v_1 supposing that all tails of all arcs ending at v_2 are triggered the same way (irrespective of whether they lie on a path from v_1 to v_2 or not). By this consideration, we get a rule to include vertices into the trigger set of a vertex v even if it is excluded from the dominator set D_v. Note that this rule is trivially met in part by Saunders as demonstrated by the arc from v_3 to v_1: by definition, v_1 is an element of D_v.

In order to implement a notion of trigger graphs that meet these two rules, we utilize the notion of a *Directed Generalized Tree* (DGT) [27]. Directed generalized trees are digraphs with a kernel hierarchical structure that is superimposed by graph-inducing upward, downward and lateral arcs as exemplified in Figure 15.9. Obviously, these types of arcs do not interfere with evaluating whether vertices are being triggered or not. That is, a vertex does not lose its status as being triggered if it is, for example, incident with a lateral arc in the DGT spanned by the successor set S_v of the trigger vertex v. Based on these considerations, we can now define the set of all *maximum trigger sets* $\mathrm{T}(D)$ of a digraph $D = (V,A)$ as

$$\mathrm{T}(D)=\{T_v=(V_{\mathrm{T}_v},A_{\mathrm{T}_v})\,|\,v\in V\wedge\forall w\in V: T_v\neq T_w=(V_{\mathrm{T}_v},A_{\mathrm{T}_v})\Rightarrow V_{\mathrm{T}_v}\not\subseteq V_{\mathrm{T}_w}\} \quad (15.11)$$

By analogy to maximum dominator sets, T_v is recursively defined as follows:

$$T_v^{[0]} = (\{v\},A_v^{[0]}),\ \ldots,\ T_v^{[i+1]} = (V_v^{[i+1]},A_v^{[i+1]}) \quad (15.12)$$

where

$$V_v^{[i+1]} = \{w \in V \,|\, IN(w) \cap V_v^{[i]} \neq \emptyset \,\wedge\, IN(w) \subseteq V_{\mathrm{S}_v} \cup IN(v)\} \quad (15.13)$$

and $A_v^{[i+1]} \subseteq A$ is the arc set of the subgraph of D induced by $V_v^{[i+1]}$. Finally, we set

$$T_v \leftarrow T_v^{[|D|]} \quad (15.14)$$

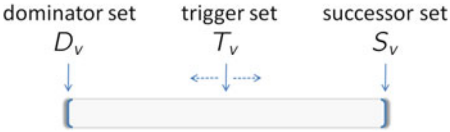


Fig. 15.10 The range of dominator, trigger and successor graphs and their vertex sets, respectively.

By analogy to $\mathrm{D}(D)$ and $\mathrm{S}(D)$, any $T_v \in \mathrm{T}(D)$ is maximal in the sense that there is no graph $T_w \in \mathrm{T}(D)$ such that T_v is a subgraph of T_w. As before, this view is too restrictive. In Wikipedia, for example, any article may be used as an entry point to compute thematic soundings. The reason is that users may directly access them from the outside, for example, via a search engine. In order to capture this scenario, we define the *trigger set* of a digraph as follows:

$$\hat{\mathrm{T}}(D) = \{T_v \,|\, v \in V\} \supseteq \mathrm{T}(D) \tag{15.15}$$

It is easy to see that any $T_v \in \hat{\mathrm{T}}(D)$ spans a directed generalized tree in the sense of [27]. In order to prove this we need to show two things:

- Firstly, we need to show that for all vertices $w \in V_v^{[|D|]}$ there exists *at least* one simple path from v to w in $T_v^{[|D|]}$.
- Secondly, we need to show that there is no arc $a \in A_v^{[|D|]}$ such that $in(a) = u$ and $out(a) = v$ for some $u \notin V_{\mathrm{S}_v}$ as the vertex set of the successor set S_v.

This follows directly from the definition of T_v. Further, for any vertex $v \in V$ it holds that

$$V_{\mathrm{D}_v} \subseteq V_{\mathrm{T}_v} \subseteq V_{\mathrm{S}_v} \text{ for } D_v = (V_{\mathrm{D}_v}, A_{\mathrm{D}_v}), T_v = (V_{\mathrm{T}_v}, A_{\mathrm{T}_v}), S_v = (V_{\mathrm{S}_v}, A_{\mathrm{S}_v}) \tag{15.16}$$

Thus, trigger sets as defined by T_v span subgraphs in the range of dominator and successor sets. This defines a range of selected subsets of vertices as input to thematic sounding as depicted by Figure 15.10: in cases where $D_v \approx T_v \not\approx S_v$, the trigger set of a vertex is restricted to its dominator set. That is, T_v mainly contains vertices that are strictly triggered by v as captured by D_v, while the remainder of its successor nodes are thematically triggered from outside of S_v. Conversely, if $D_v \not\approx T_v \approx S_v$, most vertices that are reachable from v are thematically triggered by v only. In this case, S_v is thematically unambiguously traversable from the point of view of v. This scenario is exemplified by an article in the periphery of Wikipedia that serves as an entry point to articles that further specify the topic of v and only of v.

An algorithmic specification of trigger sets is provided by Algorithm 2. Its time complexity is estimated by the following corollary.

Corollary 1. *For an input digraph $D = (V, A)$, the time complexity of Algorithm 1 is in the order of $O(|V| + |A|)$.*

```
input  : a digraph $D = (\{v_1, \ldots, v_m\}, A)$ together with a vertex $v = v_j \in V$
output: the trigger graph $T_v = (V_{\mathrm{T}_v}, A_{\mathrm{T}_v})$ induced by $v$ over $D$
compute $S_v = (V_{\mathrm{S}_v}, A_{\mathrm{S}_v})$;
$n \leftarrow |S_v|$; $V_Y \leftarrow V_{\mathrm{S}_v} = \{v_{i_1}, \ldots, v_{i_n}\}$; $A_Y \leftarrow A_{\mathrm{S}_v}$; $Y_v \leftarrow (V_Y, A_Y)$;
$\mathbb{X} \leftarrow V_{\mathrm{S}_v} \cup IN(v)$;
for $w = v_{i_1}..v_{i_n}$ do
    if $IN(w) \not\subseteq \mathbb{X}$ then
        $Y_v \leftarrow Y_v - w$;
    end
end
$T_v \leftarrow C_{Y_v}(v)$;
```

Algorithm 2. An algorithm for computing trigger sets T_v. Note that for any digraph X, $C_X(v)$ is the component (subgraph) of v in X.

Proof. $S_v = C_D(v)$ can be computed by a breadth-first search in the order of $O(|V| + |A|)$. In the worst case, $V_{\mathrm{S}_v} = V$. Thus, the for loop is in the same order. $C_{Y_v}(v)$ can be computed by a breadth-first search, too, so that, finally, $O(3(|V| + |A|)) = O(|V| + |A|)$.

15.4.2 Statistical Moments of Trigger and Dominator Sets

$\hat{\mathrm{D}}(D)$, $\hat{\mathrm{T}}(D)$ and $\hat{\mathrm{S}}(D)$ define three reference points for specifying candidate scopes of thematic sounding. Each of these sets defines a distribution of graphs that can be characterized by statistical moments. We describe any such distribution $\{X_v = (V_{\mathrm{X}_v}, A_{\mathrm{X}_v}) \,|\, v \in V\}$ by the mean eccentricity of the vertices v in conjunction with the mean order of X_v. This gives estimates of the mean *depth* and *breadth* of thematic sounding in the underlying document network when starting from any vertex v. We compute the arithmetic mean of these distributions and consider the standard deviation as an dispersion parameter. Since $\forall v \in V : V_{\mathrm{D}_v} \subseteq V_{\mathrm{T}_v} \subseteq V_{\mathrm{S}_v}$, D_v and S_v can be used to compute lower and upper bounds of thematic sounding, while T_v focuses on the corresponding mean effort.

Figure 15.11 shows the box and whisker plots of these indices computed by example of 66 Wikipedias of a size in the range of 10,000-150,000 articles. These 66 networks (in 66 different languages) have been taken from a corpus of 263 releases of Wikipedia that were downloaded in November/December 2008 [28]. In this subcorpus, the mean of the distribution of mean eccentricity values of trigger graphs is approximately 7, while the corresponding mean order is around 114. Analogously, the mean of the distribution of mean eccentricity values of successor sets is 24 and the corresponding mean order is 35,137 – obviously, the latter number is beyond what can be processed in the course of thematic sounding, while a bound of 114 vertices gives a manageable number.

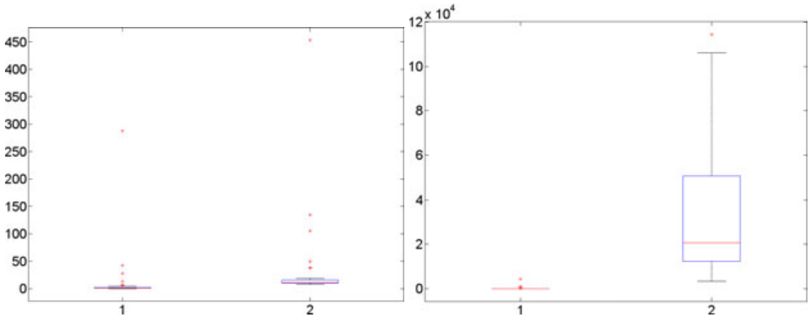


Fig. 15.11 Left: box plots of the distribution of mean eccentricity values in trigger sets (1) and successor sets (2). Right: box plots of the distribution of mean order values in trigger sets (1) and successor sets (2). All distributions are computed based on 66 releases of Wikipedia.

By the results shown in Figure 15.11, we get estimators of the *depth* and *breadth* of thematic sounding in document networks as exemplified by Wikipedia. From the point of view of these results, we get first hints on how to restrict the scope of thematic sounding in Wikipedia in a meaningful way even if the start vertex is randomly chosen. More specifically, starting from Algorithm 2, we can use the mean depth and breadth as stop conditions: whenever Algorithm 2 has spanned a graph whose eccentricity exceeds the bound of 7 vertices or whose order exceeds the number of 114 vertices, it stops. This twofold stop condition guarantees that Algorithm 2 processes in a reasonable time without considering too many articles for thematic sounding. Further, this approach opens the door to extending the findings of Section 15.3 beyond the scope of immediately linked web pages. The reason is that we can now prevent a complete expansion of vertices by restricting the set of documents to be considered in terms of the latter bounds. The present section is a first step to paving the way for making this task a manageable endeavor.

15.5 Conclusion

In this chapter, we tackled two related tasks in web mining: thematic-generic zoning, that is, delimitating websites as instances of webgenres, and thematic-generic sounding, that is, checking the continuity of interlinked pages in terms of topic or genre. In this way, we provided a framework for integrating content and structure modeling in web mining that goes beyond mono-level classification models with a focus on single pages. Further, we computed estimators for the expected depth and breadth of thematic sounding that can be used as stop conditions in corresponding sounding algorithms. Future work will deal with extending this approach to provide thematic-generic sounders as browser add-ons.

Acknowledgement

Financial support of the German Research Foundation (DFG) through the Research Group 437 *Text Technological Information Modeling* via the Project A4 *Induction of Document Grammars for the Representation of Logical Hypertext Document Structures* is gratefully acknowledged.

References

[1] Adamic, L.A.: The small world of web. In: Abiteboul, S., Vercoustre, A.M. (eds.) Research and Advanced Technology for Digital Libraries, pp. 443–452. Springer, Heidelberg (1999)

[2] Allan, J. (ed.): Topic Detection and Tracking. Event-based Information Organization. Kluwer, Boston (2002)

[3] Amitay, E., Carmel, D., Darlow, A., Lempel, R., Soffer, A.: The connectivity sonar: detecting site functionality by structural patterns. In: Proc. of the 14th ACM Conference on Hypertext and Hypermedia, pp. 38–47 (2003)

[4] Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)

[5] Barnard, D.T., Burnard, L., DeRose, S.J., Durand, D.G., Sperberg-McQueen, C.M.: Lessons for the World Wide Web from the text encoding initiative. In: Proc. of the 4th International World Wide Web Conference "The Web Revolution", Boston, Massachusetts (1995)

[6] Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. The Annals of Mathematical Statistics 37(6), 1554–1563 (1966)

[7] Berthold, M., Hand, D.J.: Intelligent data analysis. An Introduction. Springer, Heidelberg (1999)

[8] Biber, D.: Dimensions of Register Variation: A Cross-Linguistic Comparison. Cambridge University Press, Cambridge (1995)

[9] Björneborn, L.: Genre connectivity and genre drift in a web of genres. In: [34] (2010)

[10] Bloehdorn, S., Hotho, A.: Boosting for text classification with semantic features. In: Mobasher, B., Nasraoui, O., Liu, B., Masand, B. (eds.) WebKDD 2004. LNCS (LNAI), vol. 3932, pp. 149–166. Springer, Heidelberg (2006)

[11] Chakrabarti, S., Joshi, M., Punera, K., Pennock, D.M.: The structure of broad topics on the web. In: Proc. of the 11th Internat. World Wide Web Conference, pp. 251–262. ACM Press, New York (2002)

[12] Davidov, D., Gabrilovich, E., Markovitch, S.: Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), pp. 250–257. ACM, New York (2004)

[13] Dehmer, M.: Information processing in complex networks: Graph entropy and information functionals. Applied Mathematics and Computation 201, 82–94 (2008)

[14] Dehmer, M., Emmert-Streib, F., Mehler, A., Kilian, J.: Measuring the structural similarity of web-based documents: A novel approach. International Journal of Computational Intelligence 3(1), 1–7 (2006)

[15] Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, January 6-12, pp. 1606–1611 (2007)

[16] Gupta, R., Ratinov, L.: Text categorization with knowledge transfer from heterogeneous data sources. In: AAAI 2008: Proceedings of the 23rd National Conference on Artificial Intelligence, pp. 842–847. AAAI Press, Menlo Park (2008)

[17] Halliday, M.A.K., Hasan, R.: Language, Context, and Text: Aspects of Language in a Socialsemiotic Perspective. Oxford University Press, Oxford (1989)

[18] Hotho, A., Nürnberger, A., Paaß, G.: A Brief Survey of Text Mining. Journal for Language Technology and Computational Linguistics (JLCL) 20(1), 19–62 (2005)

[19] Joachims, T.: Learning to classify text using support vector machines. Kluwer, Boston (2002)

[20] Joachims, T., Cristianini, N., Shawe-Taylor, J.: Composite kernels for hypertext categorisation. In: Proceedings of the 11th International Conference on Machine Learning, pp. 250–257. Morgan Kaufmann, San Francisco (2001)

[21] Kanaris, I., Stamatatos, E.: Webpage genre identification using variable-length character n-grams. In: Proc. of the 19th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI 2007). IEEE Computer Society Press, Washington, DC, USA (2007)

[22] Leopold, E.: Models of semantic spaces. In: Mehler, A., Köhler, R. (eds.) Aspects of Automatic Text Analysis. STUDFUZZ, vol. 209, pp. 117–137. Springer, Heidelberg (2007)

[23] Lim, C.S., Lee, K.J., Kim, G.C.: Multiple sets of features for automatic genre classification of web documents. Information Processing & Management 41(5), 1263–1276 (2005), doi: http://proxy.bnl.lu:2193/10.1016/j.ipm.2004.06.004

[24] Lindemann, C., Littig, L.: Classification of web sites at super-genre level. In: [34] (2010)

[25] Martin, J.R.: English Text. System and Structure. John Benjamins, Philadelphia (1992)

[26] Mehler, A.: Structural similarities of complex networks: A computational model by example of wiki graphs. Applied Artificial Intelligence 22(7&8), 619–683 (2008)

[27] Mehler, A.: Generalized shortest paths trees: A novel graph class applied to semiotic networks. In: Dehmer, M., Emmert-Streib, F. (eds.) Analysis of Complex Networks: From Biology to Linguistics, pp. 175–220. Wiley-VCH, Weinheim (2009)

[28] Mehler, A.: A quantitative graph model of social ontologies by example of Wikipedia. In: Dehmer, M., Emmert-Streib, F., Mehler, A. (eds.) Towards an Information Theory of Complex Networks: Statistical Methods and Applications. Birkhäuser, Boston (2010)

[29] Mehler, A.: Structure formation in the web. A graph-theoretical model of hypertext types. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, Text, Speech and Language Technology, pp. 225–247. Springer, Dordrecht (2010)

[30] Mehler, A., Waltinger, U.: Enhancing document modeling by means of open topic models: Crossing the frontier of classification schemes in digital libraries by example of the DDC. Library Hi Tech 27(4) (2009)

[31] Mehler, A., Geibel, P., Pustylnikov, O.: Structural classifiers of text types: Towards a novel model of text representation. Journal for Language Technology and Computational Linguistics (JLCL) 22(2), 51–66 (2007)

[32] Mehler, A., Gleim, R., Wegner, A.: Structural uncertainty of hypertext types. An empirical study. In: Proceedings of the Workshop "Towards Genre-Enabled Search Engines: The Impact of NLP", in conjunction with RANLP 2007, Borovets, Bulgaria, September 30, pp. 13–19 (2007)

[33] Mehler, A., Gleim, R., Ernst, A., Waltinger, U.: WikiDB: Building interoperable wiki-based knowledge resources for semantic databases. Sprache und Datenverarbeitung International Journal for Language Data Processing 32(1), 47–70 (2008)

[34] Mehler, A., Sharoff, S., Santini, M. (eds.): Genres on the Web: Computational Models and Empirical Studies. Springer, Dordrecht (2010)

[35] Menczer, F.: Lexical and semantic clustering by web links. Journal of the American Society for Information Science and Technology 55(14), 1261–1269 (2004)

[36] Meyer zu Eißen, S., Stein, B.: Genre Classification of Web Pages: User Study and Feasibility Analysis. In: Biundo, S., Frühwirth, T., Palm, G. (eds.) KI 2004. LNCS (LNAI), vol. 3228, pp. 256–269. Springer, Heidelberg (2004)

[37] Mukherjea, S.: Organizing topic-specific web information. In: Proc. of the 11th ACM Conference on Hypertext and Hypermedia, pp. 133–141. ACM, New York (2000)

[38] Power, R., Scott, D., Bouayad-Agha, N.: Document structure. Computational Linguistics 29(2), 211–260 (2003)

[39] Rehm, G.: Language-independent text parsing of arbitrary html-documents. towards a foundation for web genre identification. Journal for Language Technology and Computational Linguistics, JLCL (2005)

[40] Rehm, G.: Hypertextsorten: Definition, Struktur, Klassifikation. Phd thesis, Angewandte Sprachwissenschaft und Computerlinguistik, Justus-Liebig-Universität Gießen, JLU (2007)

[41] Rehm, G., Santini, M., Mehler, A., Braslavski, P., Gleim, R., Stubbe, A., Symonenko, S., Tavosanis, M., Vidulin, V.: Towards a reference corpus of web genres for the evaluation of genre identification systems. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008), Marrakech, Morocco (2008)

[42] Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley, Reading (1989)

[43] Santini, M.: Cross-testing a genre classification model for the web. In: [34] (2010)

[44] Santini, M., Power, R., Evans, R.: Implementing a characterization of genre for automatic genre identification of web pages. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, Association for Computational Linguistics, Morristown, NJ, USA, pp. 699–706 (2006)

[45] Santini, M., Mehler, A., Sharoff, S.: Riding the rough waves of genre on the web: Concepts and research questions. In: [34], pp. 3–32 (2010)

[46] Saunders, S.: Improved shortest path algorithms for nearly acyclic graphs. PhD thesis, University of Canterbury, Computer Science (2004)

[47] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)

[48] Sharoff, S.: In the garden and in the jungle. Comparing genres in the BNC and Internet. In: [34] (2010)

[49] Thelwall, M., Vaughan, L., Björneborn, L.: Webometrics. Annual Review of Information Science Technology 6(8) (2006)

[50] Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

[51] Waltinger, U.: On social semantics in information retrieval. PhD thesis, Bielfeld University, Germany (2010)

[52] Waltinger, U., Mehler, A.: Social semantics and its evaluation by means of semantic relatedness and open topic models. In: IEEE/WIC/ACM International Conference on Web Intelligence, Milano, September 15–18 (2009)

[53] Waltinger, U., Mehler, A., Wegner, A.: A two-level approach to web genre classification. In: Proceedings of the 5th International Conference on Web Information Systems and Technologies (WEBIST 2009), pp. 689–692. INSTICC Press, Lisboa (2009)

[54] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)

[55] Zlatic, V., Bozicevic, M., Stefancic, H., Domazet, M.: Wikipedias: Collaborative web-based encyclopedias as complex networks. Physical Review E 74, 016,115 (2006)

Part VI
Interfacing Textual Data, Ontological Resources and Document Parsing

Chapter 16
Learning Semantic Relations from Text

Gerhard Heyer

Abstract. The paper presents a co-occurrence based approach to extracting semantic relations from text. We concentrate on Semantic Relations as relations among concepts, and instances of such relations, as used in taxonomies and ontologies. We show how typed semantic relations can be derived from association networks by filters based on linguistic and non-linguistic knowledge. The main point of the paper is to argue that there is no single step derivation of knowledge about semantic relations. Learning semantic relations from text requires linguistic and non-linguistic knowledge sources of different kinds and quality that need to iteratively interact in order to derive high quality knowledge about semantic relations.

16.1 Introduction

Learning *Semantic Relations* (SR) from Text is increasingly important to content engineering, in particular text mining, knowledge management and ontology generation. While the notion of SR is a key notion to all approaches that attempt to automatically process semantic information in texts, the very notion of SR in the literature covers a broad range of topics, including

- relations among concepts, e.g. superordinates, subordinates, synonyms, antonyms etc. (*taxonomies*, *ontologies*),
- instances of such relations,
- type restrictions in syntactic constructs such as N-V, A-V, N Conjunction N etc. (*dependency grammars*, *syntactic patterns*),
- instances of such relations, and

Gerhard Heyer
Universität Leipzig, Institut für Informatik, Abteilung Automatische Sprachverarbeitung
e-mail: heyerasv@informatik.uni-leipzig.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 333–346.
springerlink.com

- untyped associations among concepts and wordforms (*semantic networks*, *co-occurrence graphs*).

Pattern matching techniques, including Hearst patterns [20] and systems such as OntoLearn [26], KnowItAll [18], and SRES (Self-Supervised Relation Extraction System) [19], figure prominently in extracting semantic relations from text. Other approaches document subsumption and sequence modeling [28] and systems based on hidden Markov models (HMM) and conditional random fields (CRF) [14], as well as classification with rich features and kernel methods [33, 13, 10].

Pattern matching techniques, including Hearst patterns [20] and systems such as OntoLearn [26], KnowItAll [18], and SRES (Self-Supervised Relation Extraction System) [19], figure prominently in extracting semantic relations from text. Other approaches include document subsumption and sequence modeling [28] and systems based on hidden Markov models (HMM) and conditional random fields (CRF) [14], as well as classification with rich features and kernel methods [33, 13, 10].

