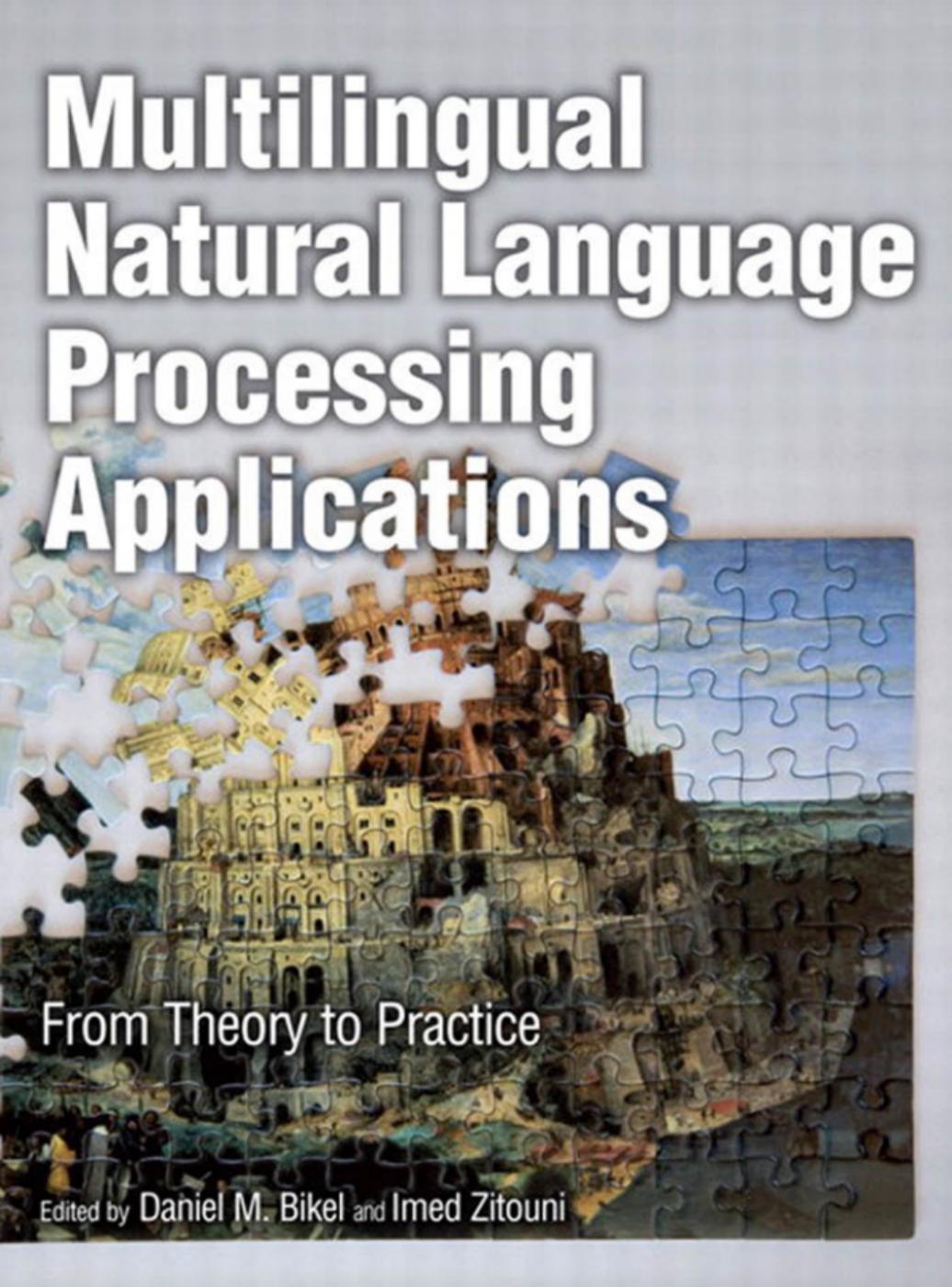


Multilingual Natural Language Processing Applications



From Theory to Practice

Edited by Daniel M. Bikel and Imed Zitouni

Register Your Book

at ibmpressbooks.com/ibmregister

Upon registration, we will send you electronic sample chapters from two of our popular IBM Press books. In addition, you will be automatically entered into a monthly drawing for a free IBM Press book.

Registration also entitles you to:

- Notices and reminders about author appearances, conferences, and online chats with special guests
- Access to supplemental material that may be available
- Advance notice of forthcoming editions
- Related book recommendations
- Information about special contests and promotions throughout the year
- Chapter excerpts and supplements of forthcoming books

Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Director, IBM Press
c/o Pearson Education
800 East 96th Street
Indianapolis, IN 46240

e-mail: IBMPress@pearsoned.com

Visit us on the Web: ibmpressbooks.com

Related Books of Interest



The IBM Style Guide

Conventions for Writers and Editors

by Francis DeRespinis, Peter Hayward, Jana Jenkins, Amy Laird, Leslie McDonald, Eric Radzinski

ISBN: 0-13-210130-0

The IBM Style Guide distills IBM wisdom for developing superior content: information that is consistent, clear, concise, and easy to translate. This expert guide contains practical guidance on topic-based writing, writing content for different media types, and writing for global audiences and can help any organization improve and standardize content across authors, delivery mechanisms, and geographic locations.

The IBM Style Guide can help any organization or individual create and manage content more effectively. The guidelines are especially valuable for businesses that have not previously adopted a corporate style guide, for anyone who writes or edits for IBM as an employee or outside contractor, and for anyone who uses modern approaches to information architecture.



DITA Best Practices

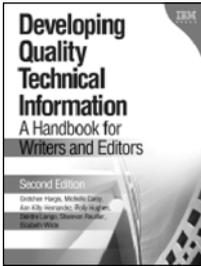
By Laura Bellamy, Michelle Carey, and Jenifer Schlotfeldt

ISBN: 0-13-248052-2

Darwin Information Typing Architecture (DITA) is today's most powerful toolbox for constructing information. By implementing DITA, organizations can gain more value from their technical documentation than ever before. In *DITA Best Practices*, three DITA pioneers offer the first complete roadmap for successful DITA adoption, implementation, and usage. Drawing on years of experience helping large organizations adopt DITA, the authors answer crucial questions the "official" DITA documents ignore. An indispensable resource for every writer, editor, information architect, manager, or consultant involved with evaluating, deploying, or using DITA.

Sign up for the monthly IBM Press newsletter at ibmpressbooks/newsletters

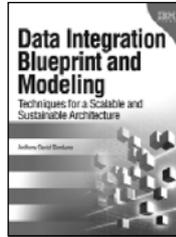
Related Books of Interest



Developing Quality Technical Information, Second Edition

By Gretchen Hargis, Michelle Carey, Ann Kilty Hernandez, Polly Hughes, Deirdre Longo, Shannon Rouiller, and Elizabeth Wilde
ISBN: 0-13-147749-8

Direct from IBM's own documentation experts, this is the definitive guide to developing outstanding technical documentation—for the Web and for print. Using extensive before-and-after examples, illustrations, and checklists, the authors show exactly how to create documentation that's easy to find, understand, and use. This edition includes extensive new coverage of topic-based information, simplifying search and retrievability, internationalization, visual effectiveness, and much more.



Data Integration Blueprint and Modeling Techniques for a Scalable and Sustainable Architecture

By Anthony David Giordano
ISBN: 0-13-708493-5

Making Data Integration Work: How to Systematically Reduce Cost, Improve Quality, and Enhance Effectiveness

This book presents the solution: a clear, consistent approach to defining, designing, and building data integration components to reduce cost, simplify management, enhance quality, and improve effectiveness. Leading IBM data management expert Tony Giordano brings together best practices for architecture, design, and methodology and shows how to do the disciplined work of getting data integration right.

Mr. Giordano begins with an overview of the “patterns” of data integration, showing how to build blueprints that smoothly handle both operational and analytic data integration. Next, he walks through the entire project lifecycle, explaining each phase, activity, task, and deliverable through a complete case study. Finally, he shows how to integrate data integration with other information management disciplines, from data governance to metadata. The book's appendices bring together key principles, detailed models, and a complete data integration glossary.

IBM
Press™

Visit ibmpressbooks.com
for all product information

Related Books of Interest



Search Engine Marketing, Inc.

By Mike Moran and Bill Hunt
ISBN: 0-13-606868-5

The #1 Step-by-Step Guide to Search Marketing Success...Now Completely Updated with New Techniques, Tools, Best Practices, and Value-Packed Bonus DVD!

In this book, two world-class experts present today's best practices, step-by-step techniques, and hard-won tips for using search engine marketing to achieve your sales and marketing goals, whatever they are. Mike Moran and Bill Hunt thoroughly cover both the business and technical aspects of contemporary search engine marketing, walking beginners through all the basics while providing reliable, up-to-the-minute insights for experienced professionals.

Thoroughly updated to fully reflect today's latest search engine marketing opportunities, this book guides you through profiting from social media marketing, site search, advanced keyword tools, hybrid paid search auctions, and much more.



Listen to the author's podcast at:
ibmpressbooks.com/podcasts



Do It Wrong Quickly

How the Web Changes the Old Marketing Rules

Moran
ISBN: 0-13-225596-0



Get Bold

Using Social Media to Create a New Type of Social Business

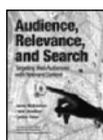
Carter
ISBN: 0-13-261831-1



The Social Factor

Innovate, Ignite, and Win through Mass Collaboration and Social Networking

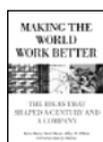
Azua
ISBN: 0-13-701890-8



Audience, Relevance, and Search

Targeting Web Audiences with Relevant Content

Mathewson, Donatone, Fishel
ISBN: 0-13-700420-6



Making the World Work Better

The Ideas That Shaped a Century and a Company

Maney, Hamm, O'Brien
ISBN: 0-13-275510-6

Sign up for the monthly IBM Press newsletter at
ibmpressbooks/newsletters

Multilingual Natural Language Processing Applications

This page intentionally left blank

Multilingual Natural Language Processing Applications

From Theory to Practice

Edited by **Daniel M. Bikel** **Imed Zitouni**

IBM Press

Pearson plc

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

ibmpressbooks.com

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2012 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer
Cover design: IBM Corporation

Executive Editor: Bernard Goodwin
Marketing Manager: Stephane Nakib
Publicist: Heather Fox
Managing Editor: John Fuller
Designer: Alan Clements
Project Editor: Elizabeth Ryan
Copy Editor: Carol Lallier
Indexer: Jack Lewis
Compositor: LaurelTech
Proofreader: Kelli M. Brooks
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact
International Sales
international@pearson.com

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, the IBM press logo, IBM Watson, ThinkPlace, WebSphere, and InfoSphere. A current list of IBM trademarks is available on the web at “copyright and trademark information” as www.ibm.com/legal/copytrade.shtml. Microsoft, Windows, Windows NT, and the Windows logo are trademarks of the Microsoft Corporation in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

Library of Congress Cataloging-in-Publication Data is on file with the Library of Congress.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

ISBN-13: 978-0-13-715144-8

ISBN-10: 0-13-715144-6

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.
First printing, May 2012

*I dedicate this book to
my mother Rita, my brother Robert, my sister-in-law Judi,
my nephew Wolfie, and my niece Freya—Bikels all.
I also dedicate it to Science.*

DMB

*I dedicate this book to
my parents Ali and Radhia, who taught me the love of science,
my wife Barbara, for her support and encouragement,
my kids Nassim and Ines, for the joy they give me.
I also dedicate it to my grandmother Zohra,
my brother Issam, my sister-in-law Chahnez,
as well as my parents-in-law Alain and Pilar.*

IZ

Contents

Preface	xxi
Acknowledgments	xxv
About the Authors	xxvii
Part I In Theory	1
Chapter 1 Finding the Structure of Words	3
1.1 Words and Their Components	4
1.1.1 Tokens	4
1.1.2 Lexemes	5
1.1.3 Morphemes	5
1.1.4 Typology	7
1.2 Issues and Challenges	8
1.2.1 Irregularity	8
1.2.2 Ambiguity	10
1.2.3 Productivity	13
1.3 Morphological Models	15
1.3.1 Dictionary Lookup	15
1.3.2 Finite-State Morphology	16
1.3.3 Unification-Based Morphology	18
1.3.4 Functional Morphology	19
1.3.5 Morphology Induction	21
1.4 Summary	22
Chapter 2 Finding the Structure of Documents	29
2.1 Introduction	29
2.1.1 Sentence Boundary Detection	30
2.1.2 Topic Boundary Detection	32
2.2 Methods	33
2.2.1 Generative Sequence Classification Methods	34
2.2.2 Discriminative Local Classification Methods	36

2.2.3	Discriminative Sequence Classification Methods	38
2.2.4	Hybrid Approaches	39
2.2.5	Extensions for Global Modeling for Sentence Segmentation	40
2.3	Complexity of the Approaches	40
2.4	Performances of the Approaches	41
2.5	Features	41
2.5.1	Features for Both Text and Speech	42
2.5.2	Features Only for Text	44
2.5.3	Features for Speech	45
2.6	Processing Stages	48
2.7	Discussion	48
2.8	Summary	49
Chapter 3	Syntax	57
3.1	Parsing Natural Language	57
3.2	Treebanks: A Data-Driven Approach to Syntax	59
3.3	Representation of Syntactic Structure	63
3.3.1	Syntax Analysis Using Dependency Graphs	63
3.3.2	Syntax Analysis Using Phrase Structure Trees	67
3.4	Parsing Algorithms	70
3.4.1	Shift-Reduce Parsing	72
3.4.2	Hypergraphs and Chart Parsing	74
3.4.3	Minimum Spanning Trees and Dependency Parsing	79
3.5	Models for Ambiguity Resolution in Parsing	80
3.5.1	Probabilistic Context-Free Grammars	80
3.5.2	Generative Models for Parsing	83
3.5.3	Discriminative Models for Parsing	84
3.6	Multilingual Issues: What Is a Token?	87
3.6.1	Tokenization, Case, and Encoding	87
3.6.2	Word Segmentation	89
3.6.3	Morphology	90
3.7	Summary	92
Chapter 4	Semantic Parsing	97
4.1	Introduction	97
4.2	Semantic Interpretation	98
4.2.1	Structural Ambiguity	99
4.2.2	Word Sense	99
4.2.3	Entity and Event Resolution	100
4.2.4	Predicate-Argument Structure	100
4.2.5	Meaning Representation	101
4.3	System Paradigms	101
4.4	Word Sense	102
4.4.1	Resources	104

4.4.2	Systems	105
4.4.3	Software	116
4.5	Predicate-Argument Structure	118
4.5.1	Resources	118
4.5.2	Systems	122
4.5.3	Software	147
4.6	Meaning Representation	147
4.6.1	Resources	148
4.6.2	Systems	149
4.6.3	Software	151
4.7	Summary	152
4.7.1	Word Sense Disambiguation	152
4.7.2	Predicate-Argument Structure	153
4.7.3	Meaning Representation	153
Chapter 5 Language Modeling		169
5.1	Introduction	169
5.2	n -Gram Models	170
5.3	Language Model Evaluation	170
5.4	Parameter Estimation	171
5.4.1	Maximum-Likelihood Estimation and Smoothing	171
5.4.2	Bayesian Parameter Estimation	173
5.4.3	Large-Scale Language Models	174
5.5	Language Model Adaptation	176
5.6	Types of Language Models	178
5.6.1	Class-Based Language Models	178
5.6.2	Variable-Length Language Models	179
5.6.3	Discriminative Language Models	179
5.6.4	Syntax-Based Language Models	180
5.6.5	MaxEnt Language Models	181
5.6.6	Factored Language Models	183
5.6.7	Other Tree-Based Language Models	185
5.6.8	Bayesian Topic-Based Language Models	186
5.6.9	Neural Network Language Models	187
5.7	Language-Specific Modeling Problems	188
5.7.1	Language Modeling for Morphologically Rich Languages	189
5.7.2	Selection of Subword Units	191
5.7.3	Modeling with Morphological Categories	192
5.7.4	Languages without Word Segmentation	193
5.7.5	Spoken versus Written Languages	194
5.8	Multilingual and Crosslingual Language Modeling	195
5.8.1	Multilingual Language Modeling	195
5.8.2	Crosslingual Language Modeling	196
5.9	Summary	198

Chapter 6 Recognizing Textual Entailment	209
6.1 Introduction	209
6.2 The Recognizing Textual Entailment Task	210
6.2.1 Problem Definition	210
6.2.2 The Challenge of RTE	212
6.2.3 Evaluating Textual Entailment System Performance	213
6.2.4 Applications of Textual Entailment Solutions	214
6.2.5 RTE in Other Languages	218
6.3 A Framework for Recognizing Textual Entailment	219
6.3.1 Requirements	219
6.3.2 Analysis	220
6.3.3 Useful Components	220
6.3.4 A General Model	224
6.3.5 Implementation	227
6.3.6 Alignment	233
6.3.7 Inference	236
6.3.8 Training	238
6.4 Case Studies	238
6.4.1 Extracting Discourse Commitments	239
6.4.2 Edit Distance-Based RTE	240
6.4.3 Transformation-Based Approaches	241
6.4.4 Logical Representation and Inference	242
6.4.5 Learning Alignment Independently of Entailment	244
6.4.6 Leveraging Multiple Alignments for RTE	245
6.4.7 Natural Logic	245
6.4.8 Syntactic Tree Kernels	246
6.4.9 Global Similarity Using Limited Dependency Context	247
6.4.10 Latent Alignment Inference for RTE	247
6.5 Taking RTE Further	248
6.5.1 Improve Analytics	248
6.5.2 Invent/Tackle New Problems	249
6.5.3 Develop Knowledge Resources	249
6.5.4 Better RTE Evaluation	251
6.6 Useful Resources	252
6.6.1 Publications	252
6.6.2 Knowledge Resources	252
6.6.3 Natural Language Processing Packages	253
6.7 Summary	253
Chapter 7 Multilingual Sentiment and Subjectivity Analysis	259
7.1 Introduction	259
7.2 Definitions	260
7.3 Sentiment and Subjectivity Analysis on English	262
7.3.1 Lexicons	262

7.3.2	Corpora	262
7.3.3	Tools	263
7.4	Word- and Phrase-Level Annotations	264
7.4.1	Dictionary-Based	264
7.4.2	Corpus-Based	267
7.5	Sentence-Level Annotations	270
7.5.1	Dictionary-Based	270
7.5.2	Corpus-Based	271
7.6	Document-Level Annotations	272
7.6.1	Dictionary-Based	272
7.6.2	Corpus-Based	274
7.7	What Works, What Doesn't	274
7.7.1	Best Scenario: Manually Annotated Corpora	274
7.7.2	Second Best: Corpus-Based Cross-Lingual Projections	275
7.7.3	Third Best: Bootstrapping a Lexicon	275
7.7.4	Fourth Best: Translating a Lexicon	276
7.7.5	Comparing the Alternatives	276
7.8	Summary	277

Part II In Practice **283**

Chapter 8	Entity Detection and Tracking	285
8.1	Introduction	285
8.2	Mention Detection	287
8.2.1	Data-Driven Classification	287
8.2.2	Search for Mentions	289
8.2.3	Mention Detection Features	291
8.2.4	Mention Detection Experiments	294
8.3	Coreference Resolution	296
8.3.1	The Construction of Bell Tree	297
8.3.2	Coreference Models: Linking and Starting Model	298
8.3.3	A Maximum Entropy Linking Model	300
8.3.4	Coreference Resolution Experiments	302
8.4	Summary	303
Chapter 9	Relations and Events	309
9.1	Introduction	309
9.2	Relations and Events	310
9.3	Types of Relations	311
9.4	Relation Extraction as Classification	312
9.4.1	Algorithm	312
9.4.2	Features	313
9.4.3	Classifiers	316