In what follows, we shall concentrate on *Semantic Relations* as relations among concepts, and instances of such relations, as used in taxonomies and ontologies, and attempt to present a unified view on learning semantic relations from text by showing how SR can be derived by co-occurrence networks generated from instances of words (*wordforms* for short) encountered in text. Typed semantic relations can be derived from such association networks by filters based on linguistic and non linguistic knowledge, see [21, 22]. Complementary to these results, the main point of this paper is to argue that there is *no single step* derivation of knowledge about semantic relations. Rather, we claim, learning semantic relations from text requires an *architecture* for integrating linguistic and non linguistic knowledge of different kind and quality. These knowledge sources need to *iteratively interact* in order to derive high quality knowledge about semantic relations. In what follows, this architecture will be based on a web service oriented architecture for iteratively generating SR for different domains.

We begin by discussing the structuralist foundations of learning semantic relations. We then elaborate on filters on co-occurrence networks, in particular word sense disambiguation (WSD), and so-called higher order co-occurrences, and finally sketch an architecture for integrating different knowledge sources that are needed for deriving semantic relations.

16.2 How Is It Possible to Automatically Process *Meaning*?

Language is not a random concatenation of words. Following the famous Swiss linguist Ferdinand de Saussure, meaning (and other notions of linguistic description) can be defined only on the basis of structural relations amongst the words of a language [29]. Linguistic Structuralism was developed and applied to understand hitherto unknown languages. In contrast to other approaches, however, reference is not the key notion for explaining and computing meaning. Following Saussure, the fundamental structural relations are syntagmatic and paradigmatic relations. In what follows we take up that idea and reconstruct it in a formal way. As a general proviso to the reader it should be pointed out, however, that by this reconstruction we are always talking about statistical distributions of wordforms, and hence a *reinterpretation* of the original Saussurian notions that only partially corresponds to the usage of these notions among linguists. In particular, the basis of our reconstruction are statistically significant co-occurrences of wordforms that we call the ***global context*** of a wordform. Such global contexts contain ***statistically significant syntagmatic***[1] relations that are then used to compute ***statistically significant paradigmatic*** relations. Hence, by our reconstruction, ***statistically significant paradigmatic*** relations are based on, and are not complementary to, ***statistically significant syntagmatic*** relations, which may deviate from some linguists expectations.

Let $L = (W, S)$ be a language with W the set of *wordforms* of this language, and S the set of sentences actually encountered in a text corpus.[2]

Each sentence $s_i \in S$ from L then represents a set of wordforms, $s_i = \{w_1, \ldots, w_n\}$, assuming that all words belong to the set of wordforms of L, i.e. for each $w_i : w_i \in W$.

Now, the ***local context*** of a wordform w_i comprises the set of all wordforms occurring together in a sentence (or text window) with w_i less w_i itself:

$$K_s(w_i) = \{w_1, \ldots, w_i, \ldots, w_n\} \backslash \{w_i\} \tag{16.1}$$

Syntagmatic and paradigmatic relations of two wordforms w_i and w_k can then be derived by considering the joint *appearance* and joint *context*, respectively.

The ***syntagmatic*** relation, $SYN(w_i, w_j)$, which is symmetrical, holds if and only if there exists a local context for one of the wordforms in which the other appears:

$$\exists K_s(w_i)(w_j \in K_s(w_i) \leftrightarrow SYN(w_i, w_j)) \tag{16.2}$$

Statistically significant co-occurrences of wordforms will be reflected in terms of frequency. Hence, there will be an expectation value X that states the absolute number of joint occurrences that must have been observed in relation to the number of all local contexts. To define this value, parameters like the frequency of wordforms, their Zipf-Distribution, the size of the text corpus etc. can be taken into account, see for example [17, 12, 30]. In effect, for any wordform w_i that is part of the local contexts of another wordform w it can be decided whether or not w_i is a *significant* constituent of these contexts based on the number of such contexts, $SIG(w_i, K(w))$:

$$SIG(w_i, K(w)) \leftrightarrow \mid \{K(w_i) | w_i \in K(w)\} \mid > X \tag{16.3}$$

[1] We use asterisks in addition to bold print to indicate that the term put between asterisks has a technical meaning as defined in this paper that may be deviant from its common or intuitive meaning in a way as Lyons has done in his books on semantics [25].

[2] Notice that we do not make any presumptions here on the syntactic or semantic validity of the sentences. A sentence is any sequence of wordforms that we wish to mark as sentence.

A ***statistical syntagmatic*** relation $SYNS(w_i, w)$ between two wordforms w_i, $w \in W$ holds if and only if the wordform w_i is a *significant* constituent of the local contexts of the wordform w:

$$SYNS(w_i, w) \leftrightarrow SIG(w_i, K(w)) \tag{16.4}$$

We compute ***statistical syntagmatic*** relations of a wordform by computing the set of its co-occurrences. In the literature, co-occurrences are being computed using statistical measures like *Tanimoto-similarity* (proportion of joint occurrence versus single occurrence), *mutual information* (deviance from statistical independence, degree of surprise), and the *Poisson Measure* (probability of simultaneous rare events), see [5] for a comprehensive survey and comparison.

Statistical syntagmatic relations typically comprise syntactic phenomena like

dependencies (*dog barks*, *Professor teaches*, ...),
enumerations (*Faculty of Mathematics and CS*, ...), and
idioms (*with kind regards*, ...).

When the distance of words related by the ***statistical syntagmatic*** relation is restricted to the immediate left or right neighbors of a word, the relation also covers

multiwords (*Royal Air Force*, ...),
head-modifier relations (*automatic indexing*, ...), and
category and function descriptions (*president Obama*, ...).

By the *global context* $K_G(w)$ of a wordform w we now refer to the set of all wordforms w_i with which the wordform w stands in ***statistical syntagmatic*** relation $SYNS(w_i, w)$:

$$K_G(w) = \{w_i | SYNS(w_i, w)\} \tag{16.5}$$

In effect, the global context of a wordform is a set of wordforms that is equivalent to the set of its co-occurrences computed on all sentences of a corpus.

Hence, by this reconstruction, when computing the co-occurrences, or ***statistical syntagmatic*** relations, of a wordform, we compute the global context of a wordform.

Now, the global context of a wordform comprises both, the set of wordforms with which it statistically significant co-occurs, i.e. the set of ***statistical syntagmatic*** relations, as well as the set of wordforms that share a similar context and therefore sometimes co-occur. Figure 16.1, depicting the most significant elements of the global context of "Dublin" based on British newspaper text[3], may illustrate the point.[4] While "Dublin Castle", "City (of) Dublin", "Dublin Co.", or "Trinity College Dublin" can be considered typical ***statistical syntagmatic*** relations of

[3] Textual data based on Leipzig Corpora Collection 2007, http://www.corpora.uni-leipzig.de

[4] Computation of co-occurrences based on Poisson Measure, visualization based on simulated annealing, see [16]. Line thickness represents the significance of the co-occurrence.

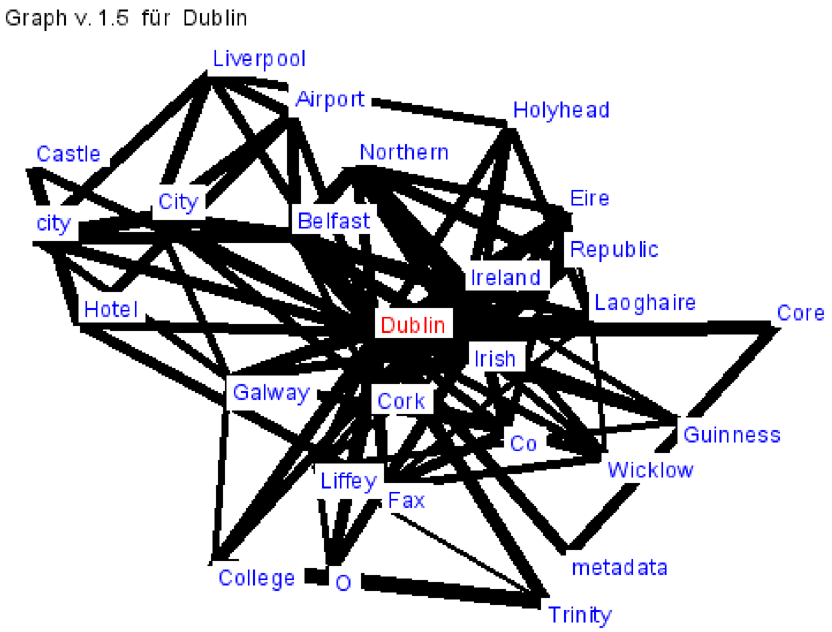


Fig. 16.1 Global context of “Dublin”, British newspaper text.

“Dublin”, “Belfast”, “Cork”, and “Galway”, being other Irish cities, represent co-hyponyms to “Dublin” that must have co-occurred in the sentences of the corpus on which the co-occurrences have been computed, e.g. by coordination or compound constructions.

As one might expect, the global context of a wordform crucially depends on the text input that is used for computing it. Figures 16.2 and 16.3 exemplify different global contexts of “Dublin” with respect to different collections of text data as input. Figure 16.2 is computed based on German newspaper text, while Figure 16.4 is based on German Wikipedia links.

Although global contexts of wordforms sometimes include other wordforms that they are related to in a paradigmatic way, such as co-hyponyms, for the computation of ***statistical paradigmatic*** relations we need to exploit the fact that wordforms that stand in a paradigmatic relation share similar contexts. Given a corpus and a set of wordforms for which we have computed their global contexts, ***statistical paradigmatic*** relations among these wordforms can now be derived when we compare their global contexts with regard to similarity. We assume a comparison operator $SIM(K_G(w_i),K_G(w_j))$ that compares global contexts and yields the value 1 in all those cases where the similarity is “similar enough”, and 0 in all other cases. In what follows, for simplicity’s sake we write $SIM(K_G(w_i),K_G(w_j))$ in the case the similarity operation on two contexts yields the value 1. Two wordforms w_i w_j can

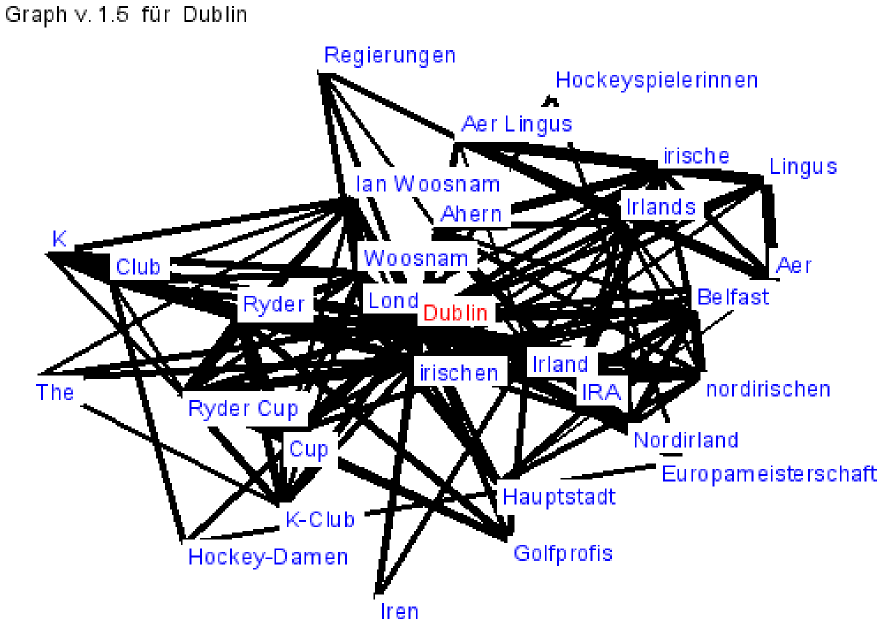


Fig. 16.2 Global context of “Dublin”, German newspaper text.

then be said to stand in a ***statistical paradigmatic*** relation $PARAS(w_i, w_j)$ if and only if their global contexts are similar to each other:

$$SIM(K_G(w_i), K_G(w_j)) \leftrightarrow PARAS(w_i, w_j) \tag{16.6}$$

Examples of comparison operator instances include known similarity measures like the Tanimoto measure, the cosine (global contexts can be interpreted as vectors), Euclidian distance, and many more. The question which measure is the “best” one, cannot be answered easily. On the one hand, this is due to evaluation difficulties, on the other hand it is also important to ask for the desired results – which ***statistical paradigmatic*** relations are to be calculated by some instance of a similarity measure. Details of this discussion are not crucial for the model introduced here, and are left open for further research. It should be noted, however, that there is no uniform usage of the notion of similarity in the literature. Quite often ‘word similarity’ is used to refer to a comparison of the global contexts of a given word, whereas sometimes it is also – misleadingly – used to refer to the co-occurrence measure of ***statistical syntagmatic*** relations, as similar words are returned by such a computation [cf. 31, 15, 8, 32].

The set of wordforms that are related to a word by way of a ***statistical paradigmatic*** relation varies depending on the similarity relation that we use to compare the global contexts. As a basic step we can consider wordforms that belong to the same **syntactic category** CAT:

$$K_G(w_i)_{CAT} = \{w | w_i, w \in W \& SYNS(w_i, w) \& CAT(w) = CAT(w_i)\} \tag{16.7}$$

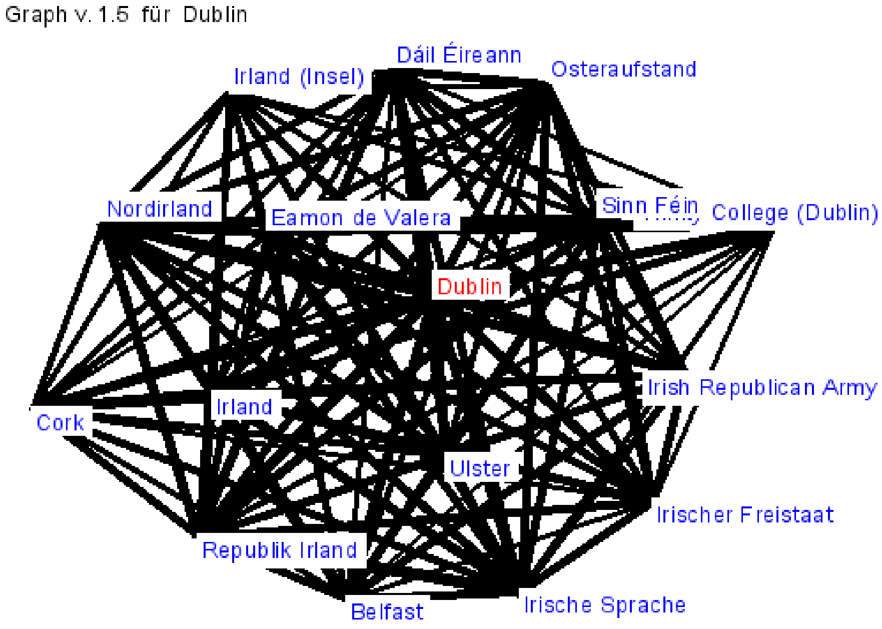


Fig. 16.3 Global context of "Dublin", German Wikipedia links.

$$SIM(K_G(w_i)_{CAT}, K_G(w)_{CAT}) \leftrightarrow PARAS_{CAT}(w_i, w) \tag{16.8}$$

The categories used can either be taken from a dictionary as an external source of linguistic knowledge, or derived by supervised or unsupervised tagging. In effect, this filter yields lists of words sharing the same distributional classes [23, 24]:

$$\{w|CAT(w) = CAT(w_i)\&PARA_{CAT}(w_i, w)\} \tag{16.9}$$

Logical relations like *superordinate*, and *synonym*, or other relations used in taxonomies and ontologies, can be derived from global contexts by admitting only those elements that fulfill a certain logical condition *LOG*:

$$K_G(w_i)_{LOG} = \{w|w_i, w \in W\&SYNS(w_i, w)\&LOG(w) = LOG(w_i)\} \tag{16.10}$$

$$SIM(K_G(w_i)_{LOG}, K_G(w)_{LOG}) \leftrightarrow PARAS_{LOG}(w_i, w) \tag{16.11}$$

Again, the particular logical relations used and their instantiations in the domain of application may either be based on external knowledge sources, such as a domain specific thesaurus, or derived by an iterative process of supervised or unsupervised learning. In effect, the set of wordforms related to a word by way of a ***statistical paradigmatic*** relation is filtered with respect to predefined logical relations:

$$\{w|LOG(w) = LOG(w_i \& PARAS_{LOG}(w_i, w)\} \qquad (16.12)$$

In general, we propose that different kinds of semantic relations can be reconstructed as ***statistical paradigmatic*** relations using constraints, or filters, on the similarity measure employed to compare global contexts. Hence, even for ***statistical paradigmatic*** relations, ***statistical syntagmatic*** relations form the basis. When it comes to computing particular semantic relations, this reconstruction also gives a hint on how to proceed: Start with the computation of co-occurrences as ***statistical syntagmatic*** relations and iteratively add more linguistic and non linguistic knowledge in order to filter out the semantic relations that are of interest in a particular content engineering application. Logical relations in this process of filtering ***statistical syntagmatic*** relations then are the result of high quality refinements, and require numerous linguistic and non linguistics knowledge sources.

16.3 Some Filters for Semantic Relations

In order illustrate the idea of iteratively filtering and refining textual data by comparing the global context of words in order to derive semantic relations, let us look in detail at an approach to word sense disambiguation, and the clustering of semantically similar words.

Taking up the above definitions, let us consider the global context of the word *space*. This word is ambiguous. It covers at least the areas of computers, real estate, and physics. As the global context of a wordform also contains the linguistic contexts in which the word is being used, the visualization of an ambiguous word's global context should make these different contexts transparent. This is indeed the case, as can be seen from Figure 16.4. We clearly recognize the three different meanings *real estate*, *computer hardware*, and *missiles*, where the connection between *address* and *memory* results from the fact that *address* again is a polysemous concept.

Based on the co-occurrences of word forms and the small-world property of their co-occurrences graph, an approach to solve the polysemy problem has been introduced by [4, 7]. The algorithm is based on the assumption that if three words are semantically homogenous, they share all, or most, of their global context, and hence are located in the same cluster of the graph. It proceeds by taking the first 15 co-occurrences of a wordform *w*, ordered by co-occurrence significance, and generating all combinations of *w* and any two of its 15 co-occurrences (i.e. in total 105 different triples of words). The intersections of co-occurrences are retrieved and clustered by a hierarchical agglomerative clustering. This is repeated until no co-occurrences are left above a predefined threshold.

As a result, for a given word one or more sets of semantically homogeneous words are found along with a set of words which are either semantically unrelated to the input word (although they are co-occurring with it), or they do not pass the threshold to allow for a reliable decision. Problems occur when the text corpus does not reflect specific contexts, or is unbalanced with respect to certain sub-languages.

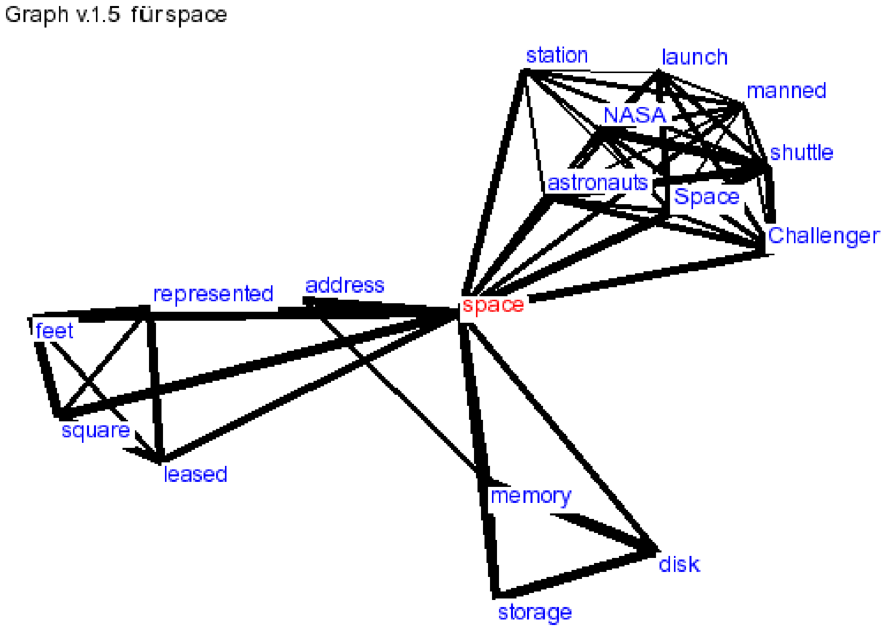
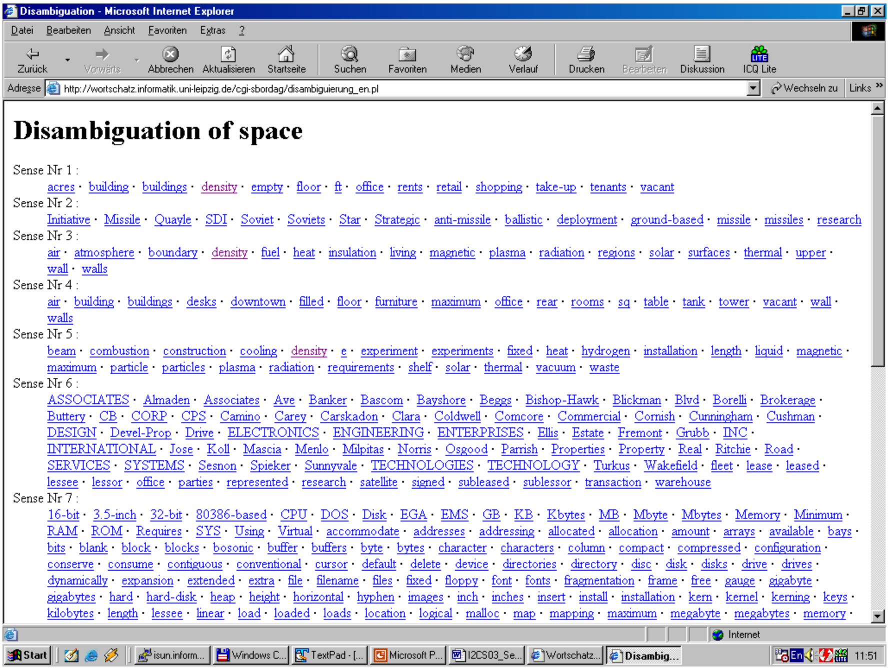


Fig. 16.4 Co-occurrence Graph for space.

The following Figure 16.5 depicts the result of the disambiguation filter for the word *space*:

A second example is the idea of co-occurrences of higher order [2, 7]. When considering ***statistical syntagmatic*** relations, we compute the set of wordforms that significantly often co-occur in *sentences*. These sentence-based co-occurrences of wordforms are called *first order* co-occurrences. *Second-order* co-occurrences of wordforms are sets of wordforms that significantly often co-occur in first order co-occurrence sets. In general, co-occurrences of nth order are sets of wordforms that significantly often co-occur in co-occurrence sets of $(n-1)$th order. When calculating a higher order, the significance values of the preceding order are not relevant. A co-occurrence set consists of the n highest ranked co-occurrences of a wordform (calculated by a particular co-occurrence measure). In general, the sets of higher order co-occurrences remain more or less stable after some iterations. While the sets are somewhat semantically homogeneous, they sometimes have nothing to do with the reference word. If two wordforms iteratively share similar global contexts, the elements of the iterated co-occurrence sets will overlap to a high degree. Figure 16.6 shows the intersection of higher-order co-occurrences for two selected wordforms in German taken as start set:[5]

[5] Thanks to Chris Biemann for providing me with these examples.