9.5	Other Approaches to Relation Extraction	317
9.5.1	Unsupervised and Semisupervised Approaches	317
9.5.2	Kernel Methods	319
9.5.3	Joint Entity and Relation Detection	320
9.6	Events	320
9.7	Event Extraction Approaches	320
9.8	Moving Beyond the Sentence	323
9.9	Event Matching	323
9.10	Future Directions for Event Extraction	326
9.11	Summary	326
Chapter 10	Machine Translation	331
10.1	Machine Translation Today	331
10.2	Machine Translation Evaluation	332
10.2.1	Human Assessment	332
10.2.2	Automatic Evaluation Metrics	334
10.2.3	WER, BLEU, METEOR, ...	335
10.3	Word Alignment	337
10.3.1	Co-occurrence	337
10.3.2	IBM Model 1	338
10.3.3	Expectation Maximization	339
10.3.4	Alignment Model	340
10.3.5	Symmetrization	340
10.3.6	Word Alignment as Machine Learning Problem	341
10.4	Phrase-Based Models	343
10.4.1	Model	343
10.4.2	Training	344
10.4.3	Decoding	345
10.4.4	Cube Pruning	347
10.4.5	Log-Linear Models and Parameter Tuning	348
10.4.6	Coping with Model Size	349
10.5	Tree-Based Models	350
10.5.1	Hierarchical Phrase-Based Models	350
10.5.2	Chart Decoding	351
10.5.3	Syntactic Models	352
10.6	Linguistic Challenges	354
10.6.1	Lexical Choice	354
10.6.2	Morphology	355
10.6.3	Word Order	356
10.7	Tools and Data Resources	356
10.7.1	Basic Tools	357
10.7.2	Machine Translation Systems	357
10.7.3	Parallel Corpora	358

10.8 Future Directions	358
10.9 Summary	359
Chapter 11 Multilingual Information Retrieval	365
11.1 Introduction	366
11.2 Document Preprocessing	366
11.2.1 Document Syntax and Encoding	367
11.2.2 Tokenization	369
11.2.3 Normalization	370
11.2.4 Best Practices for Preprocessing	371
11.3 Monolingual Information Retrieval	372
11.3.1 Document Representation	372
11.3.2 Index Structures	373
11.3.3 Retrieval Models	374
11.3.4 Query Expansion	376
11.3.5 Document A Priori Models	377
11.3.6 Best Practices for Model Selection	377
11.4 CLIR	378
11.4.1 Translation-Based Approaches	378
11.4.2 Machine Translation	380
11.4.3 Interlingual Document Representations	381
11.4.4 Best Practices	382
11.5 MLIR	382
11.5.1 Language Identification	383
11.5.2 Index Construction for MLIR	383
11.5.3 Query Translation	384
11.5.4 Aggregation Models	385
11.5.5 Best Practices	385
11.6 Evaluation in Information Retrieval	386
11.6.1 Experimental Setup	387
11.6.2 Relevance Assessments	387
11.6.3 Evaluation Measures	388
11.6.4 Established Data Sets	389
11.6.5 Best Practices	391
11.7 Tools, Software, and Resources	391
11.8 Summary	393
Chapter 12 Multilingual Automatic Summarization	397
12.1 Introduction	397
12.2 Approaches to Summarization	399
12.2.1 The Classics	399
12.2.2 Graph-Based Approaches	401
12.2.3 Learning How to Summarize	406
12.2.4 Multilingual Summarization	409

12.3	Evaluation	412
12.3.1	Manual Evaluation Methodologies	413
12.3.2	Automated Evaluation Methods	415
12.3.3	Recent Development in Evaluating Summarization Systems	418
12.3.4	Automatic Metrics for Multilingual Summarization	419
12.4	How to Build a Summarizer	420
12.4.1	Ingredients	422
12.4.2	Devices	423
12.4.3	Instructions	423
12.5	Competitions and Datasets	424
12.5.1	Competitions	424
12.5.2	Data Sets	425
12.6	Summary	426
Chapter 13	Question Answering	433
13.1	Introduction and History	433
13.2	Architectures	435
13.3	Source Acquisition and Preprocessing	437
13.4	Question Analysis	440
13.5	Search and Candidate Extraction	443
13.5.1	Search over Unstructured Sources	443
13.5.2	Candidate Extraction from Unstructured Sources	445
13.5.3	Candidate Extraction from Structured Sources	449
13.6	Answer Scoring	450
13.6.1	Overview of Approaches	450
13.6.2	Combining Evidence	452
13.6.3	Extension to List Questions	453
13.7	Crosslingual Question Answering	454
13.8	A Case Study	455
13.9	Evaluation	460
13.9.1	Evaluation Tasks	460
13.9.2	Judging Answer Correctness	461
13.9.3	Performance Metrics	462
13.10	Current and Future Challenges	464
13.11	Summary and Further Reading	465
Chapter 14	Distillation	475
14.1	Introduction	475
14.2	An Example	476
14.3	Relevance and Redundancy	477
14.4	The Rosetta Consortium Distillation System	479
14.4.1	Document and Corpus Preparation	480
14.4.2	Indexing	483
14.4.3	Query Answering	483

14.5 Other Distillation Approaches	488
14.5.1 System Architectures	488
14.5.2 Relevance	488
14.5.3 Redundancy	489
14.5.4 Multimodal Distillation	490
14.5.5 Crosslingual Distillation	490
14.6 Evaluation and Metrics	491
14.6.1 Evaluation Metrics in the GALE Program	492
14.7 Summary	495
Chapter 15 Spoken Dialog Systems	499
15.1 Introduction	499
15.2 Spoken Dialog Systems	499
15.2.1 Speech Recognition and Understanding	500
15.2.2 Speech Generation	503
15.2.3 Dialog Manager	504
15.2.4 Voice User Interface	505
15.3 Forms of Dialog	509
15.4 Natural Language Call Routing	510
15.5 Three Generations of Dialog Applications	510
15.6 Continuous Improvement Cycle	512
15.7 Transcription and Annotation of Utterances	513
15.8 Localization of Spoken Dialog Systems	513
15.8.1 Call-Flow Localization	514
15.8.2 Prompt Localization	514
15.8.3 Localization of Grammars	516
15.8.4 The Source Data	516
15.8.5 Training	517
15.8.6 Test	519
15.9 Summary	520
Chapter 16 Combining Natural Language Processing Engines	523
16.1 Introduction	523
16.2 Desired Attributes of Architectures for Aggregating Speech and NLP Engines	524
16.2.1 Flexible, Distributed Componentization	524
16.2.2 Computational Efficiency	525
16.2.3 Data-Manipulation Capabilities	526
16.2.4 Robust Processing	526
16.3 Architectures for Aggregation	527
16.3.1 UIMA	527
16.3.2 GATE: General Architecture for Text Engineering	529
16.3.3 InfoSphere Streams	530

16.4 Case Studies	531
16.4.1 The GALE Interoperability Demo System	531
16.4.2 Translingual Automated Language Exploitation System (TALES)	538
16.4.3 Real-Time Translation Services (RTTS)	538
16.5 Lessons Learned	540
16.5.1 Segmentation Involves a Trade-off between Latency and Accuracy	540
16.5.2 Joint Optimization versus Interoperability	540
16.5.3 Data Models Need Usage Conventions	540
16.5.4 Challenges of Performance Evaluation	541
16.5.5 Ripple-Forward Training of Engines	541
16.6 Summary	542
16.7 Sample UIMA Code	542
Index	551

Preface

Almost everyone on the planet, it seems, has been touched in some way by advances in information technology and the proliferation of the Internet. Recently, multimedia information sources have become increasingly popular. Nevertheless, the sheer volume of raw natural language text keeps increasing, and this text is being generated in all the major languages on Earth. For example, the English Wikipedia reports that 101 language-specific Wikipedias exist with at least 10,000 articles each. There is therefore a pressing need for countries, companies, and individuals to analyze this massive amount of text, translate it, and synthesize and distill it.

Previously, to build robust and accurate multilingual natural language processing (NLP) applications, a researcher or developer had to consult several reference books and dozens, if not hundreds, of journal and conference papers. Our aim for this book is to provide a “one-stop shop” that offers all the requisite background and practical advice for building such applications. Although it is quite a tall order, we hope that, at a minimum, you find this book a useful resource.

In the last two decades, NLP researchers have developed exciting algorithms for processing large amounts of text in many different languages. By far, the dominant approach has been to build a statistical model that can learn from examples. In this way, a model can be robust to changes in the type of text and even the language of text on which it operates. With the right design choices, the same model can be trained to work in a new domain or new language simply by providing new examples in that domain. This approach also obviates the need for researchers to lay out, in a painstaking fashion, all the rules that govern the problem at hand and the manner in which those rules must be combined. Rather, a statistical system typically allows for researchers to provide an abstract expression of possible *features* of the input, where the relative importance of those features can be learned during the *training* phase and can be applied to new text during the *decoding*, or *inference*, phase.

The field of statistical NLP is rapidly changing. Part of the change is due to the field’s growth. For example, one of the main conferences in the field is that of the Association of Computational Linguistics, where conference attendance has doubled in the last five years. Also, the share of NLP papers in the IEEE speech and language processing conferences and journals more than doubled in the last decade; IEEE constitutes one of the world’s largest professional associations for the advancement of technology. Not only are NLP researchers making inherent progress on the various subproblems of the field, but NLP continues to benefit (and borrow) heavily from progress in the machine learning community and linguistics alike. This book devotes some attention to cutting-edge algorithms and techniques, but its primary purpose is to be a thorough explication of best practices in the field. Furthermore, every chapter describes how the techniques discussed apply in a *multilingual* setting.

This book is divided into two parts. Part I, In Theory, includes the first seven chapters and lays out the various core NLP problems and algorithms to attack those problems. The

first three chapters focus on finding structure in language at various levels of granularity. Chapter 1 introduces the important concept of *morphology*, the study of the structure of words, and ways to process the diverse array of morphologies present in the world's languages. Chapter 2 discusses the methods by which documents may be decomposed into more manageable parts, such as sentences and larger units related by topic. Finally, in this initial trio of chapters, Chapter 3 investigates the various methods of uncovering a sentence's internal structure, or *syntax*. Syntax has long been a dominant area of research in linguistics, and that dominance has been mirrored in the field of NLP as well. The dominance, in part, stems from the fact that the structure of a sentence bears relation to the sentence's meaning, so uncovering syntactic structure can serve as a first step toward a full "understanding" of a sentence.

Finding a structured meaning representation for a sentence, or for some other unit of text, is often called *semantic parsing*, which is the concern of Chapter 4. That chapter covers, inter alia, a related subproblem that has garnered much attention in recent years known as *semantic role labeling*, which attempts to find the syntactic phrases that constitute the *arguments* to some verb or predicate. By identifying and classifying a verb's arguments, we come one step closer to producing a *logical form* for a sentence, which is one way to represent a sentence's meaning in such a way as to be readily processed by machine, using the rich array of tools available from logic that mankind has been developing since ancient times.

But what if we do not want or need the deep syntactico-semantic structure that semantic parsing would provide? What if our problem is simply to decide which among many candidate sentences is the most likely sentence a human would write or speak? One way to do so would be to develop a model that could score each sentence according to its grammaticality and pick the sentence with the highest score. The problem of producing a score or probability estimate for a sequence of word tokens is known as *language modeling* and is the subject of Chapter 5.

Representing meaning and judging a sentence's grammaticality are only two of many possible first steps toward processing language. Moving further toward some sense of understanding, we might wish to have an algorithm make *inferences* about facts expressed in a piece of text. For example, we might want to know if a fact mentioned in one sentence is *entailed* by some previous sentence in a document. This sort of inference is known as *recognizing textual entailment* and is the subject of Chapter 6.

Finding which facts or statements are entailed by others is clearly important to the automatic understanding of text, but there is also the *nature* of those statements. Understanding which statements are subjective and the polarity of the opinion expressed is the subject matter of Chapter 7. Given how often people express opinions, this is clearly an important problem area, all the more so in an age when social networks are fast becoming the dominant form of person-to-person communication on the Internet. This chapter rounds out Part I of our book.

Part II, In Practice, takes the various core areas of NLP described in Part I and explains how to apply them to the diverse array of real-world NLP applications. Engineering is often about trade-offs, say, between time and space, and so the chapters in this applied part of our book explore the trade-offs in making various algorithmic and design choices when building a robust, multilingual NLP application.

Chapter 8 describes ways to identify and classify *named entities* and other mentions of those entities in text, as well as methods to identify when two or more entity mentions *corefer*. These two problems are typically known as *mention detection* and *coreference resolution*; they are two of the core parts of a larger application area known as *information extraction*.

Chapter 9 continues the information extraction discussion, exploring techniques for finding out how two entities are related to each other, known as *relation extraction*, and identifying and classifying events, or *event extraction*. An event, in this case, is when something happens involving multiple entities, and we would like a machine to uncover who the participants are and what their roles are. In this way, event extraction is closely related to the core NLP problem of semantic role labeling.

Chapter 10 describes one of the oldest problems in the field, and one of the few that is an inherently multilingual NLP problem: *machine translation*, or *MT*. Automatically translating from one language to another has long been a holy grail of NLP research, and in recent years the community has developed techniques and can obtain hardware that make MT a practical reality, reaping rewards after decades of effort.

It is one thing to translate text, but how do we make sense of all the text out there in seemingly limitless quantity? Chapters 8 and 9 make some headway in this regard by helping us automatically produce structured records of information in text. Another way to tackle the quantity problem is to narrow down the scope by finding the few documents, or subparts of documents, that are relevant based on a search query. This problem is known as *information retrieval* and is the subject of Chapter 11. In many ways, commercial search engines such as Google are large-scale information retrieval systems. Given the popularity of search engines, this is clearly an important NLP problem—all the more so given the number of corpora that are *not* public and therefore searchable by commercial engines.

Another way we might tackle the sheer quantity of text is by automatically summarizing it, which is the topic of Chapter 12. This very difficult problem involves either finding the sentences, or bits of sentences, that contribute to providing a relevant summary of a larger quantity of text or else ingesting the text summarizing its meaning in some internal representation, and then *generating* the text that constitutes a summary, much as a human might do.

Often, humans would like machines to process text automatically because they have questions they seek to answer. These questions can range from simple, factoid-like questions, such as “When was John F. Kennedy born?” to more complex questions such as “What is the largest city in Bavaria, Germany?” Chapter 13 discusses ways to build systems to answer these types of questions automatically.

What if the types of questions we might like to answer are even *more* complex? Our queries might have multiple answers, such as “Name all the foreign heads of state President Barack Obama met with in 2010.” These types of queries are handled by a relatively new subdiscipline within NLP known as *distillation*. In a very real way, distillation combines the techniques of information retrieval with information extraction and adds a few of its own.

In many cases, we might like to have machines process language in an interactive way, making use of speech technology that both recognizes and synthesizes speech. Such systems are known as *dialog systems* and are covered in Chapter 15. Due to advances in speech

recognition, dialog management, and speech synthesis, such systems are becoming increasingly practical and are seeing widespread, real-world deployment.

Finally, we, as NLP researchers and engineers, might like to build systems using diverse arrays of components developed across the world. This aggregation of processing engines is described in Chapter 16. Although it is the final chapter of our book, in some ways it represents a beginning, not an end, to processing text, for it describes how a common infrastructure can be used to produce a combinatorically diverse array of processing pipelines.

As much as we hope this book is self-contained, we also hope that for you it serves as the beginning and not an end. Each chapter has a long list of relevant work upon which it is based, allowing you to explore any subtopic in great detail. The large community of NLP researchers is growing throughout the world, and we hope you join us in our exciting efforts to process text automatically and that you interact with us at universities, at industrial research labs, at conferences, in blogs, on social networks, and elsewhere. The multilingual NLP systems of the future are going to be even more exciting than the ones we have now, and we look forward to all your contributions!

Acknowledgments

This book was, from its inception, designed as a highly collaborative effort. We are immensely grateful for the encouraging support obtained from the beginning from IBM Press/Prentice Hall, especially from Bernard Goodwin and all the others at IBM Press who helped us get this project off the ground and see it to completion. A book of this kind would also not have been possible without the generous time, effort, and technical acumen of our fellow chapter authors, so we owe huge thanks to Otakar Smrž, Hyun-Jo You, Dilek Hakkani-Tür, Gokhan Tur, Benoit Favre, Elizabeth Shriberg, Anoop Sarkar, Sameer Pradhan, Katrin Kirchhoff, Mark Sammons, V.G. Vinod Vydiswaran, Dan Roth, Carmen Banea, Rada Mihalcea, Janyce Wiebe, Xiaqiang Luo, Philipp Koehn, Philipp Sorg, Philipp Cimiano, Frank Schilder, Liang Zhou, Nico Schlaefer, Jennifer Chu-Carroll, Vittorio Castelli, Radu Florian, Roberto Pieraccini, David Suendermann, John F. Pitrelli, and Burn Lewis. Daniel M. Bikel is also grateful to Google Research, especially to Corinna Cortes, for her support during the final stages of this project. Finally, we—Daniel M. Bikel and Imed Zitouni—would like to express our great appreciation for the backing of IBM Research, with special thanks to Ellen Yoffa, without whom this project would not have been possible.

This page intentionally left blank

About the Authors



Daniel M. Bikel (dbikel@google.com) is a senior research scientist at Google. He graduated with honors from Harvard in 1993 with a degree in Classics–Ancient Greek and Latin. From 1994 to 1997, he worked at BBN on several natural language processing problems, including development of the first high-accuracy stochastic name-finder, for which he holds a patent. He received M.S. and Ph.D. degrees in computer science from the University of Pennsylvania, in 2000 and 2004 respectively, discovering new properties of statistical parsing algorithms. From 2004 through 2010, he was a research staff member at IBM Research, working on a wide variety of natural language processing problems, including parsing, semantic role

labeling, information extraction, machine translation, and question answering. Dr. Bikel has been a reviewer for the *Computational Linguistics* journal, and has been on the program committees of the ACL, NAACL, EACL, and EMNLP conferences. He has published numerous peer-reviewed papers in the leading conferences and journals and has built software tools that have seen widespread use in the natural language processing community. In 2008, he won a Best Paper Award (Outstanding Short Paper) at the ACL-08: HLT conference. Since 2010, Dr. Bikel has been doing natural language processing and speech processing research at Google.



Imed Zitouni (izitouni@us.ibm.com) is a senior researcher working for IBM since 2004. He received his M.Sc. and Ph.D. in computer science with honors from University of Nancy, France in 1996 and 2000 respectively. In 1995, he obtained an MEng degree in computer science from Ecole Nationale des Sciences de l'Informatique, a prestigious national computer institute in Tunisia. Before joining IBM, he was a principal scientist at a startup company, DIALOCA, in 1999 and 2000. He then joined Bell Laboratories Lucent-Alcatel between 2000 and 2004 as a research staff member. His research interests include natural language processing, language modeling, spoken dialog systems, speech recognition, and machine learning.

Dr. Zitouni is a member of the IEEE Speech and Language Technical Committee in 2009–2011. He is the associate editor of the *ACM Transactions on Asian Language Information Processing* and the information officer of the Association for Computational Linguistics (ACL) Special Interest Group on Computational Approaches to Semitic Languages. He is a senior member of IEEE and member of ISCA and ACL. He served on the program

committee and as a chair for several peer-review conferences and journals. He holds several patents in the field and authored more than seventy-five papers in peer-review conferences and journals.



Carmen Banea (carmen.banea@gmail.com) is a doctoral student in the Department of Computer Science and Engineering, University of North Texas. She is working in the field of natural language processing. Her research work focuses primarily on multilingual approaches to subjectivity and sentiment analysis, where she developed both dictionary- and corpus-based methods that leverage languages with rich resources to create tools and data in other languages. Carmen has authored papers in major natural language processing conferences, including the Association for Computational Linguistics, Empirical Methods in Natural Language Processing, and the International Conference on Computational Linguistics. She served

as a program committee member in numerous large conferences and was also a reviewer for the *Computational Linguistics Journal* and the *Journal of Natural Language Engineering*. She cochaired the TextGraphs 2010 Workshop collocated with ACL 2010 and was one of the organizers of the University of North Texas site of the North American Computational Linguistics Olympiad in 2009 to 2011.