Disambiguation - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Medien Verlauf Drucken Bearbeiten Diskussion ICQ Lite
Adresse
Wechseln zu Links
Internet
Start isun.inform... Windows C... TextPad - [... Microsoft P... I2CS03_Se... Wortschatz... Disambig... En 11:51

http://wortschatz.informatik.uni-leipzig.de/cgi-sbordag/disambiguierung_en.pl

Disambiguation of space

Sense Nr 1 :
acres · building · buildings · density · empty · floor · ft · office · rents · retail · shopping · take-up · tenants · vacant

Sense Nr 2 :
Initiative · Missile · Quayle · SDI · Soviet · Soviets · Star · Strategic · anti-missile · ballistic · deployment · ground-based · missile · missiles · research

Sense Nr 3 :
air · atmosphere · boundary · density · fuel · heat · insulation · living · magnetic · plasma · radiation · regions · solar · surfaces · thermal · upper · wall · walls

Sense Nr 4 :
air · building · buildings · desks · downtown · filled · floor · furniture · maximum · office · rear · rooms · sq · table · tank · tower · vacant · wall · walls

Sense Nr 5 :
beam · combustion · construction · cooling · density · e · experiment · experiments · fixed · heat · hydrogen · installation · length · liquid · magnetic · maximum · particle · particles · plasma · radiation · requirements · shelf · solar · thermal · vacuum · waste

Sense Nr 6 :
ASSOCIATES · Almaden · Associates · Ave · Banker · Bascom · Bayshore · Beggs · Bishop-Hawk · Blickman · Blvd · Borelli · Brokerage · Buttery · CB · CORP · CPS · Camino · Carey · Carskadon · Clara · Coldwell · Comcore · Commercial · Cornish · Cunningham · Cushman · DESIGN · Devel-Prop · Drive · ELECTRONICS · ENGINEERING · ENTERPRISES · Ellis · Estate · Fremont · Grubb · INC · INTERNATIONAL · Jose · Koll · Mascia · Menlo · Milpitas · Norris · Osgood · Parrish · Properties · Property · Real · Ritchie · Road · SERVICES · SYSTEMS · Sesnon · Spieker · Sunnyvale · TECHNOLOGIES · TECHNOLOGY · Turkus · Wakefield · fleet · lease · leased · lessee · lessor · office · parties · represented · research · satellite · signed · subleased · sublessor · transaction · warehouse

Sense Nr 7 :
16-bit · 3.5-inch · 32-bit · 80386-based · CPU · DOS · Disk · EGA · EMS · GB · KB · Kbytes · MB · Mbyte · Mbytes · Memory · Minimum · RAM · ROM · Requires · SYS · Using · Virtual · accommodate · addresses · addressing · allocated · allocation · amount · arrays · available · bays · bits · blank · block · blocks · bosonic · buffer · buffers · byte · bytes · character · characters · column · compact · compressed · configuration · conserve · consume · contiguous · conventional · cursor · default · delete · device · directories · directory · disc · disk · disks · drive · drives · dynamically · expansion · extended · extra · file · filename · files · fixed · floppy · font · fonts · fragmentation · frame · free · gauge · gigabyte · gigabytes · hard · hard-disk · heap · height · horizontal · hyphen · images · inch · inches · insert · install · installation · kern · kernel · kerning · keys · kilobytes · length · lessee · linear · load · loaded · loads · location · logical · malloc · map · mapping · maximum · megabyte · megabytes · memory ·

Fig. 16.5 Automatic disambiguation of *space*.

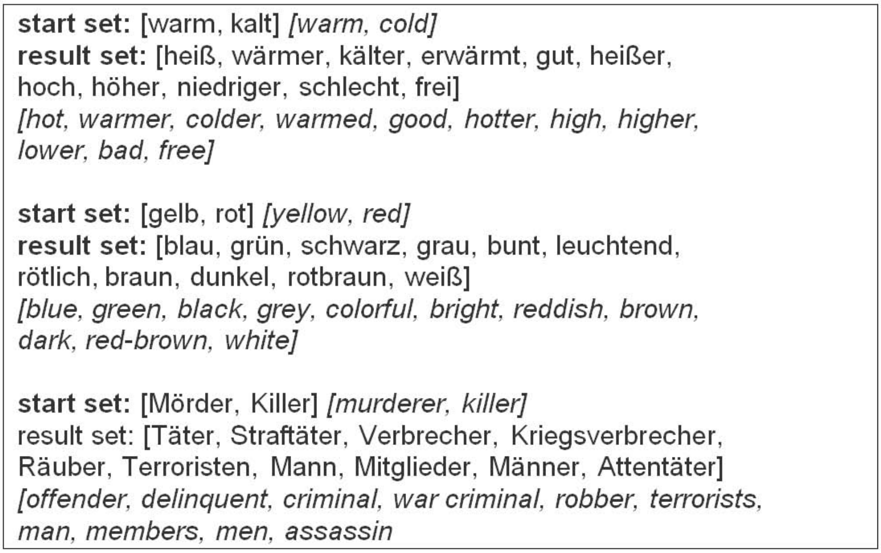
16.4 An Architecture for Learning Semantic Relations

The general architecture that we propose now for learning semantic relations is open to iteratively integrate different knowledge sources, internal as well as external ones, and allows different tools to iteratively interact. It takes a modular approach to language data and algorithms and is based on

1. a strict detachment of language processing algorithms from particular linguistic data and languages, thus encouraging the collection of language data and corpora, and the development of processing algorithms independently from each other;
2. language independent processing algorithms, thus allowing for a comparison of languages with respect to certain statistical parameters and other computed features; and
3. a modular software design, thus enabling algorithms to enrich language data and to re-use enriched data and features as well as adding new algorithms.

Figure 16.7 illustrates the key elements. After preprocessing a text input, text is iteratively being processed using linguistic processing tools, external linguistic and non-linguistic knowledge bases, and machine learning tools.

From a software engineering point of view, the increasing availability of statistical natural language processing methods and large text corpora for many languages



start set: [warm, kalt] *[warm, cold]*
result set: [heiß, wärmer, kälter, erwärmt, gut, heißer, hoch, höher, niedriger, schlecht, frei]
[hot, warmer, colder, warmed, good, hotter, high, higher, lower, bad, free]

start set: [gelb, rot] *[yellow, red]*
result set: [blau, grün, schwarz, grau, bunt, leuchtend, rötlich, braun, dunkel, rotbraun, weiß]
[blue, green, black, grey, colorful, bright, reddish, brown, dark, red-brown, white]

start set: [Mörder, Killer] *[murderer, killer]*
result set: [Täter, Straftäter, Verbrecher, Kriegsverbrecher, Räuber, Terroristen, Mann, Mitglieder, Männer, Attentäter]
[offender, delinquent, criminal, war criminal, robber, terrorists, man, members, men, assassin

Fig. 16.6 Intersections of higher-order co-occurrences.

has lead to the somewhat paradoxical situation that more than ever language data and algorithms for processing them are available, but most of them are incompatible with each other in terms of data structures and interfaces. We surpass this obstacle by taking an implementation approach based on *service-oriented architectures (SOA)*. SOA can be defined as an application oriented architecture within which all functions are defined as independent services with well-defined invocable interfaces [11, 27]:

1. all functions are defined as services,
2. all services are independent, i.e., they return the expected result without external components needing to make any assumptions of their 'inner workings'; and
3. the interfaces of a service can be invoked, i.e., at an architectural level, it is irrelevant whether the interfaces are local or remote, or what interconnect protocol is used to effect the invocation, or what infrastructure components are required to make the connection.

An application is represented as a set of services that communicate through the exchange of messages. There now are widely accepted standards for describing service interfaces such as WSDL and the transfer of those messages using SOAP. The general architecture sketched above has been implemented based on a service oriented architecture by making language resources and language processing tools available as a *service*. As data we use linguistic resources from the *Leipzig Corpora*

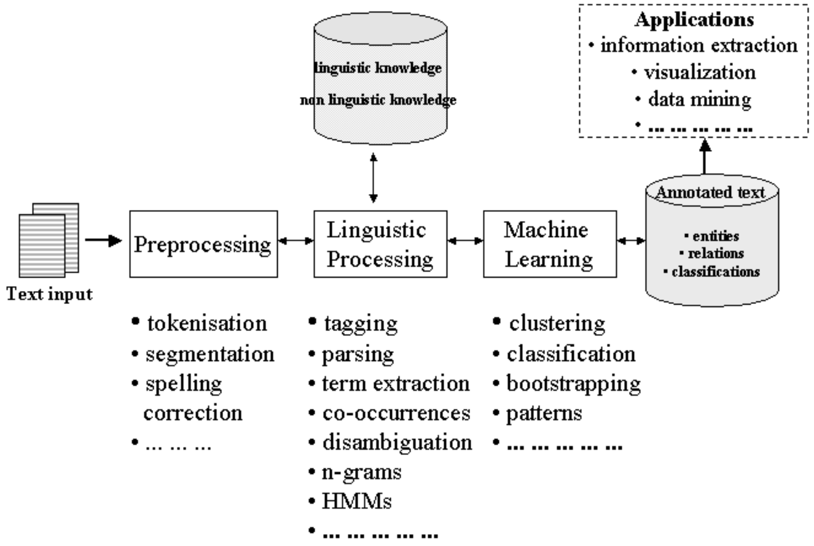


Fig. 16.7 General architecture for interactively learning semantic relations.

Collection (LCC), a collection of text corpora based on a set size and format[6], and other data provided within the CLARIN[7] network of linguistic resources. Algorithms for linguistic processing and machine learning for textual data are provided by the *ASV Toolbox*[8], a modular tool to explore statistical properties of natural languages, and a framework to combine and integrate algorithms for natural language processing [3]. A web service access to these digital text and lexical data as well as NLP algorithms and tools is running very reliably since 2004 [9][9]. The services are based on a very large frequency sorted dictionary of German word forms including POS information, sample sentences and co-occurrence statistics. For learning semantic relations from text, tools for automatic terminology extraction, synonyms and similar words computed on a word's co-occurrence profiles, as well as graph based clustering are being offered to the scientific community that allow for an iterative application of the different tools and knowledge sources.

[6] *Leipzig Corpora Collection*, see http://corpora.informatik.uni-leipzig.de/

[7] CLARIN is a large-scale pan-European collaborative effort to create, coordinate and make language resources and technology available and readily usable, see http://www.clarin.eu/

[8] ASV Toolbox, see http://www.asv.informatik.uni-leipzig.de/asv/methoden

[9] *Leipzig Linguistic Services* (LLS), see http://wortschatz.uni-leipzig.de/Webservices/

References

[1] Biemann, C.: Kookkurrenzen höherer Ordnung. In: Heyer, G., Quasthoff, U., Wittig, T. (eds.) Text Mining: Wissensrohstoff Text, W3L, Herdecke, pp. 161–167 (2006)

[2] Biemann, C., Bordag, S., Quasthoff, U.: Automatic acquisition of paradigmatic relations using iterated co-occurrences. In: Proceedings of LREC 2004, ELRA, Lisboa, Portugal (2004)

[3] Biemann, C., Quasthoff, U., Heyer, G., Holz, F.: ASV Toolbox – a modular collection of language exploration tools. In: Proceedings of LREC 2008, ELRA, Marrakech, Morocco (2008)

[4] Bordag, S.: Sentence co-occurrences as small-world-graphs: A solution to automatic lexical disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 329–333. Springer, Heidelberg (2003)

[5] Bordag, S.: Elements of knowledge free and unsupervised lexical acquisition. PhD thesis, University of Leipzig, Computer Science Department (2007)

[6] Bordag, S., Heyer, G.: A structuralist framework for quantitative linguistics. In: Mehler, A., Köhler, R. (eds.) Aspects of Automatic Text Analysis. Springer, Heidelberg (2005)

[7] Bordag, S., Heyer, G., Quasthoff, U.: Small worlds of concepts and other principles of semantic search. In: Böhme, T., Heyer, G., Unger, H. (eds.) IICS 2003. LNCS, vol. 2877, pp. 10–19. Springer, Heidelberg (2003)

[8] Brown, P.F., de Souza, P.V., Mercer, R.L., Watson, T.J., Della Pietra, V.J., Lai, J.C.: Class-based n-gram models of natural language. Computational Linguistics 18, 467–479 (1992)

[9] Büchler, M., Heyer, G.: Leipzig Linguistic Services – a 4 years summary of providing linguistic web services. In: Proceedings of the Conference on Text Mining Services (TMS 2009) Leipzig University, Leipzig, Band XIV, Leipziger Beiträge zur Informatik (2009)

[10] Bunescu, R.C., Mooney, R.J.: Statistical relational learning for natural language information extraction. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning, pp. 535–552. MIT Press, Cambridge (2007)

[11] Channabasavaiah, K., Holley, K., Tuggle, E.: SOA is more than web services (2004), http://www.looselycoupled.com/opinion/2004/chann-soa-infr0630.html
(last accessed: March 03, 2009)

[12] Church, K.W.: One term or two? In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR 1995, pp. 310–318. ACM, New York (1995)

[13] Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. ACL 2004, Morristown, NJ (2004)

[14] Culotta, A., McCallum, A., Betz, J.: Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp. 296–303 (2006)

[15] Dagan, I., Lee, L., Pereira, F.C.N.: Similarity-based models of word cooccurrence probabilities. Machine Learning 34, 43–69 (1999)

[16] Davidson, R., Harel, D.: Drawing graphs nicely using simulated annealing. ACM Transactions on Graphics 15(4), 301–331 (1996)

[17] Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1993)

[18] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence 165, 91–134 (2005)

[19] Feldman, R., Sanger, J.: The Text Mining Handbook. Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, Cambridge (2007)

[20] Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992), Nantes, France, August 23-28, pp. 539–545 (1992)

[21] Heyer, G., Läuter, M., Quasthoff, U., Wittig, T., Wolff, C.: Learning relations using collocations. In: Maedche, A., Staab, S., Nedellec, C., Hovy, E.H. (eds.) JCAI 2001 Workshop on Ontology Learning. CEUR Workshop Proceedings, Seattle, USA, August 4, vol. 38 (2001), CEUR WS.org

[22] Heyer, G., Quasthoff, U., Wittig, T.: Text Mining: Wissensrohstoff Text. W3L, Herdecke (2006)

[23] Lin, D.: Using collocation statistics in information extraction. In: Proceedings of the 7th Message Understanding Conference 1998, MUC-7 (1998)

[24] Lin, D., Zhao, S., Qin, L., Zhou, M.: Identifying synonyms among distributionally similar words. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 1492–1493. Morgan Kaufmann Publishers Inc., San Francisco (2003)

[25] Lyons, J.: Semantics, vol. I and II. Cambridge University Press, Cambridge (1977)

[26] Navigli, R., Velardi, P., Cucchiarelli, R., Neri, F.: Extending and enriching WordNet with OntoLearn. In: Proc. of the GWC 2004, pp. 279–284. Springer, Heidelberg (2004)

[27] Ort, E.: Service-oriented architecture and web services: Concepts, technologies, and tools (2005),
http://java.sun.com/developer/technicalArticles/WebServices/soa2/SOATerms.html
(last accessed: March 3, 2009)

[28] Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: SIGIR 1999: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 206–213. ACM, New York (1999)

[29] de Saussure, F.: Grundfragen der allgemeinen Sprachwissenschaft. De Gruyter, Berlin (1967)

[30] Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2002, pp. 32–41. ACM, New York (2002)

[31] Terra, E., Clarke, C.L.A.: Frequency estimates for statistical word similarity measures. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL 2003 Association for Computational Linguistics. Morristown, NJ, USA, vol. 1, pp. 165–172 (2003)

[32] Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL 2002, Association for Computational Linguistics, Morristown, NJ, USA, pp. 417–424 (2002)

[33] Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. Journal of Machine Learning Research 3, 1083–1106 (2003)

Chapter 17
Modelling and Processing Wordnets in OWL

Harald Lüngen, Michael Beißwenger, Bianca Selzam, and Angelika Storrer

Abstract. In this contribution, we discuss and compare alternative options of modelling the entities and relations of wordnet-like resources in the Web Ontology Language OWL. Based on different modelling options, we developed three models of representing wordnets in OWL, i.e. the instance model, the class model, and the metaclass model. These OWL models mainly differ with respect to the ontological status of lexical units (word senses) and the synsets. While in the instance model lexical units and synsets are represented as individuals, in the class model they are represented as classes; both model types can be encoded in the dialect OWL DL. As a third alternative, we developed a metaclass model in OWL FULL, in which lexical units and synsets are defined as metaclasses, the individuals of which are classes themselves. We apply the three OWL models to each of three wordnet-style resources: (1) a subset of the German wordnet GermaNet, (2) the wordnet-style domain ontology TermNet, and (3) GermaTermNet, in which TermNet technical terms and GermaNet synsets are connected by means of a set of "plug-in" relations. We report on the results of several experiments in which we evaluated the performance of querying and processing these different models: (1) A comparison of all three OWL models (class, instance, and metaclass model) of TermNet in the context of automatic text-to-hypertext conversion, (2) an investigation of the potential of the GermaTermNet resource by the example of a wordnet-based semantic relatedness calculation.

Harald Lüngen
Institut für Deutsche Sprache, Programmbereich Korpuslinguistik, R5, 6-13,
D-68161 Mannheim, Germany
e-mail: `luengen@ids-mannheim.de`

Michael Beißwenger · Bianca Selzam · Angelika Storrer
Institute for German Language and Literature, Technische Universität Dortmund,
Emil-Figge-Straße 50, D-44221 Dortmund, Germany
e-mail: {`michael.beisswenger,bianca.stockrahm,`
`angelika.storrer`}`@tu-dortmund.de`

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 347–376.
springerlink.com

17.1 Research Context and Motivation

Wordnets are lexical resources that follow the design principles of the English Princeton WordNet [16]. Many applications of natural language processing and information retrieval use wordnets not only as a lexical database but also as an ontological resource representing conceptual knowledge. In order to make use of this conceptual knowledge in the context of semantic-web-related research, several approaches have been put forward that aim at representing the English Princeton WordNet[1] in the Web Ontology Language OWL or RDFS (cf. Section 17.3.2).

The goal of this paper is to discuss and compare different options to represent the basic entities and relations of wordnets using OWL. For this purpose, we developed alternative OWL models on three wordnet-style resources which we have been using as lexical resources in the context of two projects of the DFG research group 437 "Text-Technological Modelling of Information" (http://www.text-technology.de): (1) a subset of the German wordnet GermaNet[2], (2) the wordnet-style thesaurus TermNet [5] representing technical terms from the domains of text-technology and hypertext research, and (3) the GermaTermNet representing relations between TermNet technical terms and GermaNet synsets. In Section 17.2, we briefly describe these resources; in Section 17.3 we present the alternative OWL models that we created for these resources. The main difference between these models lies in the ontological status of the two main entity types of wordnets: the lexical units (LU) (words) and the synsets (collections of synonymous or near-synonymous lexical units).

In Section 17.4, we compare and evaluate the three models w.r.t. their performance in a set of querying tasks as occurring in the text-technological applications that we envisage. We report the results of several experiments regarding the performance of the three OWL models (1) in the context of automatic hyperlinking and (2) in the context of calculating semantic relatedness measures on the GermaTermNet metaclass model. In particular, we are interested in how easy or how complicated it is to formulate the queries for each model using nRQL and Prolog as query languages, and in whether and how the processing time in calculating the answers differs. In addition, we test the performance and feasibility of our plug-in approach that connects the general language resource GermaNet with the domain-specific TermNet. On the basis of the results reported in Section 17.4 we discuss the advantages and drawbacks of the three model types in the application contexts. The paper should thus help readers who deal with a similar ontological modelling task to choose the model that is most suitable for their project.[3]

[1] http://wordnet.princeton.edu

[2] http://www.sfs.uni-tuebingen.de/GermaNet/

[3] The OWL resources are available for download on the website http://www.wordnets-in-owl.de

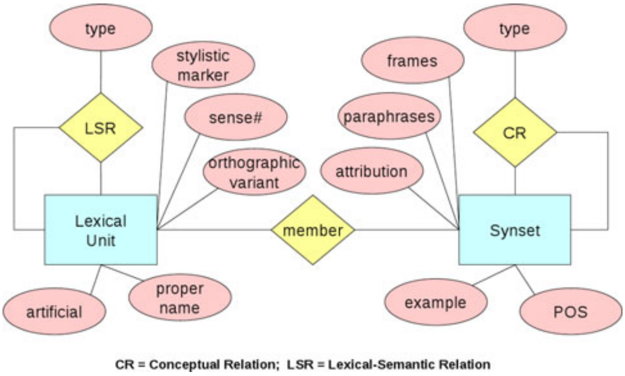


Fig. 17.1 E-R graph for GermaNet.

17.2 Resources

17.2.1 GermaNet

GermaNet is a lexical-semantic wordnet for German which was developed and is maintained at the University of Tübingen [cf. 21]. It was designed largely according to the principles of the Princeton WordNet [PWN, cf. 16] and covers the most important and frequent general language concepts and relations between the concepts and lexical units like hyponymy, meronymy and antonymy.

The central unit of representation in a wordnet is the *synonym set* (synset), in which those synonymous *lexical units* that are interchangeable in a given context are combined. The synset {*öffnen, aufmachen*}, by example, is represented as a concept node with a number of relational links: to its hyperonym synset {*wandeln, verändern*}, and to its several hyponym synsets e.g. {*aufstoßen*}. Moreover, a causation relation with the synset {*(sich) öffnen, aufgehen*} holds. Finally, the lexical unit *öffnen* is related to its antonym *schließen*.

The data model for GermaNet is shown in the entity-relationship graph in Figure 17.1[4]. The lexical objects are shown in rectangles, and the attributes that characterise them are shown in ellipses. Relations between the objects are marked as diamonds: conceptual relations (CR) like hyponymy hold between synsets, and lexical-semantic relations (LSR) like antonymy hold between lexical units, and the lexicalisation relation (“member”) holds between lexical units and synsets.

GermaNet 5.0 contains about 53000 synsets and 76000 lexical units. Although we have converted the whole GermaNet 5.0 into the OWL instance model, we employed a subset of GermaNet containing 54 synsets and 104 lexical units to develop the different OWL models for wordnets and the query implementations described in this chapter.

[4] From [22].