Vittorio Castelli (vittorio@us.ibm.com) received a Laurea degree in electrical engineering from Politecnico di Milano in 1988, an M.S. in electrical engineering in 1990, an M.S. in statistics in 1994, and a Ph.D. in electrical engineering in 1995, with a dissertation on information theory and statistical classification. In 1995 he joined the IBM T. J. Watson Research Center. His recent work is in natural language processing, specifically in information extraction; he has worked on the DARPA GALE and machine reading projects. Vittorio previously started the Personal Wizards project, aimed at capturing procedural knowledge from observation of experts performing a task. He has also done work on foundations of informa-

tion theory, memory compression, time series prediction and indexing, performance analysis, methods for improving the reliability and serviceability of computer systems, and digital libraries for scientific imagery. From 1996 to 1998 he was coinvestigator of the NASA/CAN project no. NCC5-101. His main research interests include information theory, probability theory, statistics, and statistical pattern recognition. From 1998 to 2005 he was an adjunct assistant professor at Columbia University, teaching information theory and statistical pattern recognition. He is a member of Sigma Xi, of the IEEE IT Society, and of the American Statistical Association. Vittorio has published papers on natural language processing computer-assisted instruction, statistical classification, data compression, image processing, multimedia databases, database mining and multidimensional indexing structures, intelligent user interfaces, and foundational problems in information theory, and he coedited *Image Databases: Search and Retrieval of Digital Imagery* (Wiley, 2002).



Jennifer Chu-Carroll (jence@us.ibm.com) is a research staff member in the Semantic Analysis and Integration Department at the IBM T. J. Watson Research Center. Before joining IBM in 2001, she spent five years as a member of technical staff at Lucent Technologies Bell Laboratories. Her research interests include question answering, semantic search, discourse processing, and spoken dialog management.



Philipp Cimiano (cimiano@cit-ec.uni-bielefeld.de) is professor in computer science at the University of Bielefeld, Germany. He leads the Semantic Computing Group that is affiliated with the Cognitive Interaction Technology Excellence Center, funded by the Deutsche Forschungsgemeinschaft in the framework of the excellence initiative. Philipp Cimiano graduated in computer science (major) and computational linguistics (minor) from the University of Stuttgart. He obtained his doctoral degree (summa cum laude) from the University of Karlsruhe. His main research interest lies in the combination of natural language with semantic technologies. In the last several years, he has focused on multilingual information access. He

has been involved as main investigator in a number of European (Dot.Kom, X-Media, Monnet) as well as national research projects such as SmartWeb (BMBF) and Multipla (DFG).



Benoit Favre (benoit.favre@lif.univ-mrs.fr) is an associate professor at Aix-Marseille Université, Marseille, France. He is a researcher in the field of natural language understanding. His research interests are in speech and text understanding with a focus on machine learning approaches. He received his Ph.D. from the University of Avignon, France, in 2007 on the topic of automatic speech summarization. Benoit was a teaching assistant at University of Avignon between 2003 and 2007 and a research engineer at Thales Land & Joint Systems, Paris, during the same period. Between 2007 and 2009, Benoit held a postdoctoral position at the International Computer Institute (Berkeley, CA) working with the speech group. From

2009 to 2010, he held a postdoctoral position at University of Le Mans, France. Since 2010, he is a tenured associate professor at Aix-Marseille Université, member of Laboratoire d'Informatique Fondamentale. Benoit is the coauthor of more than thirty refereed papers in international conferences and journals. He was a reviewer for major conferences in the field (ICASSP, Interspeech, ACL, EMNLP, Coling, NAACL) and for the *IEEE Transactions on Speech and Language Processing*. He is a member of the International Speech Communication Association and IEEE.



Radu Florian (raduf@us.ibm.com) is the manager of the Statistical Content Analytics (Information Extraction) group at IBM. He received his Ph.D. in 2002 from Johns Hopkins University, when he joined the Multilingual NLP group at IBM. At IBM, he has worked on a variety of research projects in the area of information extraction: mention detection, coreference resolution, relation extraction, cross-document coreference, and targeted information retrieval. Radu led research groups participating in several DARPA programs (GALE Distillation, MRP) and NIST-organized evaluations (ACE, TAC-KBP) and joint development programs with IBM partners for text mining in the medical domain (with Nuance), and contributed to the Watson *Jeopardy!* project.



Dilek Hakkani-Tür (Dilek.Hakkani-Tur@microsoft.com) is a principal scientist at Microsoft. Before joining Microsoft, she was with the International Computer Science Institute (ICSI) speech group (2006–2010) and AT&T Labs–Research (2001–2005). She received her B.Sc. degree from Middle East Technical University in 1994, and M.Sc. and Ph.D. degrees from Bilkent University, department of computer engineering, in 1996 and 2000 respectively. Her Ph.D. thesis is on statistical language modeling for agglutinative languages. She worked on machine translation at Carnegie Mellon University, Language Technologies Institute, in 1997 and at Johns Hopkins University in 1998. Between 1998 and 1999, Dilek worked on using

lexical and prosodic information for information extraction from speech at SRI International. Her research interests include natural language and speech processing, spoken dialog systems, and active and unsupervised learning for language processing. She holds 13 patents and has coauthored over one hundred papers in natural language and speech processing. She was an associate editor of *IEEE Transactions on Audio, Speech and Language Processing* between 2005 and 2008 and currently serves as an elected member of the IEEE Speech and Language Technical Committee (2009–2012).

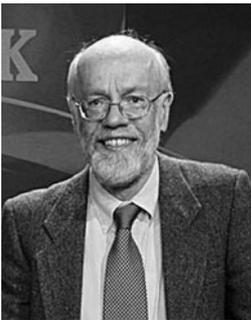


Katrin Kirchhoff (kk2@u.washington.edu) is a research associate professor in electrical engineering at the University of Washington. Her main research interests are automatic speech recognition, natural language processing, and human–computer interfaces, with particular emphasis on multilingual applications. She has authored over seventy peer-reviewed publications and is coeditor of *Multilingual Speech Processing*. Katrin currently serves as a member of the IEEE Speech Technical Committee and on the editorial boards of *Computer, Speech and Language* and *Speech Communication*.



Philipp Koehn (pkoehn@inf.ed.ac.uk) is a reader at the University of Edinburgh. He received his Ph.D. from the University of Southern California, where he was a research assistant at the Information Sciences Institute from 1997 to 2003. He was a postdoctoral research associate at the Massachusetts Institute of Technology in 2004 and joined the University of Edinburgh as a lecturer in 2005. His research centers on statistical machine translation, but he has also worked on speech, text classification, and information extraction. His major contribution to the machine translation community are the preparation and release of the Europarl corpus as well as the Pharaoh and Moses decoder. He is president of the ACL Special Interest Group on

Machine Translation and author of *Statistical Machine Translation* (Cambridge University Press, 2010).



Burn L. Lewis (burn@us.ibm.com) is a member of the computer science department at the IBM Thomas J. Watson Research Center. He received B.E. and M.E. degrees in electrical engineering from the University of Auckland in 1967 and 1968, respectively, and a Ph.D. in electrical engineering and computer science from the University of California–Berkeley in 1974. He subsequently joined IBM at the T. J. Watson Research Center, where he has worked on speech recognition and unstructured information management.



Xiaqiang Luo (xiaoluo@us.ibm.com) is a research staff member at IBM T. J. Watson Research Center. He has extensive experiences in human language technology, including speech recognition, spoken dialog systems, and natural language processing. He is a major contributor to IBM's success in many government-sponsored projects in the area of speech and language technology. He received the prestigious IBM Outstanding Technical Achievement Award in 2007, IBM ThinkPlace Bravo Award in 2006, and numerous invention achievement awards. Dr. Luo received his Ph.D. and M.S. in electrical engineering from Johns Hopkins University in 1999 and 1995, respectively, and B.A. in electrical engineering from University of Science and Technology of China in 1990. Dr. Luo is a member of the Association of Computational Linguistics and has served as program committee member for major technical conferences in the area of human language and artificial intelligence. He is a board member of the Chinese Association for Science and Technology (Greater New York Chapter). He served as an associate editor for *ACM Transactions on Asian Language Information Processing (TALIP)* from 2007 to 2010.

Dr. Luo is a member of the Association of Computational Linguistics and has served as program committee member for major technical conferences in the area of human language and artificial intelligence. He is a board member of the Chinese Association for Science and Technology (Greater New York Chapter). He served as an associate editor for *ACM Transactions on Asian Language Information Processing (TALIP)* from 2007 to 2010.



Rada Mihalcea (rada@cs.unt.edu) is associate professor in the Department of Computer Science and Engineering, University of North Texas. Her research interests are in computational linguistics, with a focus on lexical semantics, graph-based algorithms for natural language processing, and multilingual natural language processing. She is currently involved in a number of research projects, including word sense disambiguation, monolingual and crosslingual semantic similarity, automatic keyword extraction and text summarization, emotion and sentiment analysis, and computational humor. Rada serves or has served on the editorial boards of the *Journals of Computational Linguistics*, *Language Resources and Evaluations*,

Natural Language Engineering, and *Research in Language in Computation*. Her research has been funded by the National Science Foundation, Google, the National Endowment for the Humanities, and the State of Texas. She is the recipient of a National Science Foundation CAREER award (2008) and a Presidential Early Career Award for Scientists and Engineers (PECASE, 2009).



Roberto Pieraccini (www.robtopieraccini.com) is chief technology officer of SpeechCycle Inc. Roberto graduated in electrical engineering at the University of Pisa, Italy, in 1980. In 1981 he started working as a speech recognition researcher at CSELT, the research institution of the Italian telephone operating company. In 1990 he joined Bell Laboratories (Murray Hill, NJ) as a member of technical staff where he was involved in speech recognition and spoken language understanding research. He then joined AT&T Labs in 1996, where he started working on spoken dialog research. In 1999 he was director of R&D for SpeechWorks International. In 2003 he joined IBM T. J. Watson Research where he managed the Advanced Con-

versational Interaction Technology department, and then joined SpeechCycle in 2005 as their CTO. Roberto Pieraccini is the author of more than one hundred twenty papers and articles on speech recognition, language modeling, character recognition, language understanding, and automatic spoken dialog management. He is an ISCA and IEEE Fellow, a member of the editorial board of the *IEEE Signal Processing Magazine* and of the *International Journal of Speech Technology*. He is also a member of the Applied Voice Input Output Society and Speech Technology Consortium boards.



John F. Pitrelli (pitrelli@us.ibm.com) is a member of the Multilingual Natural Language Processing department at the IBM T. J. Watson Research Center in Yorktown Heights, New York. He received S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in 1983, 1985, and 1990 respectively, with graduate work in speech recognition and synthesis. Before his current position, he worked in the Speech Technology Group at NYNEX Science & Technology, Inc., in White Plains, New York; was a member of the IBM Pen Technologies Group; and worked on speech synthesis and prosody in the Human Language Technologies group at Watson. John's

research interests include natural language processing, speech synthesis, speech recognition, handwriting recognition, statistical language modeling, prosody, unstructured information management, and confidence modeling for recognition. He has published forty papers and holds four patents.



Sameer Pradhan (sameer.pradhan@Colorado.edu) is a scientist at BBN Technologies in Cambridge, Massachusetts. He is the author of a number of widely cited articles and chapters in the field of computational semantics. He is currently creating the next generation of semantic analysis engines and their applications, through algorithmic innovation, wide distribution of research tools such as Automatic Statistical SEMantic Role Tagger (ASSERT), and through the generation of rich, multilayer, multilingual, integrated resources, such as OntoNotes, that serve as a platform. Eventually these models of semantics should replace the currently impoverished, mostly word-based models, prevalent in most application domains, and help

take the area of language understanding to a new level of richness. Sameer received his Ph.D. from the University of Colorado in 2005, and since then has been working at BBN Technologies developing the OntoNotes corpora as part of the DARPA Global Autonomous Language Exploitation program. He is a member of ACL, and is a founding member of ACL's Special Interest Group for Annotation, promoting innovation in the area of annotation. He has regularly been on the program committees of various natural language processing conferences and workshops such as ACL, HLT, EMNLP, CoNLL, COLING, LREC, and LAW. He is also an accomplished chef.



Dan Roth (danr@illinois.edu) is a professor in the department of computer science and the Beckman Institute at the University of Illinois at Urbana-Champaign. He is a Fellow of AAAI, a University of Illinois Scholar, and holds faculty positions at the statistics and linguistics departments and at the Graduate School of Library and Information Science. Professor Roth's research spans theoretical work in machine learning and intelligent reasoning with a specific focus on learning and inference in natural language processing and intelligent access to textual information. He has published over two hundred papers in these areas and his papers have received multiple awards. He has developed advanced machine learning-based

tools for natural language applications that are being used widely by the research community, including an award-winning semantic parser. He was the program chair of AAAI'11, CoNLL'02, and ACL'03, and is or has been on the editorial board of several journals in his research areas. He is currently an associate editor for the *Journal of Artificial Intelligence Research* and the *Machine Learning Journal*. Professor Roth got his B.A. summa cum laude in mathematics from the Technion, Israel, and his Ph.D. in computer science from Harvard University.



Mark Sammons (mssammon@illinois.edu) is a principal research scientist working with the Cognitive Computation Group at the University of Illinois at Urbana-Champaign. His primary interests are in natural language processing and machine learning, with a focus on integrating diverse information sources in the context of textual entailment. His work has focused on developing a textual entailment framework that can easily incorporate new resources, designing appropriate inference procedures for recognizing entailment, and identifying and developing automated approaches to recognize and represent implicit content in natural language text. Mark received his M.Sc. in computer science from the University of Illinois in 2004

and his Ph.D. in mechanical engineering from the University of Leeds, England, in 2000.



Anoop Sarkar (www.cs.sfu.ca/~anoop) is an associate professor of computing science at Simon Fraser University in British Columbia, Canada, where he codirects the Natural Language Laboratory (<http://natlang.cs.sfu.ca>). He received his Ph.D. from the Department of Computer and Information Sciences at the University of Pennsylvania under Professor Aravind Joshi for his work on semi-supervised statistical parsing and parsing for tree-adjointing grammars. Anoop's current research is focused on statistical parsing and machine translation (exploiting syntax or morphology, or both). His interests also include formal language theory and stochastic grammars, in particular tree automata and tree-adjointing grammars.



Frank Schilder (frank.schilder@thomsonreuters.com) is a lead research scientist at the Research & Development department of Thomson Reuters. He joined Thomson Reuters in 2004, where he has been doing applied research on summarization technologies and information extraction systems. His summarization work has been implemented as the snippet generator for search results of West-LawNext, the new legal research system produced by Thomson Reuters. His current research activities involve the participation in different research competitions such as the Text Analysis Conference carried out by the National Institute of Standards and Technology. He obtained a Ph.D. in cognitive science from the University of

Edinburgh, Scotland, in 1997. From 1997 to 2003, he was employed by the Department for Informatics at the University of Hamburg, Germany, first as a postdoctoral researcher and later as an assistant professor. Frank has authored several journal articles and book chapters, including “Natural Language Processing: Overview” from the *Encyclopedia of Language and Linguistics* (Elsevier, 2006), coauthored with Peter Jackson, the chief scientist of Thomson Reuters. In 2011, he jointly won the Thomson Reuters Innovation challenge. He serves as reviewer for journals in computational linguistics and as program committee member of various conferences organized by the Association of Computational Linguistics.



Nico Schlaefer (nico@cs.cmu.edu) is a Ph.D. candidate in the School of Computer Science at Carnegie Mellon University and an IBM Ph.D. Fellow. His research focus is the application of machine learning techniques to natural language processing tasks. Schlaefer developed algorithms that enable question-answering systems to find correct answers, even if the original information sources contain little relevant content, and a flexible architecture that supports the integration of such algorithms. Schlaefer is the primary author of OpenEphyra, one of the most widely used open-source question-answering systems. Nico also contributed a statistical source expansion approach to Watson, the computer that won against

human champions in the *Jeopardy!* quiz show. His approach automatically extends knowledge sources with related content from the Web and other large text corpora, making it easier for Watson to find answers and supporting evidence.



Elizabeth Shriberg (elshribe@microsoft.com) is currently a principal scientist at Microsoft; previously she was at SRI International (Menlo Park, CA). She is also affiliated with the International Computer Science Institute (Berkeley, CA) and CASL (University of Maryland). She received a B.A. from Harvard (1987) and a Ph.D. from the University of California–Berkeley (1994). Elizabeth’s main interest is in modeling spontaneous speech using both lexical and prosodic information. Her work aims to combine linguistic knowledge with corpora and techniques from automatic speech and speaker recognition to advance both scientific understanding and technology. She has published roughly two hundred papers in speech science and technology and has served as associate editor of language and speech, on the boards of Speech Communication and Computational Linguistics, on a range of conference and workshop boards, on the ISCA Advisory Council, and on the ICSLP Permanent Council. She has organized workshops and served on boards for the National Science Foundation, the European Commission, NWO (Netherlands), and has reviewed for an interdisciplinary range of conferences, workshops, and journals (e.g., *IEEE Transactions on Speech and Audio Processing*, *Journal of the Acoustical Society of America*, *Nature*, *Journal of Phonetics*, *Computer Speech and Language*, *Journal of Memory and Language*, *Memory and Cognition*, *Discourse Processes*). In 2009 she received the ISCA Fellow Award. In 2010 she became a Fellow of SRI.



Otakar Smrž (otakar.smrz@cmu.edu) is a postdoctoral research associate at Carnegie Mellon University in Qatar. He focuses on methods of learning from comparable corpora to improve statistical machine translation from and into Arabic. Otakar completed his doctoral studies in mathematical linguistics at Charles University in Prague. He designed and implemented the ElixirFM computational model of Arabic morphology using functional programming and has developed other open source software for natural language processing. He has been the principal investigator of the Prague Arabic Dependency Treebank. Otakar used to work as a research scientist at IBM Czech Republic, where he explored unsupervised semantic parsing as well as acoustic modeling for multiple languages. Otakar is a cofounder of the Džám-e Džam Language Institute in Prague.



Philipp Sorg (philipp.sorg@kit.edu) is a Ph.D. student at the Karlsruhe Institute of Technology, Germany. He has a researcher position at the Institute of Applied Informatics and Formal Description Methods. Philipp graduated in computer science at the University of Karlsruhe. His main research interest lies in multilingual information retrieval. His special focus is the exploitation of social semantics in the context of the Web 2.0. He has been involved in the European research project Active, as well as in the national research project Multipla (DFG).



David Suendermann (david@speechcycle.com) is the principal speech scientist at SpeechCycle Labs (New York). Dr. Suendermann has been working on various fields of speech technology research for the last ten years. He worked at multiple industrial and academic institutions including Siemens (Munich), Columbia University (New York), University of Southern California (Los Angeles), Universitat Politècnica de Catalunya (Barcelona), and Rheinisch Westfälische Technische Hochschule (Aachen, Germany). He has authored more than sixty publications and patents, including a book and five book chapters, and holds a Ph.D. from the Bundeswehr University in Munich.



Gokhan Tur (gokhan.tur@ieee.org) is currently with Microsoft working as a principal scientist. He received his B.S., M.S., and Ph.D. from the Department of Computer Science, Bilkent University, Turkey in 1994, 1996, and 2000 respectively. Between 1997 and 1999, Tur visited the Center for Machine Translation of Carnegie Mellon University, then the Department of Computer Science of Johns Hopkins University, and then the Speech Technology and Research Lab of SRI International. He worked at AT&T Labs–Research from 2001 to 2006 and at the Speech Technology and Research Lab of SRI International from 2006 to 2010. His research interests include spoken language understanding, speech and language processing, machine learning, and information retrieval and extraction. Tur has coauthored more than one hundred papers published in refereed journals or books and presented at international conferences. He is the editor of *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech* (Wiley, 2011). Dr. Tur is a senior member of IEEE, ACL, and ISCA, was a member of IEEE Signal Processing Society (SPS), Speech and Language Technical Committee (SLTC) for 2006–2008, and is currently an associate editor for *IEEE Transactions on Audio, Speech, and Language Processing*.



V. G. Vinod Vydiswaran (vgvinodv@illinois.edu) is currently a Ph.D. student in the Department of Computer Science at the University of Illinois, Urbana-Champaign. His thesis is on modeling information trustworthiness on the Web and is advised by professors ChengXiang Zhai and Dan Roth. His research interests include text informatics, natural language processing, machine learning, and information extraction. V. G. Vinod's work has included developing a textual entailment system and applying textual entailment to relation extraction and information retrieval. He received his M.S. from Indian Institute of Technology-Bombay in 2004, where he worked on conditional models for information extraction with Professor Sunita

Sarawagi. Later, he worked at Yahoo! Research & Development Center at Bangalore, India, on scaling information extraction technologies over the Web.