17.2.2 TermNet

TermNet [5] is a lexical resource that was developed in the HyTex project on automated text-to-hypertext conversion[5]. TermNet represents technical terms occurring in a German corpus on the domains "text-technology" and "hypertext research"[6]. The two basic entities of the TermNet model are *terms* and *termsets*. The entity type *term* corresponds to the entity type *lexical unit* in GermaNet: *terms* denote well-defined concepts in the above-mentioned domains – in many cases they are part of a taxonomy in which more specific terms are defined as subclasses of broader terms. *Termsets* collect terms that denote similar concepts in different taxonomies. Termsets, thus, correspond to the entity type *synset* in GermaNet with one significant difference: lexical units in the same synsets are regarded as near-synonyms – they are interchangeable in at least one context. Such an interchange is not possible for different technical terms in a termset because the semantics of technical terms is defined by their position in the taxonomy. Different authors or scientific schools may develop different and non-isomorphic taxonomies for the same domain. Therefore, it is not possible to exchange a technical term in a document for a technical term defined in a different taxonomy, even if the two terms denote quite similar concepts and therefore belong to the same termset. Termsets represent categorical correspondences between terms of competing taxonomies.[7] Modelling categorical correspondence, thus, opens up two modes for searching for term occurrences in a corpus: (1) the *narrow search mode* (based on the entity type *term*) for occurrences of a term in a specific taxonomy, and (2) the *broader search mode* (based on the entity type *termset*) for all occurrences of terms that denote a similar concept in the domain.

Apart from these specific characteristics of termsets, the TermNet data model builds on the entity and relation types of the Princeton WordNet model and the GermaNet model. Whereas lexical semantic relations (LSR) are defined between terms, termsets are related by conceptual relations (CR). The following LSR and CR are used in the TermNet data model:

- LSR: *isAbbreviationOf* (inverse: *isExpansionOf*)
- CR: *isHyponymOf* (inverse: *isHypernymOf*), *isMeronymOf*, *isHolonymOf*

In addition, we introduced a symmetrical relation *isDisjointWith* to describe the relation between terms with disjoint extensions (which is the case with terms which denote contrary subconcepts of one and the same superconcept, e.g., *externer Verweis* and *interner Verweis*).

[5] See `http://www.hytex.info` and [36].

[6] The current version of TermNet contains 423 technical terms in 206 termsets.

[7] A concrete example: in German hypertext research, the terms *externer Verweis* and *extratextuelle Verknüpfung* both denote hyperlinks leading to other "external" websites. The definition of *extratextuelle Verknüpfung* is opposed to two disjunctive terms (*intertextuelle Verknüpfung*, *intratextuelle Verknüpfung*) whereas *externer Verweis* is opposed to only one disjunctive term (*interner Verweis*). The concept of the *termset* is described in detail in [5].

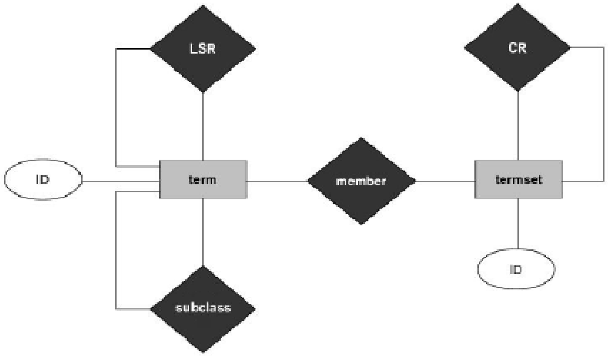


Fig. 17.2 E-R graph for TermNet.

In order to represent the relation between terms and termsets, we furthermore introduced a *membership*-relation which relates terms that denote the same category in different taxonomies with the respective termset (*isMemberOf*) and, inversely, the termset with one or several terms (*hasMember*).

As an extension to the standard WordNet model, TermNet represents subclass relations between terms of the same taxonomy. The data model is illustrated by the ER-diagram in Figure 17.2; further details are described in [6].

17.2.3 GermaTermNet

GermaTermNet connects the two above-described resources – the GermaNet subset and TermNet – following an approach inspired by [26]. The basic idea of this approach is to supplement GermaNet and TermNet by so-called plug-in relations[8]. The "plug-in" metaphor is motivated by the fact that these relations connect entities from the one with entities from the other resource without the necessity of modifying or merging the two resources. The strength of such a "plug-in" approach, thus, lies in the fact that resources which are independent of one another can be linked in order to process them in combination.

The current version of GermaTermNet distinguishes three types of plug-in relations which describe particular correspondences between general language concepts and their terminological counterparts:

1. *attachedToNearSynonym*: This relation describes correspondences between TermNet terms and GermaNet synsets; e.g. between the TermNet term *Link* and the GermaNet synset *Link*. Since we do not assume pure synonymy for a

[8] This approach has been developed in close cooperation with Claudia Kunze and Lothar Lemnitzer of the GermaNet group, see [22] and [25] for details.

corresponding term-synset pair, *attachedToNearSynonym* is the closest sense-relation between entities of the two resources.

2. *attachedToGeneralConcept*: This relation is used to relate TermNet terms to GermaNet synsets which stand in an *attachedToNearSynonym* relation with a superordinate term.[9] The *attachedToGeneralConcept* relations serve to reduce the path length between semantically similar concepts for applications in which semantic distance measures are calculated.
3. *attachedToHolonym*: This relation is used to relate a TermNet term t_1 to a synset s_1 in GermaNet when the following conditions are fulfilled: (1) t_1 is a member of a termset A, (2) this termset is a meronym of termset B, and (3) the members of B (i.e., a collection of terms $t_2, t_3, \ldots, t_n$) are connected with s_1 through the relation *attachedToNearSynonym*. An example: the term *arc*, a member of the termset *Kante*, is linked to the GermaNet synset *link* by an *attachedToHolonym* relation because *Kante* is a meronym of the termset *Link*, and one of its members, the term *link*, is attached to the GermaNet synset *link* via *attachedToNearSynonym*.

The current version of GermaTermNet represents 150 plug-in relation instances: 27 *attachedToNearSynonym*, 103 *attachedToGeneralConcept*, and 20 *attachedToHolonym* instances.

17.3 Modelling Wordnet-Like Resources in OWL

Modelling lexical-semantic and ontological resources in the Web Ontology Language OWL is a requirement for making them available on the semantic web i.e. enables developers to include them in their semantic web applications. Besides, the representation of knowledge sources in a formalism that has the status of a W3C recommendation contributes to making them truly *interoperable*. The benefit of the basic interoperability of two OWL resources is e.g. exploited in our conception of linking of GermaNet and TermNet to form GermaTermNet (cf. Section 17.3.3.3).

17.3.1 Basic Options

17.3.1.1 OWL Dialects

In the line of [17], [35] characterise an ontology as a "formal explicit specification of a shared conceptualisation for a domain of interest". It is controversial whether wordnets constitute proper ontologies; according to [14], a wordnet may count as a *light-weight ontology*, i.e. an "ontology primarily consisting of a schema (a concept taxonomy with attribute and relation definitions)" [14, p. 27, original in German]. [34] mentions PWN as an example of a "terminological ontology", i.e. an ontology that is not completely formalised.

[9] An example: the term *externer Verweis* is connected to the GermaNet synset *Link* by an *attachedToGeneralConcept* relation because it is a subclass of the term *Verweis* which itself is connected to *Link* by an *attachedToNearSynonym* relation.

The Web Ontology Language OWL is an XML application for the representation of ontologies in the semantic web, a standard created by the W3C Web Ontology working group. OWL comes in three sublanguages (dialects), OWL Full, OWL DL, and OWL Lite, which differ in their expressive power. When only constructs from the sublanguage OWL DL are used in an ontology, its semantics correspond to description logic [4]. Most available reasoning software is based on description logic, and reasoners such as Racer [18] support consistency checking and inferring new facts and relations, i.e. automatically extending OWL DL ontologies. As a consequence, most ontologies for the semantic web are encoded in OWL DL.

OWL Full is known to be undecidable. Probably no reasoning services will ever be implemented for ontologies that make full use of the expressive power of OWL Full. One important difference between OWL DL and OWL Full is that in an OWL DL ontology, the sets of the classes, individuals, and properties must be disjunct, e.g. an entity in the ontology may not be a class and an individual at the same time. This, however, is permitted in an OWL Full ontology.

17.3.1.2 Synsets and Lexical Units as Classes or Individuals?

A fundamental decision which has to be made when representing Wordnet-like resources in OWL concerns the question whether synsets and lexical units are to be described as classes or as individuals. From a linguistic point of view, synsets are *concepts (classes)* whose instances are discourse entities, and lexical units are types of linguistic expressions whose instances can be interpreted as the token occurrences of these expressions in documents. The decision to treat synsets and lexical units as OWL individuals conceives a wordnet primarily as a lexicon describing properties of individual lexical units while disregarding that nouns, in the traditional view of lexical semantics, denote concept classes, with concept classes being ordered hierarchically, superclasses including subclasses and subclasses in some cases (especially in terminologies) being pairwise disjoint.

Treating synsets and lexical items as *individuals* implies that the subconcept-superconcept relation and disjointness can only be described by non-standard individual-to-individual relations. An individual representing a subconcept does not automatically inherit the properties defined for the individual representing the superconcept.

Treating synsets and lexical items as *classes* allows for describing them as concepts and word senses, with entities of the real world or word occurrences in documents being described as their individuals. The "class model" of a Wordnet-like resource thus describes an ontology, whereas the "instance model" – much like a lexicon – describes synsets and lexical units as instances of linguistic categories such as *NounSynset* or *NounWordSense*.

17.3.2 Related Work

What OWL models of wordnets have been designed previously, and how are these related to the ones presented in this chapter? In 2006, the *Semantic Web Best Practices and Deployment Working Group* issued a working draft for a standard conversion of the PWN in an RDF/OWL (in fact OWL DL) representation that can be used by semantic web applications [2, 3]. Its top-level classes are *Synset*, *WordSense*, and *Word*, where *Synset* represents synsets and has the subclasses *NounSynset*, *VerbSynset*, *AdjectiveSynset*, and *AdverbSynset*. The individual synsets are individuals of these classes. CRs are defined over the synset class, e.g. the OWL object property *hyponymOf* has *Synset* as its domain and its range. Likewise, the class *WordSense* has the subclasses *NounWordSense*, *AdjectiveWordSense*, *VerbWordSense*, and *AdverbWordSense*. Lexical units are modelled as the individuals of these classes. LSRs like *antonymOf* are OWL object properties with *WordSense* as their domain and their range. The lexicalisation relation is an OWL object property called *synsetContainsWordSense* and has the *Synset* as its domain and *WordSense* as its range. The third top-level class, *Word*, represents a purely formal unit, i.e. not associated with a meaning. It is connected with the class *WordSense* via the OWL object property *word*, which has *WordSense* as its domain and *Word* as its range. Properties with an attribute-like character including *lexicalForm* or *synsetId* are rendered as OWL datatype properties with XML Schema datatypes such as xsd:string as their range.

There are two earlier approaches to representing wordnets in OWL, the "Neuchâtel" approach [10, 9] and the "Amsterdam" approach [1]. Neuchâtel has been explicitly acknowledged by [3], and Amsterdam and W3C partly have identical authors. The W3C approach is fairly close to Neuchâtel; one difference is that the *StemObject* class in Neuchâtel which corresponds to the *Word* class in the W3C approach is only introduced in a wordnet-external ontology for a document retrieval application. In the Amsterdam model, a different top-level structure of the ontology is specified, mostly in that lexical units are neither classes nor individuals but literal values of the OWL datatype property *wordForm* with the domain *Synset*. As a consequence, the encoding of LSRs is achieved by a different mechanism.

We took the W3C approach to convert PWN into OWL as a model for a first GermaNet representation in OWL [22], which is briefly described below in Section 17.3.3.1 as the "OWL DL Instance Model" for GN. The W3C approach was also adopted by [13] for a conversion of EuroWordNet [38] into OWL.

In [3], an alternative model is briefly discussed. For some purposes it might be useful to treat the hyponymy relation as a class hierarchy by declaring *Synset* a subclass of rdfs:class (thus make each individual synset a class and an individual at the same time) and *hyponymOf* a subproperty of rdfs:subClassOf as suggested in [40].

17.3.3 OWL Models for GermaNet, TermNet, and GermaTermNet

In this section, we describe the three alternative models that we developed for encoding wordnets in OWL, namely the *OWL DL Instance Model*, the *OWL DL class model*, and an *OWL Full Model*. They are based on different outcomes for the following modelling decisions:

1. the representation of synsets and lexical units as OWL classes, individuals, or both
2. the encoding of relation instances in OWL as property restrictions or assignments of individuals as property values (as a consequence of 1.)
3. the conformance with the dialect OWL DL or OWL Full (as a consequence of 1. and 2.)
4. the way of linking word occurrences in XML documents to lexical units (strictly speaking not part of the ontology)

We describe the implementation of the three models for GermaNet in Section 17.3.3.1 and for TermNet in 17.3.3.2. OWL Models for GermaTermNet are described in Section 17.3.3.3.

17.3.3.1 GermaNet

For GermaNet, the top-level hierarchy of classes is defined using the `<owl:class>` and `<rdfs:subClassOf>` statements and is the same in all three models. It corresponds to the indented list representation shown in Figure 17.3, and is similar to the class hierarchy of the W3C model.

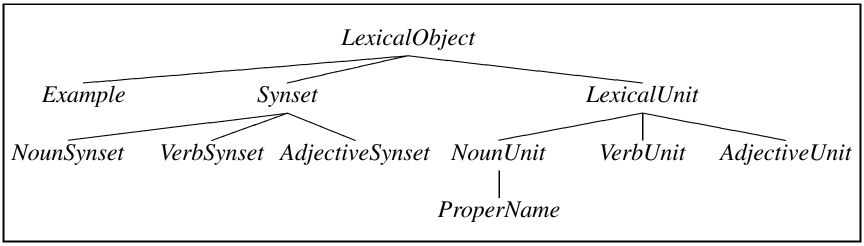


Fig. 17.3 Class hierarchy for OWL models of GN.

The GN relations shown as diamonds in the data model in Figure 17.1 are modelled as OWL object properties. To capture the commonalities of CRs and LSRs, we introduced the two superproperties *conceptualRelation* (domain and range: *Synset*) and *lexicalSemanticRelation* (domain and range: *LexicalUnit*).

Listing 17.1 shows the OWL code introducing the lexicalisation relation *hasMember* as an OWL inverse functional property (a special type of object property).

Table 17.1 Characteristics of the ObjectProperties for GermaNet.

Property	Domain	Range	Characteristics	Inverse Property	Local Restrictions
Conceptual relations (CR)					
conceptualRelation	Synset	Synset			
isHypernymOf	Synset	Synset	transitive	isHyponymOf	*pos-related*
isHolonymOf	NounSynset	NounSynset			
isMeronymOf	NounSynset	NounSynset			
isAssociatedWith	Synset	Synset			
entails	VerbSynset	VerbSynset			
causes	VerbSynset ⊔ AdjectiveSynset	VerbSynset			
Lexical-semantic relations (LSR)					
lexicalSemanticRelation	LexicalUnit	LexicalUnit			
hasAntonym	LexicalUnit	LexicalUnit	symmetric	hasAntonym	*pos-related*
hasPertonym	LexicalUnit	LexicalUnit			
isParticipleOf	VerbUnit	AdjectiveUnit			
lexicalisation relations					
hasMember	Synset	LexicalUnit	inverse-functional	memberOf	*pos-related*

Listing 17.2 shows the code for introducing *isHypernymOf* as an OWL transitive property (also a special type of object property). All other object properties are declared similarly in OWL. Most attributes in the data model of GermaNet (the ellipses in the E-R-graph in Figure 17.1) are represented as OWL datatype properties [cf. 25].

```
<owl:InverseFunctionalProperty rdf:about="#hasMember">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Synset"/>
  <rdfs:range rdf:resource="#LexicalUnit"/>
  <owl:inverseOf rdf:resource="#memberOf"/>
</owl:InverseFunctionalProperty>
```

Listing 17.1 OWL code introducing the lexicalisation relation *hasMember* for GermaNet.

```
<owl:TransitiveProperty rdf:about="#isHypernymOf">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:subPropertyOf rdf:resource="#conceptualRelation"/>
  <rdfs:domain rdf:resource="#Synset"/>
  <rdfs:range rdf:resource="#Synset"/>
  <owl:inverseOf>
    <owl:TransitiveProperty rdf:about="#isHyponymOf"/>
  </owl:inverseOf>
</owl:TransitiveProperty>
```

Listing 17.2 OWL code introducing the hypernymy relation *isHypernymOf* for GermaNet.

OWL DL Instance Model. Our first modelling variant closely follows the principles of the W3C approach to converting PWN to OWL: the individual synsets and lexical units are modelled as OWL individuals and the encoding of CRs, LSRs and the lexicalisation relation as an assignment of individuals as property values. In the Instance Model for GN, there is an additional class *LUOccurrence*, the individuals of which have (pseudo-) URIs as IDs to simulate the token occurrences of lexemes in documents. *LUOccurrence* is related to its LU individual through the object property *isOccurrenceOf* (domain: *LUOccurrence*, range: *LexicalUnit*).[10]

In Listing 17.3, OWL code examples of assignments for CRs (*isHypernymOf*) and the lexicalisation relation (*hasMember*) are shown for the synset *vVeraenderung.119*. Examples of assignments for LSRs and datatype properties are shown for the lexical unit *vVeraenderung.199.wandeln*. Finally, there is the occurrence individual *URI_LUnitInstance_vVeraenderung.119.wandeln_1* with an assignment of its LU for the property *isOccurrenceOf*.

```
<VerbSynset rdf:ID="vVeraenderung.119">
   <hasMember rdf:resource="#vVeraenderung.119.wandeln"/>
   <hasMember rdf:resource="#vVeraenderung.119.ändern"/>
   <hasMember rdf:resource="#vVeraenderung.119.mutieren"/>
   <hasMember rdf:resource="#vVeraenderung.119.verändern"/>
   <isHypernymOf rdf:resource="#vVeraenderung.421"/>
   <isHypernymOf rdf:resource="#vVeraenderung.517"/>
</VerbSynset>

<VerbUnit rdf:ID="vVeraenderung.119.wandeln">
   <hasOrthographicForm
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      wandeln</hasOrthographicForm>
   ...
</VerbUnit>

<LUOccurrence rdf:ID="URI_LUnitInstance_vVeraenderung.119.wandeln_1">
    <isOccurrenceOf rdf:resource="#vVeraenderung.119.wandeln"/>
</LUOccurrence>
```

Listing 17.3 OWL code for a synset, lexical unit, and occurrence individual in the instance model for GN. Note the encoding of the *hasMember* and *isHypernymOf* relation instances.

OWL DL Class Model. For the reasons discussed in Section 17.3.1.2, we alternatively devised an *OWL DL Class Model* for GermaNet. In the class model, each individual synset and lexical unit is a subclass of one of the top-level classes *NounSynset, VerbSynset, NounUnit* etc. Since within OWL DL property value assignments can only be defined between individuals, the GermaNet relation instances are modelled as property restrictions over the single synset or unit classes using *owl:Restriction* in combination with the *owl:allValuesFrom* construct. This holds for CRs, LSRs, as well as the lexicalisation relation, cf. the example of hyponym declarations for the synset *vVeraenderung.119* in Listings 17.4 and 17.5.

[10] Although we think that *LUOccurrence* and *TermOccurrence* (see Section 17.3.3.2) and their properties are strictly speaking not part of wordnets or the lexicon, we have included them in our ontologies for the purpose of linking our annotated text corpora to them.

```
<owl:Class rdf:ID="vVeraenderung.119">
  <rdfs:subClassOf rdf:resource="#VerbSynset"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:TransitiveProperty rdf:about="#isHypernymOf"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#vVeraenderung.421"/>
            <owl:Class rdf:about="#vVeraenderung.517"/>
          </owl:unionOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Listing 17.4 OWL code for the *isHypernymOf* relations in the class model for GN using restrictions

In the class model, the place of individuals of the LU classes can be straightforwardly taken by their textual occurrences. Unlike in the case of the instance model, the GN top-level ontology does not need to be extended to accommodate them.

```
<vVeraenderung.119.verändern
  rdf:ID="URI_LUnit_Instance_vVeraenderung.119.verändern_1"/>
```

Listing 17.5 Occurrence individual in the class model

OWL Full Model. In [24], we put forward some criticism of the instance model for wordnets in OWL (summarised in Section 17.3.1.2 above). In the class model, on the other hand, the relation assignments via property restrictions seem counterintuitive; moreover, although occurrence instances of LUs are conveniently included as individuals of their LU classes, it does not seem adequate that they inherit all property specifications by virtue of their instancehood (CRs and LSRs are *lexical* relations and are not supposed to hold between word occurrences in a document).

Thus, as a third option we converted the OWL instance model of GermaNet into an *OWL Full Metaclass Model* by adding the following line to the definitions of the classes *Synset* and *LexicalUnit*.

```
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

This makes *Synset* and *LexicalUnit* metaclasses, i.e. their individuals, the single synsets and lexical units, are also classes. Consequently, an LU class can be populated with occurrence individuals that do *not* inherit lexical properties, *and* CRs, LSRs and the lexicalisation relation can be added as simple property value assignments (the code for which looks just like that for the instance model in Listing 17.3).

Unfortunately, the ontology now lies outside the scope of OWL DL, i.e. is in OWL Full [cf. 33], and DL-based reasoners cannot be applied to it.

A point of criticism one could formulate against the OWL Full model is that when single synsets (and LUs) are classes besides being individuals, then one would also expect these classes to be part of another (second) class hierarchy. In fact, in [29], [28], and in the third example in [31], a major motivation for introducing metaclasses is the fact that certain individuals of an ontology should be subclasses in a second class hierarchy, e.g. an *ArchetypalOrangUtan* individual should be an individual of an *Archetype* class and at the same time subclass of an *OrangUtan* class. On the other hand, in the meta-modelling examples in [31] (first two examples) and in [30] (dealing with the PWN), there is no such second hierarchy; the motivation given is simply that some entities are both a class and an individual. In [3], where a metaclass approach to modelling the PWN in OWL is proposed as a variant, the motivation is to interpret the hyponym relation as a (second) class hierarchy.

17.3.3.2 TermNet

For each one of the three modelling options of TermNet a separate top-level hierarchy of classes is defined using the *¡owl:class¿* and *¡rdfs:subclassOf¿* statements. The hierarchy for the OWL DL instance model is shown in Figure 17.4, the hierarchy for the OWL DL class model and the OWL Full model in Figure 17.5.