Janyce Wiebe (wiebe@cs.pitt.edu) is a professor of computer science and codirector of the Intelligent Systems Program at the University of Pittsburgh. Her research with students and colleagues has been in discourse processing, pragmatics, word-sense disambiguation, and probabilistic classification in natural language processing. A major concentration of her research is subjectivity analysis, recognizing and interpreting expressions of opinions and sentiments in text, to support natural language processing applications such as question answering, information extraction, text categorization, and summarization. Janyce's current and past professional roles include ACL program cochair, NAACL program chair, NAACL executive

board member, computational linguistics, and language resources and evaluation, editorial board member, AAI workshop cochair, ACM special interest group on artificial intelligence (SIGART) vice-chair, and ACM-SIGART/AAAI doctoral consortium chair.



Hyun-Jo You (youhyunjo@gmail.com) is currently a lecturer in the Department of Linguistics, Seoul National University. He received his Ph.D. from Seoul National University. His research interests include quantitative linguistics, statistical language modeling, and computerized corpus analysis. He is especially interested in studying the morpho-syntactic and discourse structure in morphologically rich, free word order languages such as Korean, Czech, and Russian.



Liang Zhou (liangz@isi.edu) is a research scientist at Thomson Reuters Corporation. She has extensive knowledge in natural language processing, including sentiment analysis, automated text summarization, text understanding, information extraction, question answering, and information distillation. During her graduate studies at the Information Sciences Institute, she was actively involved in various government-sponsored projects, such as NIST Document Understanding conferences and DARPA Global Autonomous Language Exploitation. Dr. Zhou received her Ph.D. from the University of Southern California in 2006, M.S. from Stanford University in 2001, and B.S. from the University of Tennessee in 1999, all in computer science.

This page intentionally left blank

Chapter 1

Finding the Structure of Words

Otakar Smrž and Hyun-Jo You

Human language is a complicated thing. We use it to express our thoughts, and through language, we receive information and infer its meaning. Linguistic expressions are not unorganized, though. They show structure of different kinds and complexity and consist of more elementary components whose co-occurrence in context refines the notions they refer to in isolation and implies further meaningful relations between them.

Trying to understand language en bloc is not a viable approach. Linguists have developed whole disciplines that look at language from different perspectives and at different levels of detail. The point of morphology, for instance, is to study the variable forms and functions of words, while syntax is concerned with the arrangement of words into phrases, clauses, and sentences. Word structure constraints due to pronunciation are described by phonology, whereas conventions for writing constitute the orthography of a language. The meaning of a linguistic expression is its semantics, and etymology and lexicology cover especially the evolution of words and explain the semantic, morphological, and other links among them.

Words are perhaps the most intuitive units of language, yet they are in general tricky to define. Knowing how to work with them allows, in particular, the development of syntactic and semantic abstractions and simplifies other advanced views on language. Morphology is an essential part of language processing, and in multilingual settings, it becomes even more important.

In this chapter, we explore how to identify words of distinct types in human languages, and how the internal structure of words can be modeled in connection with the grammatical properties and lexical concepts the words should represent. The discovery of word structure is **morphological parsing**.

How difficult can such tasks be? It depends. In many languages, words are delimited in the orthography by whitespace and punctuation. But in many other languages, the writing system leaves it up to the reader to tell words apart or determine their exact phonological forms. Some languages use words whose form need not change much with the varying context; others are highly sensitive about the choice of word forms according to particular syntactic and semantic constraints and restrictions.

1.1 Words and Their Components

Words are defined in most languages as the smallest linguistic units that can form a complete utterance by themselves. The minimal parts of words that deliver aspects of meaning to them are called morphemes. Depending on the means of communication, morphemes are spelled out via graphemes—symbols of writing such as letters or characters—or are realized through phonemes, the distinctive units of sound in spoken language.¹ It is not always easy to decide and agree on the precise boundaries discriminating words from morphemes and from phrases [1, 2].

1.1.1 Tokens

Suppose, for a moment, that words in English are delimited only by whitespace and punctuation [3], and consider Example 1–1:

EXAMPLE 1–1: Will you read the newspaper? Will you read it? I won't read it.

If we confront our assumption with insights from etymology and syntax, we notice two words here: *newspaper* and *won't*. Being a compound word, *newspaper* has an interesting derivational structure. We might wish to describe it in more detail, once there is a lexicon or some other linguistic evidence on which to build the possible hypotheses about the origins of the word. In writing, *newspaper* and the associated concept is distinguished from the isolated *news* and *paper*. In speech, however, the distinction is far from clear, and identification of words becomes an issue of its own.

For reasons of generality, linguists prefer to analyze *won't* as two syntactic words, or tokens, each of which has its independent role and can be reverted to its normalized form. The structure of *won't* could be parsed as *will* followed by *not*. In English, this kind of **tokenization** and **normalization** may apply to just a limited set of cases, but in other languages, these phenomena have to be treated in a less trivial manner.

In Arabic or Hebrew [4], certain tokens are concatenated in writing with the preceding or the following ones, possibly changing their forms as well. The underlying lexical or syntactic units are thereby blurred into one compact string of letters and no longer appear as distinct words. Tokens behaving in this way can be found in various languages and are often called clitics.

In the writing systems of Chinese, Japanese [5], and Thai, whitespace is not used to separate words. The units that are delimited graphically in some way are sentences or clauses. In Korean, character strings are called *eojeol* 'word segment' and roughly correspond to speech or cognitive units, which are usually larger than words and smaller than clauses [6], as shown in Example 1–2:

EXAMPLE 1–2: 학생들에게만 주셨는데

hak.sayng.tul.ey.key.man cwu.syess.nun.te²

haksayng-tul-eykey-man cwu-si-ess-nunte

student+plural+dative+only give+honorific+past+while

while (he/she) gave (it) only to the students

1. Signs used in sign languages are composed of elements denoted as phonemes, too.

2. We use the Yale romanization of the Korean script and indicate its original characters by dots. Hyphens mark morphological boundaries, and tokens are separated by plus symbols.

Nonetheless, the elementary morphological units are viewed as having their own syntactic status [7]. In such languages, tokenization, also known as **word segmentation**, is the fundamental step of morphological analysis and a prerequisite for most language processing applications.

1.1.2 Lexemes

By the term word, we often denote not just the one linguistic form in the given context but also the concept behind the form and the set of alternative forms that can express it. Such sets are called lexemes or lexical items, and they constitute the lexicon of a language. Lexemes can be divided by their behavior into the lexical categories of verbs, nouns, adjectives, conjunctions, particles, or other parts of speech. The citation form of a lexeme, by which it is commonly identified, is also called its lemma.

When we convert a word into its other forms, such as turning the singular *mouse* into the plural *mice* or *mouses*, we say we inflect the lexeme. When we transform a lexeme into another one that is morphologically related, regardless of its lexical category, we say we derive the lexeme: for instance, the nouns *receiver* and *reception* are derived from the verb *to receive*.

EXAMPLE 1–3: Did you see him? I didn't see him. I didn't see anyone.

Example 1–3 presents the problem of tokenization of *didn't* and the investigation of the internal structure of *anyone*. In the paraphrase *I saw no one*, the lexeme *to see* would be inflected into the form *saw* to reflect its grammatical function of expressing positive past tense. Likewise, *him* is the oblique case form of *he* or even of a more abstract lexeme representing all personal pronouns. In the paraphrase, *no one* can be perceived as the minimal word synonymous with *nobody*. The difficulty with the definition of what counts as a word need not pose a problem for the syntactic description if we understand *no one* as two closely connected tokens treated as one fixed element.

In the Czech translation of Example 1–3, the lexeme *vidět* 'to see' is inflected for past tense, in which forms comprising two tokens are produced in the second and first person (i.e., *viděla jsi* 'you-FEM-SG saw' and *neviděla jsem* 'I-FEM-SG did not see'). Negation in Czech is an inflectional parameter rather than just syntactic and is marked both in the verb and in the pronoun of the latter response, as in Example 1–4:

EXAMPLE 1–4: Vidělas ho? Neviděla jsem ho. Neviděla jsem nikoho.
saw+you-are him? not-saw I-am him. not-saw I-am no-one.

Here, *vidělas* is the contracted form of *viděla jsi* 'you-FEM-SG saw'. The *s* of *jsi* 'you are' is a clitic, and due to free word order in Czech, it can be attached to virtually any part of speech. We could thus ask a question like *Nikoho neviděla?* 'Did you see no one?' in which the pronoun *nikoho* 'no one' is followed by this clitic.

1.1.3 Morphemes

Morphological theories differ on whether and how to associate the properties of word forms with their structural components [8, 9, 10, 11]. These components are usually called **segments** or **morphs**. The morphs that by themselves represent some aspect of the meaning of a word are called **morphemes** of some function.

Human languages employ a variety of devices by which morphs and morphemes are combined into word forms. The simplest morphological process concatenates morphs one by one, as in *dis-agree-ment-s*, where *agree* is a free lexical morpheme and the other elements are bound grammatical morphemes contributing some partial meaning to the whole word.

In a more complex scheme, morphs can interact with each other, and their forms may become subject to additional phonological and orthographic changes denoted as morpho-phonemic. The alternative forms of a morpheme are termed **allomorphs**.

Examples of morphological alternation and phonologically dependent choice of the form of a morpheme are abundant in the Korean language. In Korean, many morphemes change their forms systematically with the phonological context. Example 1–5 lists the allomorphs *-ess-*, *-ass-*, *-yess-* of the temporal marker indicating past tense. The first two alter according to the phonological condition of the preceding verb stem; the last one is used especially for the verb *ha-* ‘do’. The appropriate allomorph is merely concatenated after the stem, or it can be further contracted with it, as was *-si-ess-* into *-syess-* in Example 1–2. During morphological parsing, normalization of allomorphs into some canonical form of the morpheme is desirable, especially because the contraction of morphs interferes with simple segmentation:

EXAMPLE 1–5:	concatenated	contracted	
(a)	보았- <i>po-ass-</i>	봤- <i>pwass-</i>	‘have seen’
(b)	가지었- <i>ka.ci-ess-</i>	가졌- <i>ka.cyess-</i>	‘have taken’
(c)	하였- <i>ha-yess-</i>	했- <i>hayss-</i>	‘have done’
(d)	되었- <i>toy-ess-</i>	됐- <i>twayss-</i>	‘have become’
(e)	놓았- <i>noh-ass-</i>	놨- <i>nwass-</i>	‘have put’

Contractions (a, b) are ordinary but require attention because two characters are reduced into one. Other types (c, d, e) are phonologically unpredictable, or lexically dependent. For example, *coh-ass-* ‘have been good’ may never be contracted, whereas *noh-* and *-ass-* are merged into *nwass-* in (e).