As in our GermaNet models, the CRs and LSRs in TN (diamonds in the data model in Figure 17.2) are defined as OWL object properties. Listing 17.6 shows the OWL code defining the *isMemberOf* relation between the term concept *TermConcept_Relation* and the termset *TermSet_Link* in the OWL DL class model. All other object properties in our TN models (as listed in Table 17.2) are declared similarly.

```
 <owl:Class rdf:ID="TermConcept_Relation">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isMemberOf"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:ID="TermSet_Link"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Listing 17.6 OWL code describing the relation between the term concept *TermConcept_Relation* and the termset *TermSet_Link* in the OWL DL class model of TermNet

OWL DL Instance Model. Following the principles of the W3C approach to converting PWN to OWL, the term concepts and termsets of the TN instance model are realised as OWL individuals of the top-level classes *TermConcept* and *TermSet*, and all of the relations mentioned in Table 17.2 are described as object properties

Table 17.2 Characteristics of the ObjectProperties for TermNet; *CM* = OWL DL class model, *IM* = OWL DL instance model, *MM* = OWL Full metaclass model.

Property	Domain	Range	Characteristics	Inverse Property	CM	IM	MM
Membership relations							
hasMember	TermSet	TermConcept		isMemberOf	X	X	X
Type/token relations							
isTypeOf	TermForm	TermOccurrence		isTokenOf	X	X	X
Lexicalisation relations							
lexicalizes	TermForm	TermConcept		isLexicalizedAs		X	
Occurrence relations							
occursIn	TermConcept	TermOccurrence		isOccurrenceOf		X	
Lexical-semantic relations (LSR)							
isAbbreviationOf	TermConcept	TermConcept		isExpansionOf	X	X	X
Disjointness							
isDisjointWith	TermConcept	TermConcept	symmetric	isDisjointWith		X	
Conceptual hierarchy between term concepts							
isNarrowerTermOf	TermConcept	TermConcept	transitive	isBroaderTermOf		X	
Conceptual relations (CR) between termsets							
isHyponymOf	TermSet	TermSet	transitive	isHypernymOf	X	X	X
isMeronymOf	TermSet	TermSet			X	X	X
isHolonymOf	TermSet	TermSet			X	X	X

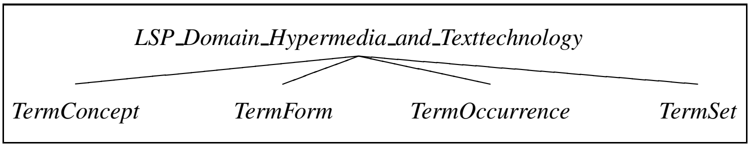


Fig. 17.4 Class hierarchy for the OWL DL Instance Model of TermNet

between instances. Occurrences of technical terms in documents from our corpus are given as URI references and are described as individuals of the top-level class *TermOccurrence*, and the mutually inverse object properties *occursIn* – *isOccurrenceOf* are used to link individuals of term occurrences with individuals of the *TermConcept* class. In order to describe the lexicalisation relation between term forms and term concepts, we additionally introduced the relations *isLexicalizedAs* – *lexicalizes* which relate instances of *TermConcept* with instances of *TermForm* (cf. Figure 17.4).

In order to represent the further class hierarchy below the top-level class *TermConcept*, inclusion between general and more specific terminological concepts is described through the mutually inverse relations *isBroaderTerm* – *isNarrowerTerm*. Since modelling individual terminological concepts as individuals does not allow for using the OWL construct *¡owl:disjointwith¿*, disjointness between instances of *TermConcept* was described through the object property *isDisjointWith*.

OWL DL Class Model. The OWL DL Class Model of TermNet was created due to the same reasons as its GermaNet counterpart (see Section 17.3.3.1). In the class model, each individual term concept and termset is a subclass of one of the top-level classes *TermConcept* and *TermSet*. Given that two concepts A and B belong to one and the same terminological system with the extension of B representing

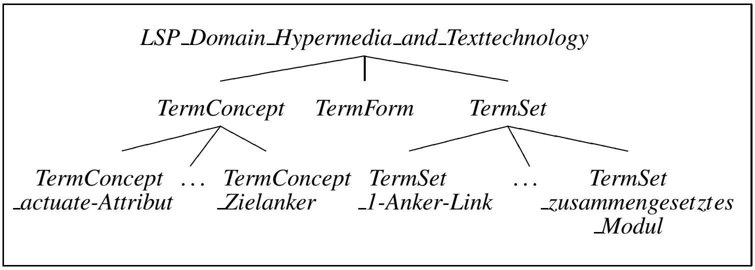

LSP_Domain_Hypermedia_and_Texttechnology
TermConcept
TermForm
TermSet
TermConcept _actuate-Attribut
...
TermConcept _Zielanker
TermSet _1-Anker-Link
...
TermSet _zusammengesetztes _Modul

Fig. 17.5 Class hierarchy for the OWL DL Class Model and OWL Full Model of TermNet.

a subclass of the extension of A, the classes representing A and B are connected through a *superclass-subclass* relation. Given that two subclasses of *TermConcept* can be regarded as being disjoint, we labelled A and B with the OWL construct *<owl:disjointwith>*.

URI references to occurrences of individual technical terms in our corpus are described as OWL individuals of the corresponding specific *TermConcept* classes.

The top-level class *TermForm* bundles the entirety of orthographic forms with which the terminological concepts can be instantiated when used in discourse (in our case in one of our corpus documents). *TermForm* has no subclasses; instead, every single form is described as an individual of this class. Each individual is linked to an occurrence of the respective form in the corpus using a *type/token* relation.

Since in OWL DL classes cannot be assigned as property values for classes, CRs, LSRs, and membership relations could be established only indirectly using *owl:Restriction* in combination with the *<owl:allValuesFrom>* construct (cf. Listing 17.6). Even though restrictions are defined for classes, they do not describe a pair of classes as an instance of a relation. Instead, they determine that the individuals of the respective classes may be connected by OWL object property value assignments. Modelling relation instances between classes would imply a transition to OWL Full. The restrictions in the class model are thus rather workarounds, mainly serving consistency purposes.

OWL Full Model. By adding the line of code mentioned in Section 17.3.3.1 into the definitions of the classes *TermConcept* and *TermSet*, the OWL DL instance model of TermNet was converted into the OWL Full Metaclass Model. Through this modification, *TermConcept* and *TermSet* become metaclasses which means that their instances (the individual term concepts and termsets) are both instances and classes at the same time. As a consequence and in contrast to the OWL DL class model, relation instances between subclasses of *TermConcept* and *TermSet* could be explicitly described as property value assignments between classes. Nonetheless and as mentioned in Section 17.3.1.1, OWL Full ontologies cannot be queried and processed by reasoning services which are based on description logics, but e.g. by using a logic programming language such as Prolog.

17.3.3.3 GermaTermNet

Theoretically, based on the three OWL models of GermaNet and TermNet described above, 3 x 3 = 9 differing OWL models for GermaTermNet may be derived. In the following, we describe the implementation of the plug-in relations of GermaTermNet for:

- the combinations of the respective OWL DL instance models (henceforth referred to as the *GermaTermNet instance-instance model*);
- the combinations of the respective OWL DL class models (henceforth referred to as the *GermaTermNet class-class model*);
- the combination of the OWL DL instance model of GermaNet with the OWL DL class model of TermNet (henceforth referred to as the *GermaTermNet "hybrid model"*);
- the combinations of the respective OWL Full models.

In the instance-instance model, the plug-in relations described in Section 17.2.3 as well as their inverses have been established directly between individuals of TermNet and GermaNet (cf. Listing 17.7, below). In the class-class model these relations and their inverses have been indirectly realised through establishing *allValuesFrom*-restrictions which apply to the TermNet subclasses of *TermConcept* and the GermaNet subclasses of *Synset*.

```
<rdf:Description
  rdf:about="http://www.owl-ontologies.com/TermNet.owl#TermConcept_Knoten">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue
       rdf:resource=
       "http://www.owl-ontologies.com/GermaNet-instances.owl#nArtefakt.5816"/>
      <owl:onProperty rdf:resource="#attachedToNearSynonym"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</rdf:Description>
```

```
<rdf:Description
  rdf:about="http://www.owl-ontologies.com/TermNet.owl#TermConcept_Knoten">
  <attachedToNearSynonym
    rdf:resource=
    "http://www.owl-ontologies.com/GermaNet-instances.owl#nArtefakt.5816"/>
</rdf:Description>
```

Listing 17.7 Connecting resources: Instance of the *attachedToNearSynonym*-relation in the "hybrid model" (above) and in the "instance-instance model" (below); in both examples the TermNet term concept *Knoten* is linked to the GermaNet synset individual *nArtefakt.5816* which comprises the lexical unit *Knoten*.

The plug-in relations of the "hybrid model" [22] have been realised through establishing *hasValue*-restrictions which link subclasses of *TermConcept* with individuals of *Synset* (cf. Listing 17.7, above). Since OWL DL does not allow for specifying inverse relations between classes and instances, the restrictions are defined only one-way from classes to instances, and not vice-versa. The plug-in relations of the OWL Full model have been modelled in the same way as in the instance-instance model, namely by establishing direct relations between *TermConcept* and *Synset* individuals.

17.4 Processing WordNet-Like Resources in OWL

In this section, we will discuss the advantages and drawbacks of the three OWL models for wordnets described above when it comes to using them as lexical-semantic resources in the context of hypermedia applications (Section 17.4.1) and in the context of semantic relatedness calculations (Section 17.4.2). Section 17.4.1 describes, with the example of the different models of the TermNet resource, how the decision for a certain OWL sublanguage as well as for one of the three models leads to different conditions for querying the resource. We discuss which of the three OWL models can be processed more efficiently and compare three sample queries with respect to the complexity of the query terms needed to implement them on the resource and with respect to the elapsed time between sending the query to the server and receiving the query results. In Section 17.4.2, we firstly investigate ways to process OWL Full models of wordnets and secondly the potential of a connection of a general language with a specialised wordnet within OWL by the example of wordnet-based semantic relatedness calculation. The queries that we use do involve reasoning, but not logically complex reasoning with the purpose of drawing inferences or deriving theorems over individuals and concepts in our knowledge base. Instead, they represent certain text-technological tasks motivated by our project contexts of text parsing and automatic hypertextualisation.

17.4.1 Processing the OWL Models of TermNet

Application scenario: For the comparison of the three OWL models of the TermNet resource, we defined three sample queries. The queries have been defined with respect to the application scenario of the HyTex project, which describes a hypermedia application that supports the selective reading of and the intellectual information retrieval from documents from a corpus of scientific texts [36]. The application is designed especially with respect to users who are not experts in the scientific domain – instead, it is meant to support especially those users who only have basic knowledge of the concepts which are relevant to the domain (user groups such as e.g. students, journalists, or scholars from other scientific disciplines) and provide them with all information that they need for a proper understanding when selectively browsing through the documents.

This support may be given on the one hand by providing the readers with hyperlinks to definitions of the respective technical terms given in other hypertext nodes or in a glossary which are relevant for the understanding of the occurrence of the respective term in the hypertext nodes currently being read. In addition, the comprehension of a terminological concept often requires not only the availability of a definition of the respective technical term but also at least a rudimentary knowledge of its lexical-semantic or conceptual neighbourhood. Readers with a non-expert status therefore also need support for the exploration of the conceptual structure of the respective domain and its representation in the terminologies used in the scientific community.

In our application scenario, TermNet serves as a resource to provide users with the possibility to explore the terminological context of a given technical term (e.g. after having found this term in one of the hypertext nodes) as well as to explore the conceptual structure of the domain irrespective of a specific terminological representation of it (e.g. as given in the terminological system of a certain author or scientific paradigm). Our three sample queries represent needs which can be considered typical for non-expert users when selectively browsing through scientific documents:

User scenario 1: Making use of the TermNet component of the application is driven by the motivation to view all text passages in the corpus that have to do with one particular terminological concept.

Sample scenario: A lexicographer who has to compose the entry "Hyperlink" for a specialised dictionary on information technology should not only be provided with all text passages in which the TermForm(s) *hyperlink* and *link* (as lexicalised forms of the TermConcept *Link*) occur, but in addition with all text passages in which other TermForms occur that do not lexicalise the TermConcept Link, but one of its subconcepts (e.g., *bidirektionaler Link*, *1:1-Link*, *Inhalts-Link*).

→ **Query 1:** "Search for all term occurrences in the corpus which denote a given term concept or a more specific one." (e.g., "Search for all URIs that have to do with links.").

User scenario 2: Making use of the TermNet component of the application is driven by the motivation to view all text passages in the corpus that have to do with a particular scientific category.

Sample scenario: A student who has to write a term paper on the hyperlink concept in hypermedia research should not only be provided with all text passages in which the TermForm(s) *hyperlink* and *link* occur, but in addition with all text passages in which different TermForms occur that relate to the same category, but do not lexicalise the same TermConcept (e.g., also text passages which contain occurrences of the TermForms *Verknüpfung* and *Kante* as well as of TermForms which lexicalise subconcepts of the TermConcepts *Verknüpfung* and *Kante*).

→ **Query 2:** "Search for all URIs that relate to TermConcepts which belong to a given termset." (e.g., "Search for all term occurrences in the corpus that have to do with links *or similar concepts*.").

User scenario 3: While browsing the hypertext nodes of the application, a non-expert user is suddenly faced with the occurrence of a term which seems to be fundamental for the comprehension of the text; s/he decides that s/he should learn more about the category to which this term refers, in order to improve his/her knowledge prerequisites for a proper understanding.

Sample scenario: A student who is planning to attend a course "Introduction to Hypermedia Research" in the upcoming semester wants to get a first overview of the field of hypermedia research. He browses some hypertext nodes of the application and is faced with an occurrence of the TermForm *link*. He already has an intuitive concept of *link* from his everyday internet use but has the intuition that in the scientific literature, the concept area associated with the form *link* might be more differentiated. He thus wants to view other hypertext nodes that have to do with *links*. Since he is motivated to gather information about a certain conceptual area, or category, he should not only be led to hypertext nodes that contain further occurrences of the TermForm link, but also nodes that contain occurrences of different TermForms which are not associated with the same TermConcept Link but with TermConcepts which are members of the same termset to which the TermConcept Link belongs.

→ **Query 3:** "Start from a given term occurrence in the corpus and search for occurrences of any technical terms which denote the same or a similar concept." (e.g., "Search for all URIs that have to do with the same or a similar concept as the term occurrence link in node #127.").

Implementation: For the OWL DL models (TN-IM, TN-CM), we implemented the queries using the query language *nRQL* in combination with RacerPro[11] for processing. Due to the different models, the complexity of the queries vary between TN-CM and TN-IM (as can be seen from the examples in Listing 17.8 which give the nRQL expressions for query 1). This is, on the one hand, due to the lack of class hierarchies between term concepts in TN-IM: since in TN-IM no superclass-subclass relations with implicit class inclusion are available, the inclusion of specific concepts under general concepts can only be considered by explicitly querying the object property *isNarrowerTermOf*. On the other hand, URIs in TN-IM are modelled as individuals of a top-level class *TermOccurrence* and not as individuals of term concepts. This is due to the fact that term concepts are already individuals themselves and, thus, in OWL-DL can not have further individuals.

Reasoners such as *RacerPro* are based on description logics and cannot be applied to OWL Full ontologies. However, OWL Full ontologies can be parsed and queried using the Thea OWL Library for *SWI Prolog* [37] which in turn utilises the SWI-Prolog Semantic Web Library [39]. We thus used Prolog for querying the OWL Full Metaclass model (TN-MM).[12] The Prolog predicates for query 1 are given in Listing 17.8. It searches for all term occurrences of a certain concept or one of its subconcepts.

[11] The acronym *RacerPro* stands for *Renamed ABox and Concept Expression Reasoner Professional* and has been developed by Racer Systems GmbH & Co. KG (`http://www.racer-systems.com`). In Racer, knowledge bases can be queried using the *new Racer Query Language* (nRQL) which is described in [19].

[12] Another reason why we are particularly interested in using Prolog to access our resources is that we potentially want to interface the relational discourse parser which was developed in one of the projects involved in this research and which has been coded in Prolog (see [23] in this volume).

```
(retrieve
  (?x)
  (?x |TermConcept_Link|)
)
```

```
(retrieve
  (?x)
  (or
    (and
      (?x |TermOccurrence|)
      (|TermConcept_Link| ?x |occursIn|)
    )
    (and
      (?y |TermConcept|)
      (?y |TermConcept_Link| |isNarrowerTermOf|)
      (?x |TermOccurrence|)
      (?y ?x |occursIn|)
    )
  )
)
```

```
termOccurrenceForConceptOrSubConcept(Concept, URI):-
  termOccurrenceForConcept(Concept, URI).

termOccurrenceForConceptOrSubConcept(Concept, URI):-
  transitive_subclassOf(SubConcept, Concept),
  termOccurrenceForConcept(SubConcept, URI).

findOccurrencesForConceptOrSubconcept(Concept, L):-
  findall(URI, termOccurrenceForConceptOrSubConcept(Concept, URI), L).
```

Listing 17.8 Query 1 as expressed for the different TermNet models: in nRQL for TN-CM (above), in nRQL for TN-IM (middle), and the Prolog predicates for TN-MM (below).

```
(retrieve
  (?x)
  (and
    (?x |http://www.owl-ontologies.com/TermNet.owl#TermConcept|)
    (?x |http://www.owl-ontologies.com/TermNet.owl#TS_Link|
      |http://www.owl-ontologies.com/TermNet.owl#isMemberOf|)
  )
)
```

```
(retrieve
  (?x)
  (and
    (?x |http://www.owl-ontologies.com/TermNet.owl#TermOccurrence|)
    (?y |http://www.owl-ontologies.com/TermNet.owl#TermConcept|)
    (?y |http://www.owl-ontologies.com/TermNet.owl#TermSet_Link|
      |http://www.owl-ontologies.com/TermNet.owl#isMemberOf|)
    (?y ?x |http://www.owl-ontologies.com/TermNet.owl#occursIn|)
  )
)
```

```
termOccurrenceForSet(Set, URI):-
  termConceptForSet(Set, Concept),
  termOccurrenceForConcept(Concept, URI).

findOccurrencesForSet(Set, L):-
  findall(URI, termOccurrenceForSet(Set, URI), L).
```

Listing 17.9 Query 2 as expressed for the different TermNet models: in nRQL syntax for TN-CM (above), in nRQL for TN-IM (middle), and the Prolog predicates for TN-MM (below).

In order to evaluate the efficiency with which the three models can be processed, we compared the queries 1 and 2 according to (a) the complexity of the query expression and (b) the average time elapsed between sending the query and receiving its answer. We defined "query complexity" as the number of conjuncts, i.e. the number of atoms which are connected through AND-relations, in the query body. We measured the average elapsed time by averaging over the results from executing each query one hundred times per model on one and the same machine. This experiment was conducted on a 32-bit Windows XP machine (AMD Turion 64 X2 2.2 GHz with 2.4 GB RAM). As a reasoner, we used *RacerPro 1.9.0* running at *localhost*. The Prolog queries had been processed with *SWI Prolog*[13] in combination with the semweb.pl and Thea libraries for parsing and querying OWL documents.

Table 17.3 shows the results of the evaluation of TN-CM, TN-IM, and TN-MM with the two queries. The results show that the nRQL expressions needed to query TN-IM are significantly more complex than the expressions needed for TN-CM. Concerning the time needed to process the queries, the findings are converse: although the query expressions for TN-CM are less complex than the expressions for TN-IM, the elapsed time between sending the query and receiving the answer is 46.7% higher at average for TN-CM than for TN-IM. Since the TN-CM resource consists of 72% axioms and 28% facts whereas the TN-IM resource consists of 1% axioms and 99% facts (cf. Table 17.4), this may be due to the fact that axioms have to go through several preprocessing steps before being able to be handled by a reasoner (cf. [32]) so that knowledge bases with large numbers of axioms may degrade performance while being queried [20].

The complexity of the queries to TM-MM in Prolog+Thea measured by the number of conjunct predicates called is higher because Thea does not store the OWL facts as simple triples of atoms but uses a list representation for the types of an individual. Besides, a predicate for namespace expansion needs to be called several times, and the transitivity of the subclass needed to be implemented and called specifically. The number in brackets in Table 17.3 shows the complexity when the member, namespace expansion, and transitivity calls are subtracted and is supposedly better comparable with the nRQL/SWRL complexities given for the TM-IM and TM-CM. Even though they are more complex than the nRQL queries for TN-CM, the Prolog queries for TN-MM are processed faster than the queries for TN-CM. While *SWI Prolog* needed 0.95 s per conjunction for processing the TN-MM queries, Racer Pro needed 3.19 s (query 1) and 1.52 s (query 2) per conjunction for processing the TN-CM queries. Nevertheless, the best results were obtained for the TN-IM queries (0.35 s and 0.5375 s per conjunction) which might be due to the fact that when generating TN-MM from TN-IM, class inclusion between term concept classes had been re-introduced so that TN-MM contains 56 times more axioms than TN-IM (cf. Table 17.4).

Besides the processing speed, TN-MM has the advantage that Prolog is a much more powerful querying device than nRQL. This can be illustrated by means of query 3: While it is no problem to express this query in Prolog, the nRQL syntax

[13] http://www.swi-prolog.org/

Table 17.3 Differences in processing the three TN models with queries 1 and 2 according to query complexity and elapsed time.

Query	Model	Complexity	Elapsed time		
			average (seconds)	*variance*	*standard deviation*
1	CM	1	3.19	0.0428	0.20688
	IM	6	2.10	0.0027	0.05196
	MM	7 (3)	2.85	0.0016	0.04000
2	CM	2	3.04	0.0031	0.05568
	IM	4	2.15	0.0014	0.03742
	MM	11 (3)	2.85	0.0014	0.03742

Table 17.4 Number of axioms and facts in the different TN models (for TN-CM and TN-IM before and after applying the SWRL rules given in Listing 17.10).

	TN-CM	TN-CM (expanded)	TN-IM	TN-IM (expanded)	TN-MM
Axioms	5,511	5,516	82	87	4,595
Facts	2,145	2,754	6,153	7,814	5,459

is restricted to the definition of simple inference patterns on the basis of class inclusion, superclass-subclass relations, or object properties. In order to define an nRQL expression to process this query for the two OWL DL models with *RacerPro*, we therefore first had to expand our resources by a new symmetrical object property which connects URI individuals which are occurrences of term concepts that are members of one and the same termset. For this purpose, we used the *Semantic Web Rule Language*[14] (*SWRL*) in order to automatically infer instances of a new object property named *instantiatesSimilarConceptAs* into TN-CM and TN-IM. *SWRL* allows for the augmentation of ontologies by inferring new facts (information about individuals) through the definition of implications between antecedents and consequents.

For the definition of our SWRL rule, we used the *Rules Tab* module[15] of the Protégé editor. For the application of SWRL rules to OWL resources, *Rules Tab* uses the *Jess* reasoner[16]. The application of the SWRL rule then automatically added new facts (information about individuals) to the resources, namely 609 instances of the property to TN-CM and 1,661 instances to TN-IM (cf. Table 17.4). The difference in the number of properties added to TN-CM and to TN-IM is due to the different modelling of term concepts and termsets as described in Section 17.3.3. In order to test querying relatedness between classes in TN-CM anyway, we added dummy instances to some (but not all) of our TermSet classes. Thus, the number of object properties added by applying the SWRL rule is less for TN-CM than for TN-IM. The SWRL rule for TN-CM and TN-IM is given in Listing 17.10.

[14] http://www.w3.org/Submission/SWRL/

[15] http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

[16] http://herzberg.ca.sandia.gov/jess/

```
TermConcept(?y) ∧ TermSet(?z) ∧ isMemberOf(?y, ?z) ∧ TermConcept(?a) ∧
  hasMember(?z, ?a) → instantiatesSimilarConceptAs(?y, ?a)
```

```
TermOccurrence(?x) ∧ TermConcept(?y) ∧ occursIn(?y, ?x) ∧ TermSet(?z) ∧
  isMemberOf(?y, ?z) ∧ TermConcept(?a) ∧ isMemberOf(?a, ?z) ∧
  TermOccurrence(?b) ∧ occursIn(?a, ?b) → instantiatesSimilarConceptAs(?x, ?b)
```

Listing 17.10 SWRL rules used for query 3 as specified for TN-CM (above) and TN-IM (below).

17.4.2 Processing the OWL Full Version of GermaTermNet

Previously, we implemented a set of basic queries to the GermaNet OWL Full model in Prolog on top of Thea, reported in [24]. That implementation mostly allowed for querying hyponym sets and could be applied as well to the OWL DL instance model of GN. For the present study we aimed at a more complex querying scenario for wordnets. It should

- include different kinds of elementary queries and ways of combining them
- utilise genuine OWL Full features, i.e. the class+individual status of lexical units in GN and of term concepts in TN
- utilise GermaTermNet, i.e. analyse the plug-in relations connecting the general language GN with the domain-specific TN
- start from or involve the occurrence individuals which emulate word token occurrences in documents
- constitute a component in a typical text-technological application

We chose the area of wordnet-based calculation of semantic distance/relatedness for querying the GermaTermNet OWL Full model. Determining the semantic relatedness of words/concepts is a task in many computational linguistic and text-technological applications. Lexical Chains, for example, are computed for a given text by identifying semantically related content words on the basis of a lexical-semantic resource [11]. In automatic hypertextualisation, they can be exploited to establish whether two sections of a text are thematically related or unrelated and thus whether they should end up in one hypertext module or in separate hypertext modules. Other applications of semantic relatedness are word sense disambiguation, discourse parsing, or information retrieval.

According to [7] and also cited in [11], *semantic relatedness* should be distinguished from *semantic similarity*. Two concepts (or via lexicalisation, lexemes) are semantically *similar* only if a synonymy or hypernymy relation holds between them. W.r.t the concepts and relations defined for TermNet and the plug-in relations defined for GermaTermNet in this chapter, we suggest to also count membership in one TermSet and an *attachedToNearSynonym* link as a sufficient criterion for semantic similarity. Examples of this are the pairs *aufgehen-sich öffnen*, *Gewässer-Meer*, *Link-Relation*, and *Dokument (GN)-Dokument (TN)*.

Two concepts are semantically *related* if any systematic relation such as synonymy, antonymy, hypernymy, holonymy, or any unsystematic, functional relation or frequent association holds, as in *day-night*, *tree-branch*, or *flower-gardener* [cf. 7, 11]; see also [12] in this volume. Semantic similarity is thus a subconcept of semantic relatedness.

How best to approximate semantic relatedness computationally is an area of current research. Several semantic relatedness measures have been defined, partly using WordNet (or more general, a wordnet) as a lexical resource. Overviews and evaluations of semantic relatedness measures are included in the recent publications by [7, 8], and [11]. To test the queriability of our OWL Full model of (merged) wordnets, we implemented Wu and Palmer's *conceptual similarity* [41]. It is defined as in the following notation found in [8].

$$wp(c_1,c_2) = \frac{2 * depth(lcs(c_1,c_2))}{len(c_1,lcs(c_1,c_2)) + len(c_2,lcs(c_1,c_2)) + 2depth(lcs(c_1,c_2))} \tag{17.1}$$

where $c_{1,2}$ are concepts in a wordnet, *lcs* is their lowest common superconcept, *len* is the length of a path along the hypernymy hierarchy, and *depth* is the length of the path to a given concept from the root node of the hypernymy tree.

Consequently, the implementation of *wp* in Prolog using the Thea library consists of three main functions: first, the construction of the hypernymy (or more general, superconcept) tree; second, the calculation of the depth of one concept node; and third, the calculation of the *lcs* of two concept nodes.

Generating the superconcept tree consists of inserting a root node and connecting it to those concepts that had no superconcepts in the first place. In GermaTermNet, the superconcept hierarchy is formed by four different path types:

1. the usual *isHyponymOf* relation between synsets in GermaNet
2. the *subclassOf* relation between term concepts in TermNet
3. the plug-in relation *attachedToNearSynonym* between term concepts in TN and synsets in GN
4. the plug-in relation *attachedToGeneralConcept* between term concepts in TN and synsets in GN

We implemented the calculation of depth and path lengths (*len*) in the *wp* formula such that an *attachedToNearSynonym* link counts 0, and the other three types of link count 1 (cf. Listing 17.11).

Concerning the *subclassOf* links between term concepts in TN, only those below the attachment points to GN are taken into account, i.e. the upper part of the term concept hierarchy is "eclipsed" in this application [cf. 26, 24]; otherwise, certain specialised terms would become situated too close to the root node and distort *wp* calculation.

Because the superconcept hierarchy includes multiple inheritance, there may be multiple depth values for one concept. We disambiguate this naturally by considering only the minimum depth in the calculation of *wp*. For the same reason, two concepts may have more than one *lcs*; in the implementation, we choose the one that is found first. Furthermore, the Thea OWL parser does not seem to be able to deal with *owl:import* specifications, so we worked with a version where the components of GermaTermNet were merged in one file.

The present implementation has some limitations: the GermaTermNet OWL Full ontology that we tested does not contain the complete GermaNet but only the representative subset mentioned in Section 17.2.1. The complete GermaNet could not be parsed and loaded into Prolog in reasonable time on the type of PCs that we used. We also noticed that the multiple *owl:disjointWith* specifications of TermNet delayed the parsing of GermaTermNet considerably. We thus removed them for *wp* calculation.

```
% direct_isSubConceptOf_GTN/3

direct_isSubConceptOf_GTN(A, B, 1):-
    direct_isHyponymOf_S(A, B).

direct_isSubConceptOf_GTN(A, B, 1):-
    attachedToGeneralConcept(A, B).

direct_isSubConceptOf_GTN(A, B, 0):-
    attachedToNearSynonym(A, B).

direct_isSubConceptOf_GTN(A, B, 1):-
    subclassOf(A, B).
%---
attachedToGeneralConcept(S, T):-
    individual(S, _, _, PrValueList),
    member(value(’attachedToGeneralConcept’, T), PrValueList).
```

Listing 17.11 Prolog code for calculating relatedness. *subclassOf/2*, *individual/4*, and *value/2* are predicates from the Thea library.

The Prolog coding of *wp* was straightforward using the OWL predicates that are made available by Thea. Examples of queries for a Wu-Palmer calculation of a concept pair and an occurrence pair are given in Listing 17.12.

```
?- wp(’tn:TermConcept_extensional_definierter_Verweis’,
     ’tn:TermConcept_Beziehung’, W).
   W = 0.571429

?- wpo(’tn:URI_TermInstance_extensional_definierter_Verweis_1’,
       ’gn:URI_LUnit_Instance_aVerhalten.239.arrogant_2’, W).
   W = 0.222222.
```

Listing 17.12 Semantic relatedness queries in Prolog, wp = query with concepts, wpo = query with occurrences.

Thus, the OWL Full model of a wordnet merged in OWL according to the plug-in approach can be processed in the Prolog-plus-semantic-web-libraries framework. A number of further interesting observations was made as well:

In the implementation, it is fairly easy to extend or reduce the set of link types used for the calculation of the superconcept hierarchy and to specify their respective distance counts. This allows for e.g. including other relations like the holonymy relation in the calculation of the superconcept hierarchy or for a different weighting of plug-in links as opposed to GermaNet links as opposed to TermNet links.

It was also straightforward to "eclipse" a part of the GermaTermNet ontology dynamically as part of the *wp* application (*eclipse* was originally described as a static effect in the merger of a general with a specialised wordnet in [26]).

Furthermore, *wp* calculation for a pair of concepts could be projected to *wp* calculation for a pair of occurrences easily by making reference to the properties that represented occurrence in Prolog. This is not an advantage of the OWL Full model, though, as the implementation would be similar for the instance model. However, processing property value specifications in the OWL Full model is a true advantage over processing the OWL class restrictions of the class model.

17.5 Conclusion

In this paper, we discussed modelling and processing issues of wordnets represented in the web ontology language *OWL*. We introduced the features of an Instance Model, a Class Model, and a Metaclass Model for rendering wordnets in OWL by the example of the three resources that we have been working with in the context of two projects of the DFG research group 437 "Text-Technological Modelling of Information", namely GermaNet, TermNet, and GermaTermNet.

In the Instance Model, which is favoured for the Princeton WordNet for English by the W3C Semantic Web Best Practices and Deployment Group, synsets and lexical units are rendered as OWL individuals, and the links representing lexical-semantic relations are rendered as simple property assignments. A drawback of this model is that occurrence individuals cannot be modelled as instances of their lexical units. Instead, an additional occurrence class must be introduced, and occurrences are linked in a distinguished *occursAs* relation. Moreover, the class/individual distinction in OWL cannot be used to model wordnet instances as introduced for PWN 2.1 (cf. [27]).

In the Class Model, on the other hand, synsets and lexical units are rendered as OWL classes. Many domain ontologies or terminologies in OWL are represented according to this model (e.g. the GOLD ontology for linguistic description, cf. [15], or TermNet, cf. [25]). With regard to lexical-semantic networks, the Class Model is best compatible with the traditional view of lexical semantics according to which a noun semantically denotes a concept class. That means that even wordnet instances as described in [27] could be incorporated naturally as OWL individuals of synset classes. However, lexical-semantic and conceptual relations as well as the lexicalisation relation and the plug-in relations of GermaTermNet must all be encoded using OWL property restrictions on synset and LU classes, which is a less than ideal solution when modelling simple wordnet relation assignments. Moreover, when introducing occurrence individuals of lexical unit classes (Section 17.3.3.2), it seems inadequate that these also inherit all the lexical properties of the LUs such as style marking or the lexical-semantic relations relation assignments.

Thus, in view of the drawbacks of the Instance and Class Models, we introduced the Metaclass Model for wordnets in OWL. It offers a combination of the advantages of the Class and Instance Model because synsets and lexical units are viewed as classes and individuals at the same time. On the other hand, it introduces a new practical problem as a wordnet encoded according the Metaclass Model is in the dialect OWL Full and thus not processable by most standard DL-based reasoning software.

In Section 17.4, we examined ways to process the different models for wordnets in OWL in two scenarios. In the first scenario, we evaluated the performance of queries for related term occurrences in documents such as they are typically queried on TermNet in the hypertextualisation application. For querying the Class Model and the Instance Model, we employed the query language nRQL with the DL reasoner software RacerPro, and the rule language SWRL with the java-based rule engine Jess. The rules and queries for the Metaclass Model were all formulated in Prolog, using SWI Prolog and the semweb.pl and Thea libraries.

In general, the formulation of queries in nRQL was less complex on the Class Model than on the Instance Model, mainly due to the rendering of occurrence individuals as instances of lexical unit classes in the Class Model. However, it turned out that the Class Model still had to be extended by dummy individuals to get the desired results for our queries. Also the processing of all three queries took about one-third longer than processing the corresponding queries on the Instance Model.

For querying the Metaclass Model, DL reasoners such as RacerPro could not be employed; thus we used Prolog and its Semantic Web Library as a rule and query language. Formulating the queries in the first scenario was straightforward but still somewhat more complex than querying the Class Model in nRQL due to the way property assignments are stored in Prolog facts for individuals by the Thea OWL parser. The time it took to execute the queries lay between the times for querying the Class Model and the Instance Model using nRQL and RacerPro.

In the second scenario, we evaluated the feasibility of implementing the more complex text-technological task of calculating semantic similarity on the combined ontology GermaTermNet in the Instance+Instance Model (using SWRL rules) and the Metaclass+Metaclass Model (using SWI Prolog and its semantic web libraries).

Unlike in the Class Model (where OWL property restrictions must be checked, cf. [24]), combinations of path types can easily be analysed on the Thea representation of the Metaclass+Metaclass Model of GermaTermNet in Prolog (and the same would hold for the Instance+Instance Model).

Although SWRL is a logical rule language based on Horn logic like Prolog and also includes a built-in function library, it was impossible to approximate semantic similarity calculation on the Instance+Instance Model using only SWRL. The reason is that SWRL lacks all the general programming language features of Prolog, such as binding temporary results to a variable and changing the variable content, or a maximum function, which is needed in the calculation of semantic similarity. To implement this on the basis of SWRL, one would have to embed the OWL ontology+SWRL rules in a programming language environment as well, e.g. a Java application using the SWRL-Java API[17].

Based on the findings of the experiments reported in this chapter, we favour the Metaclass Model for wordnets in OWL and process them using the Prolog semantic web libraries semweb and Thea. However, one has to keep in mind that Thea is not a W3C-endorsed effort and that the semantics of pure Prolog differ from the semantics of OWL in various respects, such as Prolog's closed world assumption vs. an open world assumption for OWL.

Using SWRL, on the other hand, may be more sustainable as it has the status of a W3C member submission and is more easily combinable with OWL. More and more implementations of SWRL features and APIs are likely to emerge in the future so that it might become easier to integrate OWL+SWRL ontologies into procedural programming applications as well.

References

[1] van Assem, M., Menken, M.R., Schreiber, G., Wielemaker, J., Wielinga, B.J.: A method for converting thesauri to RDF/OWL. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 17–31. Springer, Heidelberg (2004)

[2] van Assem, M., Gangemi, A., Schreiber, G.: Conversion of WordNet to a standard RDF/OWL representation. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy (2006)

[3] van Assem, M., Gangemi, A., Schreiber, G.: RDF/OWL Representation of WordNet. W3C Public Working Draft of June 19 2006 of the Semantic Web Best Practices and Deployment Working Group (2006), http://www.w3.org/TR/wordnet-rdf/ (visited January 15, 2010)

[4] Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, International Handbooks on Information Systems, pp. 3–28. Springer, Heidelberg (2004)

[5] Beißwenger, M.: Ein wortnetzbasierter Ansatz für die korpusgestützte Modellierung von Fachterminologie. Ein Beitrag zur digitalen Fachlexikographie. Zeitschrift für germanistische Linguistik 38(3), 346–369 (2010) (Themenheft Semantik und Lexikographie)

[6] Beißwenger, M., Storrer, A., Runte, M.: Modellierung eines Terminologienetzes für das automatische Linking auf der Grundlage von WordNet. LDV-Forum 19(1-2), 113–125 (2004)

[7] Budanitsky, A., Hirst, G.: Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures. In: Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL), Pittsburgh (2001)

[8] Budanitsky, A., Hirst, G.: Evaluation wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)

[9] Ciorăscu, C., Ciorăscu, I., Stoffel, K.: knOWLer – Ontological support for information retrieval systems. In: Proceedings of the 26th Annual International ACM-SIGIR Conference, Workshop on Semantic Web, Toronto, Canada (2003)

[17] http://www.daml.org/rules/proposal/jaxb/

[10] Cioră̆scu, I., Cioră̆scu, C., Stoffel, K.: Scalable ontology implementation based on knOWLer. In: Proceedings of the 2nd International Semantic Web Conference (ISWC2003). Workshop on Practical and Scalable Semantic Systems, Sanibel Island, Florida (2003)

[11] Cramer, I., Finthammer, M.: An evaluation procedure for word net based lexical chaining: Methods and issues. In: Proceedings of the Global Wordnet Conference - GWC 2008, Szeged, Hungary, pp. 121–146 (2008)

[12] Cramer, I., Wandmacher, T., Waltinger, U.: Exploring resources for lexical chaining: A comparison of automated semantic relatedness measures and human judgments. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modelling, Learning and Processing of Text-Technological Data Structures. Springer, Berlin (2011)

[13] De Luca, E.W., Eul, M., Nürnberger, A.: Converting EuroWordNet in OWL and Extending it with Domain Ontologies. In: Kunze, C., Lemnitzer, L., Osswald, R (eds.) Proceedings of the GLDV-2007 Workshop on Lexical-Semantic and Ontological Resources, Fern-Universität Hagen, Hagen, Informatik. Berichte, vol. 336(3), pp. 39–48 (2007)

[14] Erdmann, M.: Ontologien zur konzeptuellen Modellierung der Semantik von XML. Books on Demand, Karlsruhe (2001)

[15] Farrar, S., Langendoen, D.T.: A linguistic ontology for the semantic web. GLOT International 7(3), 97–100 (2003)

[16] Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)

[17] Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)

[18] Haarslev, V., Möller, R.: RACER User's Guide and Reference Manual Version 1.7.19. Tech. rep., Technische Universität Hamburg-Harburg, with contributions from Michael Wessel (2004), http://www.sts.tu-harburg.de/r.f.moeller/racer/ (visited January 15, 2010)

[19] Haarslev, V., Möller, R., Wessel, M.: Querying the Semantic Web with Racer + nRQL. In: Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL 2004), Ulm, Germany, September 24 (2004)

[20] Horrocks, I., Tobies, S.: Reasoning with axioms: Theory and practice. In: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), pp. 285–296. Morgan Kaufmann, San Francisco (2000)

[21] Kunze, C., Lemnitzer, L.: GermaNet - representation, visualization, application. In: Proceedings of LREC, Las Palmas, vol. V, pp. 1485–1491 (2002)

[22] Kunze, C., Lemnitzer, L., Lüngen, H., Storrer, A.: Repräsentation und Verknüpfung allgemeinsprachlicher und terminologischer Wortnetze in OWL. Zeitschrift für Sprachwissenschaft 26(2) (2007)

[23] Lobin, H., Lüngen, H., Hilbert, M., Bärenfänger, M.: Processing text-technological resources in discourse parsing. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modelling, Learning and Processing of Text-Technological Data Structures. Springer, Berlin (2011)

[24] Lüngen, H., Storrer, A.: Domain ontologies and wordnets in OWL: Modelling options. LDV-Forum GLDV-Journal for Computational Lingjistics and Language Technologie 22(2), 1–17 (2008)

[25] Lüngen, H., Kunze, C., Lemnitzer, L., Storrer, A.: Towards an integrated OWL model for domain-specific and general language wordnets. In: Proceedings of the Fourth Global WordNet Conference (GWC 2008), Szeged, Hungary, pp. 281–296 (2008)

[26] Magnini, B., Speranza, M.: Merging Global and Specialized Linguistic Ontologies. In: Proceedings of Ontolex 2002, Las Palmas de Gran Canaria, Spain, pp. 43–48 (2002)

[27] Miller, G.A., Hristea, F.: Word net nouns: Classes and instances. Computational Linguistics 32(1), 1–3 (2006)

[28] Motik, B.: On the properties of metamodeling in OWL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 548–562. Springer, Heidelberg (2005)

[29] Noy, N.: Representing classes as property values on the semantic web. W3C Working Group Note (2005), http://www.w3.org/TR/swbp-classes-as-values/ (visited January 15, 2010)

[30] Pan, J.Z., Horrocks, I., Schreiber, G.: OWL FA: A metamodeling extension of OWL DL. In: Proceedings of the Workshop OWL: Experiences and directions, Galway, Ireland (2005)

[31] Schreiber, G.: The web is not well-formed. IEEE Intelligent Systems 17(2), 79–80 (2002)

[32] Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 51–53 (2007)

[33] Smith, M.K., Welty, C., Deborah, L.: McGuinness e OWL Web Ontology Language – Guide. Tech. rep. W3C (World Wide Web) Consortium (2004), http://www.w3.org/TR/2004/REC-owl-guide-20040210/ (visited January 15, 2010)

[34] Sowa, J.F.: Knowledge Representation. Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove (2000)

[35] Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems. Springer, Heidelberg (2004)

[36] Storrer, A.: Mark-up driven strategies for text-to-hypertext conversion. In: Metzing, D., Witt, A. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, Text, Speech and Language Technology. Springer, Dordrecht (2010)

[37] Vassiliadis, V.: Thea. A web ontology language - OWL library for (SWI) Prolog. Web-published manual (2006), http://www.semanticweb.gr/TheaOWLLib/ (visited January 15, 2010)

[38] Vossen, P.: EuroWordNet: a mutlilingual database with lexical-semantic networks. Kluwer Academic Publishers, Dordrecht (1999)

[39] Wielemaker, J.: SWI-Prolog Semantic Web Library (2005), http://www.swi-prolog.org/pldoc/package/semweb.html (visited January 15, 2010)

[40] Wielemaker, J., Schreiber, G., Wielinga, B.J.: Prolog-based infrastructure for RDF: Scalability and performance. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 644–658. Springer, Heidelberg (2003)

[41] Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics - ACL 1994, New Mexico, pp. 133–138 (1994)

Chapter 18
Exploring Resources for Lexical Chaining: A Comparison of Automated Semantic Relatedness Measures and Human Judgments

Irene Cramer, Tonio Wandmacher, and Ulli Waltinger

Abstract. In the past decade various semantic relatedness, similarity, and distance measures have been proposed which play a crucial role in many NLP-applications. Researchers compete for better algorithms (and resources to base the algorithms on), and often only few percentage points seem to suffice in order to prove a new measure (or resource) more accurate than an older one. However, it is still unclear which of them performs best under what conditions. In this work we therefore present a study comparing various relatedness measures. We evaluate them on the basis of a human judgment experiment and also examine several practical issues, such as run time and coverage. We show that the performance of all measures – as compared to human estimates – is still mediocre and argue that the definition of a shared task might bring us considerably closer to results of high quality.

18.1 Motivation

The computation of semantic relatedness (SR) has become an important task in many NLP-applications such as spelling error detection, automatic summarization, word sense disambiguation, and information extraction. In recent years a large

Irene Cramer
Institute for German Language and Literature, Technische Universität Dortmund,
Emil-Figge-Straße 50, D-44221 Dortmund, Germany
e-mail: irene.cramer@tu-dortmund.de

Tonio Wandmacher
Systran S.A.
Paris, France
e-mail: tonio.wandmacher@gmail.com

Ulli Waltinger
Faculty of Technology, Bielefeld University, Universitätsstraße 25,
D-33615 Bielefeld, Germany
e-mail: ulli_marc.waltinger@uni-bielefeld.de

A. Mehler et al. (Eds.): Modeling, Learning, and Proc. of Text-Tech. Data Struct., SCI 370, pp. 377–396.
springerlink.com

variety of approaches in computing SR has been proposed. However, algorithms and results differ depending on resources and experimental setup.

It is obvious that SR plays a crucial role in the lexical retrieval of humans. In various priming experiments it could be shown that semantically related terms influence the semantic processing of each other (e.g. if "*street*" is primed by "*car*" it is recognized more quickly). Moreover, many theories of memory are based on the notion of SR. The spreading activation theory of Collins & Loftus [12] for example groups lexical items according to their SR in a conceptual graph. Similar ideas can be found in Anderson's *Adaptive Control of Thought* (ACT) theory [1].

The question that we want to discuss here is, how this kind of relatedness can be determined by automatic means. In the literature the notion of SR is often confounded with semantic *similarity* [5, pp 3]. There is however a clear distinction between these terms. Two terms are semantically similar if they behave similarly in a given context and if they share some aspects of meaning (e.g. in the case of synonyms or hypernyms). On the other hand two terms can be semantically strongly related without behaving similarly. For example they can show a strong associative relationship (e.g. *ball* - *goal*), and they can be related over different linguistic categories (e.g. *milk* - *white, dog* - *bark*). That is, *semantic similarity* can be seen as *a special case of semantic relatedness* [37]. With respect to the automatic computation of SR, however, many research questions remain unanswered. As stated above, many algorithms were presented in the past decade, but thorough evaluations and comparisons of their ability to capture SR in a human-like manner are still rare.

In this work we therefore present a study comparing various SR measures. We evaluate sixteen different algorithms involving four different resources based on a human judgment experiment, and we analyze the algorithms from a theoretical and practical point of view.

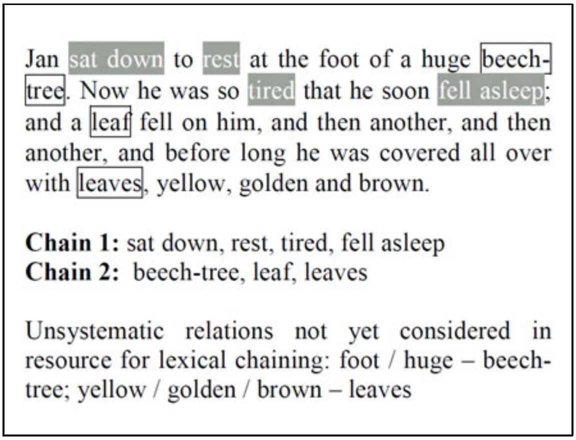
We perform this evaluation in the context of *lexical chaining*, a task that aims to determine sequences of terms which are semantically interrelated in a given text. Such term sequences (or chains) represent an important intermediate structure for purposes involving higher-order semantic analysis.

The following sections are organized as follows: The principle concepts of lexical chaining are introduced in Section 18.2. Section 18.3 outlines related work by means of two different human judgment experiments. The various SR measures used in our experiment are described in Section 18.4. An evaluation experiment and its results are presented in Section 18.5. Conclusions and suggestions for future work are given in Section 18.6.

18.2 Lexical Chaining

Based on the notion of lexical *cohesion*, as described by Halliday and Hasan in 1976 [20], computational linguists, e.g. Morris and Hirst [32], developed in the 1990s a method to compute partial text representations: *lexical chains*. To illustrate the idea of lexical chains, an annotated text passage is given in Figure 18.1. These chains (e.g. *sit down* - *rest* - *tired* - *fall asleep*) consist of semantically related terms, and they describe the cohesive structure of a given text. They can be constructed automatically by linking lexical items with respect to the SR holding between them. Many approaches of lexical chaining employ a lexical-semantic resource such as Princeton *WordNet* (cf. [16], which has been used in the majority of cases, e.g. [21], [18], [44]), *Roget's Thesaurus* (e.g. [32]), or the open encyclopedia *Wikipedia* and its offshoot *Wiktionary*[1] (e.g. [52]).

However, since the construction of lexical chains does not necessarily depend on explicit relation types, distributional SR measures (such as PMI or LSA, cf. Section 18.4.2) represent an alternative resource for the calculation of lexical chains.



Jan sat down to rest at the foot of a huge beech-tree. Now he was so tired that he soon fell asleep; and a leaf fell on him, and then another, and then another, and before long he was covered all over with leaves, yellow, golden and brown.

Chain 1: sat down, rest, tired, fell asleep
Chain 2: beech-tree, leaf, leaves

Unsystematic relations not yet considered in resource for lexical chaining: foot / huge – beech-tree; yellow / golden / brown – leaves

Fig. 18.1 Chaining example adapted from Halliday and Hasan's work [20]

A variety of NLP-applications, namely text summarization (e.g. [3], [42]), malapropism recognition (e.g. [21]), automatic hyperlink generation (e.g. [18]), question answering (e.g. [35]), and topic detection or tracking (e.g. [8]), benefit from lexical chaining as a valuable resource and preprocessing step.

In order to formally evaluate the performance of a lexical chaining system, a standardized test set would be required. However, the output of a chainer is normally assessed with respect to an application; although in several works more general evaluation criteria have been proposed (e.g. [6], [14]), no consensus among researchers could yet be achieved, and, consequently, the different sets of criteria have not yet been systematically applied.

18.3 Related Work

Semantic Relatedness constitutes a phenomenon, which can be observed practically in every text or discourse; while it is easy to approach intuitively, it is rather difficult

[1] http://www.wikipedia.org

to detect and analyze in a formal manner. Although many researchers of different scientific communities have proposed various approaches intended to formalize and compute structures of semantic relatedness, there are nevertheless only few prominent strands of research which differ especially with respect to the names attributed to the phenomenon, the features and types of relations subsumed, and last but not least the methods to deal with it. We argue that all these works might best be split into two groups: firstly, research intended to understand, describe, and finally formalize the underlying concepts (cf. [33]), and secondly, studies mainly focused on technical aspects, namely efficient algorithms (cf. [42]) and promising resources (cf. [52]). In the following we describe in depth one prominent work of each of the two strands of research, in order to illustrate the central issues under discussion.

18.3.1 Budanitsky and Hirst

Budanitsky and Hirst's work [6] aims at an extensive comparison of the performance of various SR measures, i.e. different algorithms. For this purpose, Budanitsky & Hirst indicate three evaluation methods: firstly, the theoretical examination (of e.g. the mathematical properties of the respective measure); secondly, the comparison with human judgments; thirdly, the evaluation of a measure with respect to a given NLP-application. In their opinion the second and third method are the most appropriate ones; they therefore focus on them in the empirical work presented. As a basis for the second evaluation method, i.e. the comparison between SR measures and human judgments, they use two lists of word pairs: the first has been compiled by Rubenstein and Goodenough [38] and contains 65 word pairs[2], while the second, containing 30 word pairs, has been created by Miller and Charles [30]. In order to evaluate the performance of five different measures, Budanitsky and Hirst [6] compute respective relatedness values for the word pairs, and they compare them with the human judgments. In this way they determine the correlation coefficients summarized in Table 18.1.

Table 18.1 Correlation Coefficients by Budanitsky and Hirst

r	Leacock-Chodorow	Hirst-St-Onge	Resnik	Jiang-Conrad	Lin
M&C	0.816	0.744	0.774	0.850	0.82
R&G	0.838	0.786	0.779	0.781	0.819
mean	0.83	0.77	.78	0.82	0.82

[2] Rubenstein & Goodenough [38] investigated the relationship between ’*similarity of context*’ and ‘*similarity of meaning*’. They asked 51 subjects to rate on a scale of 0 to 4 the similarity of meaning for the 65 word pairs. Miller and Charles [30] selected 30 out of the 65 original word pairs (according to their relatedness strength) and asked 38 subjects to rate this list. They used the same experimental setup as [38].

In examining the results of this comparison, Budanitsky and Hirst identify several limitations of this evaluation method: i.e. they stress that the amount of data available (65 word pairs) is inadequate for real NLP-applications, however, the development of a large-scale data set would be time-consuming and expensive (cf. Section 18.5.1). Moreover, they argue that the experiments by Rubenstein and Goodenough [38] as well as Miller and Charles [30] focus on relations between words rather than word-senses (concepts), which would be more appropriate for most NLP-applications. Nevertheless, they consider it difficult to trigger a specific concept without biasing the subjects.

18.3.2 Boyd-Graber et al.

In contrast to the above-mentioned experiments by Budanitsky and Hirst [6], the research reported by Boyd-Graber et al. [4] strives for the development of a new, conceptually different layer of word net relation types and is motivated by three widely acknowledged, yet unsolved challenges:

- The lack of cross-POS links connecting the noun, verb, and adjective sub-graphs respectively.
- The lack of systematic relation types (such as “actor” or “instrument”), causing a low relation density of relations, especially in the subgraphs.
- The absence of weights assigned to the relations, i.e. representing the degrees of semantic distance.

Unlike Rubenstein and Goodenough [38] or Miller and Charles [30], Boyd-Graber et al. do not restrict themselves to systematic relation types but introduce the concept of *evocation*[3]. While systematic relations are well defined[4] , evocation seemingly represents a diffuse accumulation of various aspects, which intuitively account for semantic relatedness but can only in parts be precisely characterized.

Table 18.2 Correlation Coefficients by Boyd-Graber et al. [4].

r	Lesk	Path	LC	LSA
all	0.008			
verbs		0.046		
nouns		0.013	0.013	
closest				0.131

[3] Boyd-Graber et al. define this term in a rather loose sense as “*how much one concept evokes or brings to mind the other*”.

[4] Relations which can be extrinsically defined and verified, e.g. by replacement tests. WordNet and other lexical-semantic resources normally comprise systematic relations, such as synonymy or antonymy, only.

In their experiment, Boyd-Graber et al. asked 20 subjects to rate evocation in 120,000 pairs of words (these pairs form a random selection of all possible word pairs stemming from 1000 core synsets in WordNet). The subjects were given a detailed manual explaining the task, and they were trained on a sample of 1000 (two sets of 500) randomly selected pairs. Although the research objective of their work is to construct a new layer of relations for WordNet rather than to evaluate SR measures, Boyd-Graber et al. compare the results of their human judgment experiment with the relatedness values of four different semantic measures. The correlation coefficients of this comparison are summarized in Table 18.2.

Boyd-Graber et al. arrive at the conclusion that – given the obvious lack of correlation (cf. Table 18.2) – evocation constitutes an empirically supported semantic relation type which is still not captured by the semantic measures (at least not by those considered in this experiment).

While the first work mentioned above discusses in detail various SR algorithms, provides a survey of evaluation methods, and finally presents their applicability from a technical point of view, the second additionally sheds light on the linguistic and psycholinguistic aspects of the set-up of a human assessment experiment and the comparison between relatedness values and human judgments.

18.4 Semantic Relatedness Measures

18.4.1 Net-Based Measures

The following eight SR measures draw on a lexical-semantic net like Princeton *WordNet* or its German counterpart *GermaNet* [26]. Although all of these measures are based on the same resource, they use different features (some additionally rely on a word frequency list[5]) and therefore also cover different aspects of SR.

Most of the relatedness measures mentioned in this section are continuous, with the exception of *Hirst-StOnge*, *Tree-Path*, and *Graph-Path* which are discrete.

All of the measures range in a closed interval between 0 (not related) and a maximum value (mostly 1), or they can be normalized: the distance value calculated by the three distance measures (*Jiang-Conrath*, *Tree-Path*, and *Graph-Path*) is mapped into a closed range relatedness value by subtracting it from the theoretical maximum distance.

The first four measures use a hyponym-tree induced from a given lexical-semantic net, i.e. all other edges except the hyponym links are disregarded. The resulting unconnected trees are subsequently reconnected by an artificial root in order to construct the required hyponym-tree.

[5] We used a word frequency list computed by Dr. Sabine Schulte im Walde on the basis of the *Huge German Corpus* (see
`http://www.schulteimwalde.de/resource.html`).
We thank Dr. Schulte im Walde for kindly permitting us to use this resource in the framework of our project.

- **Leacock-Chodorow** [25]: Given a hyponym-tree, this measure computes the length of the shortest path between two synonym sets and scales it by the depth of the complete tree.

$$\mathrm{rel}_{\mathrm{LC}}(s_1, s_2) = -\log \frac{2 \cdot \mathrm{sp}(s_1, s_2)}{2 \cdot D_{Tree}} \tag{18.1}$$

  s_1 and s_2: the two synonym sets examined; $\mathrm{sp}(s_1, s_2)$: length of shortest path between s_1 and s_2 in the hyponym-tree; D_{Tree}: depth of the hyponym-tree
- **Wu-Palmer** [49]: Given a hyponym-tree, the *Wu-Palmer* measure utilizes the least common subsumer in order to compute the similarity between two synonym sets. The least common subsumer is the deepest vertex which is a direct or indirect hypernym of both synonym sets.

$$\mathrm{rel}_{\mathrm{WP}}(s_1, s_2) = \frac{2 \cdot \mathrm{depth}(\mathrm{lcs}(s_1, s_2))}{\mathrm{depth}(s_1) + \mathrm{depth}(s_2)} \tag{18.2}$$

  $\mathrm{depth}(s)$: length of the shortest path form root to vertex s; $\mathrm{lcs}(s)$: least common subsumer of s
- **Resnik** [37]: Given a hyponym-tree and the frequency list mentioned above, the *Resnik* measure utilizes the information content in order to compute the similarity between two synonym sets. As typically defined in information theory, the information content is the negative logarithm of the probability. Here the probability is calculated on the basis of subsumed frequencies. A subsumed frequency of a synonym set is the sum of frequencies of the set of *all* words which are in this synonym set, *or* a direct or indirect hyponym synonym set.

$$\mathrm{p}(s) := \frac{\sum_{w \in W(s)} \mathrm{freq}(w)}{TotalFreq} \tag{18.3}$$

$$\mathrm{IC}(s) := -\log \mathrm{p}(s) \tag{18.4}$$

$$\mathrm{rel}_{\mathrm{Res}}(s_1, s_2) = \mathrm{IC}(\mathrm{lcs}(s_1, s_2)) \tag{18.5}$$

  $\mathrm{freq}(w)$: frequency of a word within a corpus; $W(s)$: set of the synonym set s and all its direct/indirect hyponym synonym sets; *TotalFreq*: sum of the frequencies of all words in the respective lexical-semantic net; $\mathrm{IC}(s)$: information content of the synonym set s
- **Jiang-Conrath** [22]: Given a hyponym-tree and the frequency list mentioned above, the *Jiang-Conrath* measure computes the distance (as opposed to similarity) of two synonym sets. The information content of each synonym set is included separately in this distances value, while the information content of the least common subsumer of the two synonym sets is subtracted.

$$\mathrm{dist}_{\mathrm{JC}}(s_1, s_2) = \mathrm{IC}(s_1) + \mathrm{IC}(s_2) - 2 \cdot \mathrm{IC}(\mathrm{lcs}(s_1, s_2)) \tag{18.6}$$

- **Lin** [28]: Given a hyponym-tree and the frequency list mentioned above, the *Lin* measure computes the SR of two synonym sets. As the formula clearly shows, the same expressions are used as in *Jiang-Conrath*. However, the structure is different, as the expressions are divided and not subtracted.

$$\mathrm{rel}_{\mathrm{Lin}}(s_1,s_2) = \frac{2 \cdot \mathrm{IC}(\mathrm{lcs}(s_1,s_2))}{\mathrm{IC}(s_1)+\mathrm{IC}(s_2)} \tag{18.7}$$

- **Hirst-StOnge** [21]: In contrast to the four above-mentioned methods, the *Hirst-StOnge* measure computes the semantic relatedness on the basis of the whole graph structure. It classifies the relations considered into the following four classes: *extra strongly related* (the two words are identical), *strongly related* (the two words are e.g. synonym or antonym), *medium strongly related* (there is a relevant path between the two), and *not related* (there is no relevant path between the two). The relatedness values in the case of extra strong and strong relations are fixed values, whereas the medium strong relation is calculated based on the path length and the number of changes in direction.
- **Tree-Path** (Baseline 1): Given a hyponym-tree, the simple *Tree-Path* measure computes the length of a shortest path between two synonym sets. Due to its simplicity, the Tree-Path measure serves as a baseline for more sophisticated similarity measures.

$$\mathrm{dist}_{\mathrm{Tree}}(s_1,s_2) = \mathrm{sp}(s_1,s_2) \tag{18.8}$$

- **Graph-Path** (Baseline 2): Given the whole graph structure of a lexical-semantic net, the *Graph-Path* measure calculates the length of a shortest path between two synonym sets in the whole graph, i.e. the path can make use of all relations available. Analogous to Tree-Path, the Graph-Path measure gives us a very rough baseline for other relatedness measures.

$$\mathrm{dist}_{\mathrm{Graph}}(s_1,s_2) = \mathrm{sp}_{Graph}(s_1,s_2) \tag{18.9}$$

$\mathrm{sp}_{Graph}(s_1,s_2)$: Length of a shortest path between s_1 and s_2 in the graph

In the task of determining SR, humans do not seem to distinguish between systematic relations (e.g. synonymy or hyponymy) and unsystematic ones: either a pair of words is (more or less) related or it is not (cf. [30] and [34]). However, a net like *GermaNet* only models systematic relations such as hyponymy or meronymy. Unsystematic (i.e. associative) connections are not directly taken into account in any of the measures mentioned above. We therefore expect all of them to produce many false negatives, i.e. low relation values for word pairs which are judged by humans to be (strongly) related.

18.4.2 Distributional Measures

A recent branch of lexical semantics aims to exploit statistics of word usage to derive meaning. Based on the assumption that words with similar distributional properties have similar meanings, such approaches infer semantic relatedness from the co-occurrence of words in text corpora. Distributional similarity can be defined in (at least) two ways: One group of measures establishes relatedness on direct co-occurrence in text (1^{st} order relatedness); many of these measures can be related to standard statistical tests. The other group aims to compare the similarity of contexts in which two terms occur (2^{nd} order relatedness); such measures usually operate on the vector space. In the following an overview of the different 1^{st} and 2^{nd} order measures is given:

- **Pointwise Mutual Information**: A typical representative of a 1^{st} order measure is *pointwise mutual information* (PMI) [10]. Here, the co-occurrence probability of two terms is set in relation to the probability of the singular terms. In [36] and [46] it could be shown that 1^{st} order measures are able to determine semantically related terms, even though the relations tend to be of syntagmatic nature.

$$rel_{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i) \times P(w_j)} \tag{18.10}$$

  Where $P(w_i)$, $P(w_i, w_j)$ is the probability estimate of a key word w_i or a word pair w_i, w_j. Since the reliability of the collected parameters usually grows with the size of the training corpus (cf. for example [7]), it was also proposed to use the web as a corpus [45]. In such settings *hit counts*, as the number of pages found by search engine for a given query, are used as the probability parameter.

- **Normalized Search Distance** (NSD) [11]: This measure is inherently based upon the idea of using hit counts from a search engine. As the web-based PMI measure, NSD is calculated from the singular ($hc(w_i)$) and the joined ($hc(w_i, w_j)$) hit counts as well as the total number of pages M.

$$rel_{NSD}(w_i, w_j) = \frac{max[\log hc(w_i) \log hc(w_j)] - \log hc(w_i, w_j)}{\log M - \min[\log hc(w_i), \log hc(w_j)]} \tag{18.11}$$

- **Google Quotient**: Another measure has been proposed by [14], the *Google quotient*. It is defined as follows:

$$rel_{GQ}(w_i, w_j) = \frac{2 \cdot hc(w_i, w_j)}{hc(w_i) + hc(w_j)} \tag{18.12}$$

  Again, $hc(w_i)$, $hc(w_i, w_j)$ are the hit counts of a key word w_i or a word pair w_i, w_j.

- **Latent Semantic Analysis (LSA)** [15]: Among the 2^{nd} order approaches *Latent Semantic Analysis* has obtained particular attention, due to its success in a large variety of tasks involving semantic processing. When it was first presented by Deerwester et al. [15], it aimed mainly at improving the vector space model in information retrieval (cf. [39]), but in the meantime it has become a helpful tool in NLP as well as in cognitive science (cf. [24]). As the vector space model, LSA is based on a term×context matrix A, displaying the occurrences of each word in each context. When only term relationships are considered, a slightly different setting, as described by Schütze [41] and Cederberg & Widdows [9] is more appropriate; here the original matrix is not based on occurrences of terms in documents, but on other co-occurring terms (term×term-matrix). We thus count the frequency with which a given term occurs with others in a predefined context window ($\pm 10 - 100$ words).

  The decisive step in the LSA process is then a *singular value decomposition* (SVD) of the matrix, which enhances the contrast between reliable and unreliable relations. The high-dimensional input matrix is hereby reduced to a subspace of k dimensions ($k \approx 100$ to 300). After applying SVD, each word is represented as a k-dimensional vector, and for every word pair w_i, w_j of our vocabulary we can calculate a relatedness value $rel_{LSA}(w_i, w_j)$, based on the *cosine* measure. The cosine of the angle between any two vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ of dimensionality m with components w_{ik} und w_{jk}, $k \leq m$ is defined as follows:

$$\cos(\mathbf{w_i}, \mathbf{w_j}) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \sum_{k=1}^{m} w_{jk}^2}} \qquad (18.13)$$

  Since the denominator normalizes the vector length, frequency influences are leveled out. In addition, the result becomes standardized ($[-1;1]$), which facilitates further comparisons.

- **Semantic Vectors (Sem.Vec.)** [47]: The open source *Semantic-Vectors* package[6] creates a word space model from a term-document matrix using a random projection algorithm. It is supposed to perform similarly to techniques like LSA but it does not rely on complex procedures such as SVD, making it a more scalable technique. Word similarity is performed by producing a query vector and calculating its distance to the term vectors (using the cosine).

The important advantage of 2^{nd} order approaches is that they are better able to capture paradigmatic relations such as synonymy or hyponymy, since paradigmatically similar words tend to occur in similar contexts. However, they also have a disadvantage with respect to direct co-occurrence measures, because the matrix computations are computationally demanding, so that they cannot be performed online. This means that usually far smaller training corpora must be used.

[6] http://code.google.com/p/semanticvectors/

18.4.3 Wikipedia-Based Measures

With regard to *Wikipedia*-based SR computation, some approaches have been proposed which mainly focus either on the hyperlink structure [31], the vector space model (VSM), or on category concepts for graph related measures [43, 51]. We have implemented three different algorithms using Wikipedia as a resource in computing semantic relatedness:

- **Explicit Semantic Analysis (ESA)** [17]: This method represents term similarity by an inverted term-document index in a high-dimensional space of concepts derived from Wikipedia. In this case, concepts are defined as Wikipedia articles. Each concept is represented as an attribute vector of terms occurring in the corresponding article (weighted by a *tf.idf* scheme [39]). Semantic relatedness of a pair of terms is computed by comparing their respective concept vectors using the cosine metric (cf. equation 18.13). We have adopted the approach of Gabrilovich and Markovitch [17] to the German Wikipedia data (lemmatized). We have also removed small and overly specific concepts (articles having fewer than 100 words and fewer than 5 hyperlinks), leaving 126,475 articles on which the inverted index was built.
- **Wikipedia Graph-Path**: This measure operates on the Wikipedia hyperlink graph $G_w = (V, E)$, where Wikipedia articles denote a set of vertices V, and hyperlinks between articles denote a set of edges $E \subseteq V^2$. The Wikipedia Graph-Path distance calculates the length of the shortest path (sp) between two articles in G_w.
$$distW_{Gw}(v1, v2) = \mathrm{sp}_{Gw}(v1, v2) \tag{18.14}$$
- **Category Concept Analysis (CCA)**: For this measure an inverted concept-term matrix is constructed on the full Wikipedia corpus (lemmatized). In contrast to [17], concepts are defined as Wikipedia categories, i.e. we assigned each article to its categories in Wikipedia. For term weighting the *tf.idf* scheme was used. Small articles have been removed using a threshold value for a minimum length of the termvector (more than 400 lemmata). The relatedness computation was performed using the cosine metric, the dice coefficient, and the jaccard similarity coefficient.

18.5 Evaluation

18.5.1 Method

In order to evaluate the quality of a SR measure, a set of pre-classified word pairs is needed. As mentioned above, in previous work on English data, most researchers used the word-pair list by Rubenstein and Goodenough [38] as well as the list by Miller and Charles [30] as an evaluation resource. For German there are — to our knowledge — two research groups, who have compiled lists of word-pairs with respective human judgment: Gurevych et al. constructed three lists (a translation of Rubenstein and Goodenough's list [19], a manually generated set of word pairs, and a semi-automatically generated one [50]).

Cramer and Finthammer (cf. [14], [13]) compiled two lists of word pairs for which they obtained human judgments.[7] We make use of these two lists by Cramer and Finthammer, since they cover a wide range of relatedness types, i.e. systematic and unsystematic relations, and relatedness levels, i.e. various degrees of relation strength. However, they only include nouns, since cross-part-of-speech (cross-POS) relations can be considered to be an additional challenge[8]. In order to better understand the impact (and the potentially included bias) of the construction method of a list, two different methods were applied for the compilation of the word pairs.

For the first list nouns were collected manually from diverse semantic classes, e.g. abstract nouns, such as `Wissen` (engl. *knowledge*), and concrete nouns, such as `Bügeleisen` (engl. *flat-iron*; cf. [14] and [13] for further information). This list of 100 word pairs represents our test set A.

A different method was applied for the second list (set B): firstly, word pairs which are part of collocations were again manually collected, i.e. the two nouns `Rat` and `Tat` (`mit Rat und Tat helfen`; "*to help with words and deeds*") or `Qual` and `Wahl` (`die Qual der Wahl haben`; "*to be spoilt for choice*"). Secondly, word pairs featuring association relations were assembled, i.e. `Afrika` ('*Africa*') and `Tiger` ('*tiger*') or `Weihnachten` ('*christmas*') and `Zimt` ('*cinnamon*'). Thirdly, a list of random word pairs was automatically constructed using the *Wacky* corpus [2] as a resource; *adhoc* constructions were manually excluded. Finally, out of these three resources a set of 500 pairs of words was compiled with no more than 20% of the collocation and association word pairs.

Subjects were asked to rate the word pairs on a 5-level scale (0=*not related* to 4=*strongly related*). The subjects were instructed to base the rating on their intuition about any kind of conceivable relation between the two words. Thus, in contrast to the experimental set-up by Boyd-Graber et al., subjects had no manual, and they were not trained beforehand. Set A was rated by 35 subjects and set B was rated by 75 subjects. For each word pair a human judgment score was calculated by averaging the singular judgments of all subjects. Secondly, the Pearson correlation coefficients were calculated comparing the human scores with each of the measures on the test sets A and B.

[7] Cramer and Finthammer actually worked on 6 separate lists; we merged them into two according to their method of compilation.

[8] Since in most word nets cross-POS relations are very sparse, researchers currently investigate relation types able to connect the noun, verb, and adjective sub-graphs (e.g. [29] or [27]). However, these new relations are not yet integrated on a large scale and therefore should not (or even cannot) be used in SR measures. Furthermore, calculating SR between words with different POS might introduce additional challenges potentially as yet unidentified, which calls for a careful exploration.

18.5.2 Results

Net-Based Measures

The net-based measures were calculated on *GermaNet* v. 5.0 using *GermaNet Pathfinder v. 0.83*[9]. Table 18.3 lists the correlations (Pearson) for test sets A and B, as well as the coverage (percentage of word pairs for which a measure could be calculated) and the average processing time per word pair[10].

Table 18.3 Correlations (*Pearson* coefficient to human estimates), coverage, and processing time per pair of the GermaNet-based measures tested

				WordNet-based measures				
Test set	Leacock & Chodorow	Wu & Palmer	Resnik	Jiang & Conrath	Lin	Hirst & St-Onge	Tree path	Graph path
r Set *A*	0.48	0.36	0.44	0.46	0.48	0.47	0.41	0.42
r Set *B*	0.17	0.21	0.24	0.25	0.27	0.32	0.11	0.31
Coverage	86.9%	86.9%	86.9%	86.9%	86.9%	86.9%	86.9%	86.9%
t/pair (ms)	<10	<10	<10	<10	<10	1110	<10	3649

Distributional Measures

The three web-based (first order) measures obtained their hit counts via the *Google* API; all counts were calculated beforehand and stored in a repository. The LSA word space was calculated using the *Infomap toolkit*[11] v. 0.8.6 on a newspaper corpus (*Süddeutsche Zeitung*) of 145 million words, which had been lemmatized. The co-occurrence matrix (window size: ±75 words) comprised 80,000×3,000 terms and was reduced by SVD to 300 dimensions. For the vector comparisons the cosine measure was applied. Table 18.4 shows the results (correlation, coverage and processing time) for all distributional measures tested.

Wikipedia-Based Measures

The calculation of the Wikipedia measures is based upon the German version of Wikipedia (October 2007). The *Semantic Vector* package[12] utilizes the *Apache Lucene* library. *ESA* and *Graph Path* are implemented in C++ using *Trolltech Qt*. For both *CCA* and *ESA* we had to reduce the matrices on the lemma-dimension for computational reasons, i.e. when building the matrix we excluded those lemmata whose corpus frequency did not exceed a certain threshold (>300). Building the

[9] http://www.hytex.info/030_ergebnisse/030_tools/index_eng_html

[10] The computation was performed on an AMD Athlon XP 2400+, 2.0 GHz and 1GB of RAM.

[11] http://infomap-nlp.sourceforge.net/

[12] http://code.google.com/p/semanticvectors/

Table 18.4 Correlations (*Pearson* coefficent to human estimates), coverage, and processing time per pair of the distributional measures tested

Test set	PMI *Google*	*Google* Quotient	NSD *Google*	LSA (newspaper)
r Set *A*	0.37	0.27	0.37	0.64
r Set *B*	0.34	0.31	0.36	0.63
Coverage	100%	100%	100%	87.0%
t/pair (ms)	<10	<10	<10	<10

NSD measures, we have directly connected to the special page *search* of Wikipedia (`http://de.wikipedia.org/wiki/Spezial:Suche`).

Furthermore, we calculated an LSA word space on Wikipedia, again on an 80,000×3,000-matrix using a window of ±75 terms; however, due to computational limitations we had to use only a subcorpus, by taking the first 800 words of each article (148 mill. tokens in total). Table 18.5 lists the results for all Wikipedia-based measures.

Table 18.5 Correlations (*Pearson* coefficient to human estimates), coverage and processing time per pair of the Wikipedia-based measures tested

Test set	NSD (Wiki)	CCA	Sem. Vec. (Wiki)	ESA	Wiki Graph Path	LSA (Wiki)
r Set *A*	0.69	0.57	0.51	0.52	0.49	0.65
r Set *B*	0.61	0.36	0.28	0.44	0.37	0.57
Coverage	100%	79.8%	99.1%	75.9%	92.0%	83.8%
t/pair (ms)	850	<10	1299	240	2301	<10

Comparing the correlation results shown in Tables 18.3, 18.4, and 18.5, it can be observed that the net-based measures show rather low coefficients ($r = 0.11 - 0.48$); interestingly they score quite similarly within one test set, despite their rather different calculation. For the distributional measures a clear difference can be seen between the three web-based techniques ($0.27 - 0.37$) and the LSA results (scoring up to 0.64); this may either be due to the fact that LSA is a 2^{nd} order approach, being able to establish more paradigmatic relations, or the hit counts, obtained from *Google* are insufficiently precise indicators of co-occurrence. Among the Wikipedia measures the *WikiSearch Distance* scores significantly better than the others (up to 0.69).

A second observation of the results concerns the differences between the correlations of the test sets A and B. Especially the net-based measures, but also most of the Wikipedia-based show significantly worse correlations for set B. Recalling that set B contains a large fraction of random word pairs (80%), a probable explanation is that such measures tend to overestimate relatedness, i.e. they cannot well discriminate between related and unrelated word pairs.

The differences between the approaches tested clearly show how important the influence of the resource is. One conclusion that may be drawn from our results is that a small, hand-crafted and structured resource such as a word net is clearly inferior to a large and semi-structured (Wikipedia) or even completely unstructured resource such as plain text.

With respect to coverage, the web-based measures (including the *WikiSearch Distance* clearly outperform all other approaches. This is not astonishing, given the fact that they operate on the largest vocabulary available. The off-line approaches on the other hand are not as sparse as one might have imagined; the lowest scores are still over 75%, and the net-based as well as the LSA approach achieve a coverage of approximately 87%.

The processing time (per word pair) however differs quite strongly. It is also to be taken with a grain of salt, since it depends strongly on the implementation chosen. Most of the approaches show almost negligible processing times ($<$10 ms), however if complex tree or graph traversals are involved (e.g. *GermaNet* or *Wiki graph path*), the processing time can reach up to several seconds per calculation.

In general, we observed that the distributional measures, especially LSA, perform better than the net-based measures and those using explicit categorial information (ESA, CCA). We therefore conclude that the use of explicit structural information, in the form of semantic links, categories, or of hyperlink graphs, does not establish SR as well as distributional information.

Secondly we could clearly see, that the choice of the resource plays an important role. Interestingly, those measures using the web as a corpus were inferior to distributional or Wikipedia-based measures that operate on a smaller but better controlled training corpora (cf. particularly the important difference between the web-based and the Wikipedia-based NSD). In this context and with respect to corpus choice we can conclude that quality is more important than quantity, an observation which is in line with [23]).

A factor that we disregarded in our study is the influence of context. It is quite obvious that SR is not a static and independent size. On the contrary, it is dynamically interrelated with the current lexical, syntactic, and semantic context, and a proper theory of (or algorithm computing) SR will have to take this into account.

Considering all the results above, it can be stated that the calculation of semantic relatedness is far from being solved in a satisfying manner. Each of the resources that we used certainly captures an important part of lexical meaning; however, it seems that this is not yet sufficient for describing the complex nature of SR between any two terms.

18.5.3 Meta-level Evaluation

Given the statistical spread shown in Table 18.3 to Table 18.5 as well as the obvious discrepancies of the various experimental results exposed by Cramer [13], we argue that the calculation of SR should be considered a continuous problem. We suspect that (no fewer than) the following aspects influence the results of the human judgment experiments and thus the correlation between humans and semantic measures:

- **Research objective:** Seemingly, most studies intend to model the same, i.e. a phenomenon observable in natural language which accounts for lexical cohesion and which is called – depending on the specific research community – semantic similarity/relatedness, association (e.g. [40]), evocation, or semantic distance. However, practically none of these concepts is well defined; there is no consensus among researchers as to which types of relation are to be included and whether these are to be established between words, any kind of lexical unit, concepts etc.
- **Setting of the human judgment experiment:** The studies summarized above differ with respect to the subjects asked to participate, their background and training, as well as the manuals used to explain the task. The different experimental set-ups therefore represent an uncontrolled parameter and might seriously influence the results.
- **Construction of experimental data:** As mentioned above, different methods may be employed in order to construct the experimental data, i.e. randomly selected word-/concept-pairs vs. analytically constructed ones. In addition, data sets might considerably vary with respect to their size, i.e. only a few hand-picked vs. several thousand pairs.
- **Evaluation method:** Finally, the results might be influenced by the specific statistical methods, i.e. different correlation coefficient algorithms, drawn on to determine the correlation between humans and semantic measures, the inter-subject, and the intra-subject correlation[13].

Furthermore, it is – in our opinion – an unsettled issue whether the three types of semantic relations at hand, thus the relations

1. represented in a word net, ontology, corpus etc. (computed via semantic measure),
2. existing between any given word pair in a text (which is mostly relevant for NLP-applications),
3. and the one assigned by subjects in a human judgment experiment

correspond at all. In principle, word nets, ontologies, corpus statistics, and human judgments should represent the (at least partially) shared lexical semantic system encoding the collective knowledge of humans. While the relations between words in a concrete text represent more than just an instantiation of this semantic system.

[13] The inter-subject as well as the intra-subject correlation depends on various parameters, e.g. the complexity of the task, the subjects (and their background, age, etc.) as well as the experimental setup (task definition, training phase, etc.).

That means, the individual comprehension while reading a text might considerably alter the relation strength perceived by the reader between a pair of words. Thus, a (in the sense of the semantic system) moderately related word pair, might be strongly related given a specific context. And consequently, there are at least two aspects which need to be considered in order to model and successfully compute lexical cohesion: firstly, the shared knowledge of the relations in principle, secondly, the concrete relations in a given context. Thus, it is – in our opinion – vital to distinguish between the first step, that is the calculation of SR, for which, as mentioned above, controlled but unstructured resources and distributional (2^{nd} order) methods seem to perform best, and the second, that is the calculation of lexical cohesion, for which additional text-grammatical features also need to be taken into account (q.v. [34]).

18.6 Conclusions and Future Work

We presented a study comparing sixteen different semantic relatedness measures on various lexical resources, which we classified into WordNet-based, distributional and Wikipedia-based SR measures. The algorithms implemented and resources employed were analyzed with respect to practical issues, e.g. run time and coverage. Furthermore, we conducted an extensive evaluation on the basis of a human judgment experiment using Pearson's coefficient to measure correlation. We found that the distributional measures perform best – in terms of coverage and correlation. However, our experiments also show that none of the algorithms, proposed in the literature and implemented for this study, performs outstanding. Taking our experimental results into account, we conclude that the less structure (e.g. semantically typed relations or links) a resource exhibits and the more controlled (e.g. in terms of quality of the documents, language and so forth) it is, the better seem to be the correlation coefficients. In the future, we therefore argue that the research should continue on three different levels: firstly, the concept of semantic relatedness should be stated more precisely, which also means that research results of psycholinguistics, linguistics, cognitive science, and computational linguistics should be integrated. Secondly, on the basis of the thus elaborated concept of semantic relatedness, one or more resources can be determined which best fit the requirements at hand; if more than one resource needs to be considered, a method to combine them needs to be devised in addition. Thirdly, given a substantiated concept of SR and the most suitable resources, an (or a family of) algorithm(s) need to be developed, adapted correspondingly, and evaluated again e.g. on the basis of a human judgment experiment. Hence, we propose the definition of a shared task which might bring the research community interested in SR considerably closer to results of high performance. In addition to this, we plan to experiment with a combination of the above-mentioned relatedness measures; more precisely, we intend to feed the various elements of the measures in Sections 18.4.1 to 18.4.3 into a machine learning toolkit such as Weka (cf. [48]). A pilot study already demonstrated that it is thus possible to enhance the performance by at least 10% compared with the currently best performing measure mentioned in Section 18.5.1.

Acknowledgements. The authors would like to thank Christiane Fellbaum, Lothar Lemnitzer, Alexander Mehler, Sabine Schulte im Walde, Angelika Storrer, Armin Wegner, and Torsten Zesch for their helpful comments. This research was partially funded by the DFG Research Group 437.

References

[1] Anderson, J.R.: A spreading activation theory of memory. Journal of Verbal Leaning and Verbal Behaviour 22, 261–295 (1983)

[2] Baroni, M., Bernardini, S. (eds.): Wacky! Working papers on the web as corpus. GEDIT, Bologna (2006)

[3] Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the Intelligent Scalable Text Summarization Workshop, pp. 10–17 (1997)

[4] Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R.: Adding dense, weighted, connections to wordnet. In: Proceedings of the 3rd Global WordNet Meeting, pp. 29–35 (2006)

[5] Budanitsky, A.: Lexical semantic relatedness and its application in natural language processing. Tech. rep., Department of Computer Science, University of Toronto (1999), http://citeseerx.ist.psu.edu/viewdoc/summary?doi0.1.1.34.1036

[6] Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)

[7] Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods 39(1), 510–526 (2007)

[8] Carthy, J.: Lexical chains versus keywords for topic tracking. In: Computational Linguistics and Intelligent Text Processing. LNCS, pp. 507–510. Springer, Heidelberg (2004)

[9] Cederberg, S., Widdows, D.: Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy. In: Proc. of CoNNL 2003 (2003)

[10] Church, K., Hanks, P.: Word association norms, mutual information and lexicography. In: Proceedings of the 27th ACL, vol. 27, pp. 76–83 (1989)

[11] Cilibrasi, R., Vitanyi, P.M.B.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)

[12] Collins, A., Loftus, E.: A spreading activation theory of semantic processing. Psychological Review 82, 407–428 (1975)

[13] Cramer, I.: How Well Do Semantic Relatedness Measures Perform? A Meta-Study. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing. STEP 2008 Conference Proceedings, Research in Computational Semantics, vol. 1, pp. 59–70. College Publications (2008), http://www.aclweb.org/anthology/W08-2206

[14] Cramer, I., Finthammer, M.: An evaluation procedure for word net based lexical chaining: Methods and issues. In: Proceedings of the 4th Global WordNet Meeting, pp. 120–147 (2008)

[15] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990), http://citeseer.nj.nec.com/deerwester90indexing.html

[16] Fellbaum, C. (ed.): WordNet. An Electronic Lexical Database. MIT Press, Cambridge (1998)

[17] Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 6–12 (2007)

[18] Green, S.J.: Building hypertext links by computing semantic similarity. IEEE Transactions on Knowledge and Data Engineering 11(5) (1999)

[19] Gurevych, I.: Using the structure of a conceptual network in computing semantic relatedness. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 767–778. Springer, Heidelberg (2005)

[20] Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)

[21] Hirst, G., St-Onge, D.: Lexical chains as representation of context for the detection and correction malapropisms. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press, Cambridge (1998)

[22] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of ROCLING X, pp. 19–33 (1997)

[23] Kilgarriff, A.: Googleology is bad science. Computational Linguistics 33(1), 147–151 (2007)

[24] Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review 104(1), 211–240 (1997)

[25] Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–284. The MIT Press, Cambridge (1998)

[26] Lemnitzer, L., Kunze, C.: Germanet – representation, visualization, application. In: Proceedings of the 4th Language Resources and Evaluation Conference, pp. 1485–1491 (2002)

[27] Lemnitzer, L., Wunsch, H., Gupta, P.: Enriching germanet with verb-noun relations – a case study of lexical acquisition. In: Proceedings of the 6th International Language Resources and Evaluation (2008)

[28] Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304 (1998)

[29] Marrafa, P., Mendes, S.: Modeling adjectives in computational relational lexica. In: Proceedings of the COLING/ACL 2006, pp. 555–562 (2006) (poster session)

[30] Miller, G.A., Charles, W.G.: Contextual correlates of semantic similiarity. Language and Cognitive Processes 6(1), 1–28 (1991)

[31] Milne, D.: Computing semantic relatedness using wikipedia link structure. In: Proc. of NZCSRSC 2007 (2007)

[32] Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational linguistics 17(1) (1991)

[33] Morris, J., Hirst, G.: Non-classical lexical semantic relations. In: Proc. of HLT-NAACL Workshop on Computational Lexical Semantics (2004)

[34] Morris, J., Hirst, G.: The subjectivity of lexical cohesion in text. In: Chanahan, J.C., Qu, C., Wiebe, J. (eds.) Computing attitude and affect in text. Springer, Heidelberg (2005)

[35] Novischi, A., Moldovan, D.: Question answering with lexical chains propagating verb arguments. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 897–904 (2006)

[36] Rapp, R.: The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In: Proceedings of COLING 2002, Taipei, Taiwan (2002)

[37] Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Martin, T., L. Ralescu, A. (eds.) IJCAI-WS 1995. LNCS, vol. 1188, Springer, Heidelberg (1997)

[38] Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Communications of the ACM 8(10), 627–633 (1965)

[39] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw Hill, New York (1983)

[40] Schulte im Walde, S., Melinger, A.: Identifying semantic relations and functional properties of human verb associations. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 612–619 (2005)

[41] Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–124 (1998)

[42] Silber, G.H., McCoy, K.F.: Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics 28(4) (2002)

[43] Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: Proceedings of the 21st national conference on Artificial intelligence, vol. 2, pp. 1419–1424. AAAI Press, Menlo Park (2006)

[44] Teich, E., Fankhauser, P.: Wordnet for lexical cohesion analysis. In: Proc. of the 2nd Global WordNet Conference, GWC 2004 (2004)

[45] Turney, P.D.: Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: Proceedings of the 12th European Conference on Machine Learning EMCL 2001, pp. 491–502. Springer, London (2001), http://portal.acm.org/citation.cfm?id=645328.650004

[46] Wandmacher, T.: How semantic is Latent Semantic Analysis? In: Proceedings of TALN/RECITAL 2005, Dourdan, France (2005)

[47] Widdows, D., Ferraro, K.: Semantic vectors: a scalable open source package and online technology management application. In: Elra, E. (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

[48] Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

[49] Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133–138 (1994)

[50] Zesch, T., Gurevych, I.: Automatically creating datasets for measures of semantic relatedness. In: Proceedings of the Workshop on Linguistic Distances at COLING/ACL 2006, pp. 16–24 (2006)

[51] Zesch, T., Gurevych, I., Mühlhäuser, M.: Comparing wikipedia and german wordnet by evaluating semantic relatedness on multiple datasets. In: Proc. of NAACL-HLT (2007)

[52] Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from wikipedia and wiktionary. In: Proceedings of the Conference on Language Resources and Evaluation (LREC). Electronic Proceedings (2008)

Author Index

Bärenfänger, Maja 35
Beißwenger, Michael 347
Bieler, Heike 15
Boyd-Graber, Jordan 81

Chamberlain, Jon 175
Cramer, Irene 377

Denoyer, Ludovic 249
Diewald, Nils 175

Fellbaum, Christiane 81

Gallinari, Patrick 249
Geibel, Peter 267
Goecke, Daniela 175, 197

Heyer, Gerhard 333
Hilbert, Mirco 35
Hirst, Graeme 61
Huitfeldt, Claus 157

Jettka, Daniel 175

Kracht, Marcus 111
Konya, Iuliu 221
Kruschwitz, Udo 175
Kühnberger, Kai-Uwe 1, 129, 267

Lobin, Henning 1, 35
Lüngen, Harald 1, 35, 347

Maes, Francis 249
Mehler, Alexander 1, 267, 299
Metzing, Dieter 197
Michaelis, Jens 129
Mohammad, Saif 61
Mönnich, Uwe 129

Nikolova, Sonya 81

Oltramari, Alessandro 97
Ovchinnikova, Ekaterina 129

Paaß, Gerhard 221
Poesio, Massimo 175

Selzam, Bianca 347
Sperberg-McQueen, C.M. 157
Stede, Manfred 15
Storrer, Angelika 1, 347
Stührenberg, Maik 175, 197

Waltinger, Ulli 299, 377
Wandmacher, Tonio 129, 377
Witt, Andreas 1, 197