There are yet other linguistic devices of word formation to account for, as the morphological process itself can get less trivial. The concatenation operation can be complemented with infixation or intertwining of the morphs, which is common, for instance, in Arabic. Nonconcatenative inflection by modification of the internal vowel of a word occurs even in English: compare the sounds of *mouse* and *mice*, *see* and *saw*, *read* and *read*.

Notably in Arabic, internal inflection takes place routinely and has a yet different quality. The internal parts of words, called stems, are modeled with root and pattern morphemes. Word structure is then described by templates abstracting away from the root but showing the pattern and all the other morphs attached to either side of it.

<p>EXAMPLE 1–6: h1 stqr0 h*h AljrA}d?³ <i>hal sa-taqrau hāḏihi 'l-ǧarāḏa?</i> whether will+you-read this the-newspapers? h1 stqrWhA? ln Oqr0hA. <i>hal sa-taqrauhā? lan ʔaqraahā.</i> whether will+you-read+it? not-will I-read+it.</p>	<p>هل ستقرأ هذه الجرائد؟ هل ستقرأها؟ لن أقرأها.</p>
---	--

3. The original Arabic script is transliterated using Buckwalter notation. For readability, we also provide the standard phonological transcription, which reduces ambiguity.

The meaning of Example 1–6 is similar to that of Example 1–1, only the phrase *hādīhi 'l-ġarā'ida* refers to ‘these newspapers’. While *sa-taqrā'u* ‘you will read’ combines the future marker *sa-* with the imperfective second-person masculine singular verb *taqrā'u* in the indicative mood and active voice, *sa-taqrā'uhā* ‘you will read it’ also adds the cliticized feminine singular personal pronoun in the accusative case.⁴

The citation form of the lexeme to which *taqrā'u* ‘you-MASC-SG read’ belongs is *qarā*, roughly ‘to read’. This form is classified by linguists as the basic verbal form represented by the template *faʿal* merged with the consonantal root *q r ʿ*, where the *f ʿ l* symbols of the template are substituted by the respective root consonants. Inflections of this lexeme can modify the pattern *faʿal* of the stem of the lemma into *ʿal* and concatenate it, under rules of morphophonemic changes, with further prefixes and suffixes. The structure of *taqrā'u* is thus parsed into the template *ta-ʿal-u* and the invariant root.

The word *al-ġarā'ida* ‘the newspapers’ in the accusative case and definite state is another example of internal inflection. Its structure follows the template *al-faʿā'il-a* with the root *ġ r d*. This word is the plural of *ġarīdah* ‘newspaper’ with the template *faʿīl-ah*. The links between singular and plural templates are subject to convention and have to be declared in the lexicon.

Irrespective of the morphological processes involved, some properties or features of a word need not be apparent explicitly in its morphological structure. Its existing structural components may be paired with and depend on several functions simultaneously but may have no particular grammatical interpretation or lexical meaning.

The *-ah* suffix of *ġarīdah* ‘newspaper’ corresponds with the inherent feminine gender of the lexeme. In fact, the *-ah* morpheme is commonly, though not exclusively, used to mark the feminine singular forms of adjectives: for example, *ġadīd* becomes *ġadīdah* ‘new’. However, the *-ah* suffix can be part of words that are not feminine, and there its function can be seen as either emptied or overridden [12]. In general, linguistic forms should be distinguished from functions, and not every morph can be assumed to be a morpheme.

1.1.4 Typology

Morphological typology divides languages into groups by characterizing the prevalent morphological phenomena in those languages. It can consider various criteria, and during the history of linguistics, different classifications have been proposed [13, 14]. Let us outline the typology that is based on quantitative relations between words, their morphemes, and their features:

Isolating, or **analytic**, languages include no or relatively few words that would comprise more than one morpheme (typical members are Chinese, Vietnamese, and Thai; analytic tendencies are also found in English).

Synthetic languages can combine more morphemes in one word and are further divided into agglutinative and fusional languages.

Agglutinative languages have morphemes associated with only a single function at a time (as in Korean, Japanese, Finnish, and Tamil, etc.).

4. The logical plural of things is formally treated as feminine singular in Arabic.

Fusional languages are defined by their feature-per-morpheme ratio higher than one (as in Arabic, Czech, Latin, Sanskrit, German, etc.).

In accordance with the notions about word formation processes mentioned earlier, we can also discern:

Concatenative languages linking morphs and morphemes one after another.

Nonlinear languages allowing structural components to merge nonsequentially to apply tonal morphemes or change the consonantal or vocalic templates of words.

While some morphological phenomena, such as orthographic collapsing, phonological contraction, or complex inflection and derivation, are more dominant in some languages than in others, in principle, we can find, and should be able to deal with, instances of these phenomena across different language families and typological classes.

1.2 Issues and Challenges

Morphological parsing tries to eliminate or alleviate the variability of word forms to provide higher-level linguistic units whose lexical and morphological properties are explicit and well defined. It attempts to remove unnecessary irregularity and give limits to ambiguity, both of which are present inherently in human language.

By irregularity, we mean existence of such forms and structures that are not described appropriately by a prototypical linguistic model. Some irregularities can be understood by redesigning the model and improving its rules, but other lexically dependent irregularities often cannot be generalized.

Ambiguity is indeterminacy in interpretation of expressions of language. Next to accidental ambiguity and ambiguity due to lexemes having multiple senses, we note the issue of **syncretism**, or systematic ambiguity.

Morphological modeling also faces the problem of productivity and creativity in language, by which unconventional but perfectly meaningful new words or new senses are coined. Usually, though, words that are not licensed in some way by the lexicon of a morphological system will remain completely unparsed. This **unknown word** problem is particularly severe in speech or writing that gets out of the expected domain of the linguistic model, such as when special terms or foreign names are involved in the discourse or when multiple languages or dialects are mixed together.

1.2.1 Irregularity

Morphological parsing is motivated by the quest for generalization and abstraction in the world of words. Immediate descriptions of given linguistic data may not be the ultimate ones, due to either their inadequate accuracy or inappropriate complexity, and better formulations may be needed. The design principles of the morphological model are therefore very important.

In Arabic, the deeper study of the morphological processes that are in effect during inflection and derivation, even for the so-called irregular words, is essential for mastering the

whole morphological and phonological system. With the proper abstractions made, irregular morphology can be seen as merely enforcing some extended rules, the nature of which is phonological, over the underlying or prototypical regular word forms [15, 16].

EXAMPLE 1–7: h1 r0yth? 1m 0rh. 1m 0r OHdA.

هل رأيته؟ لم أر أحدا.

hal raʔaytihi? lam ʔarahu. lam ʔara ʔahadan.

whether you-saw+him? not-did I-see+him. not-did I-see anyone.

In Example 1–7, *raʔayti* is the second-person feminine singular perfective verb in active voice, member of the *raʔā* ‘to see’ lexeme of the *r ʔ y* root. The prototypical, regularized pattern for this citation form is *faʔal*, as we saw with *qaraʔ* in Example 1–6. Alternatively, we could assume the pattern of *raʔā* to be *faʔā*, thereby asserting in a compact way that the final root consonant and its vocalic context are subject to the particular phonological change, resulting in *raʔā* like *faʔā* instead of *raʔay* like *faʔal*. The occurrence of this change in the citation form may have possible implications for the morphological behavior of the whole lexeme.

Table 1–1 illustrates differences between a naive model of word structure in Arabic and the model proposed in Smrž [12] and Smrž and Bielický [17] where morphophonemic merge rules and templates are involved. Morphophonemic templates capture morphological processes by just organizing stem patterns and generic affixes without any context-dependent variation of the affixes or ad hoc modification of the stems. The merge rules, indeed very terse, then ensure that such structured representations can be converted into exactly the surface forms, both orthographic and phonological, used in the natural language. Applying the merge rules is independent of and irrespective of any grammatical parameters or information other than that contained in a template. Most morphological irregularities are thus successfully removed.

Table 1–1: Discovering the regularity of Arabic morphology using morphophonemic templates, where uniform structural operations apply to different kinds of stems. In rows, surface forms *S* of *qaraʔ* ‘to read’ and *raʔā* ‘to see’ and their inflections are analyzed into immediate *I* and morphophonemic *M* templates, in which dashes mark the structural boundaries where merge rules are enforced. The outer columns of the table correspond to *P* perfective and *I* imperfective stems declared in the lexicon; the inner columns treat active verb forms of the following morphosyntactic properties: *I* indicative, *S* subjunctive, *J* jussive mood; 1 first, 2 second, 3 third person; *M* masculine, *F* feminine gender; *S* singular, *P* plural number

P-STEM	P–3MS	P–2FS	P–3MP	II2MS	IS1–S	IJ1–S	I-STEM	
<i>qaraʔ</i>	<i>qaraʔa</i>	<i>qaraʔti</i>	<i>qaraʔū</i>	<i>taqraʔu</i>	<i>ʔaqraʔa</i>	<i>ʔaqraʔ</i>	<i>qraʔ</i>	S
<i>faʔal</i>	<i>faʔal-a</i>	<i>faʔal-ti</i>	<i>faʔal-ū</i>	<i>ta-fal-u</i>	<i>ʔa-fal-a</i>	<i>ʔa-fal</i>	<i>faʔal</i>	I
<i>faʔal</i>	<i>faʔal-a</i>	<i>faʔal-ti</i>	<i>faʔal-ū</i>	<i>ta-fal-u</i>	<i>ʔa-fal-a</i>	<i>ʔa-fal-</i>	<i>faʔal</i>	M
...	...-a	...-ti	...-ū	ta-...-u	ʔa-...-a	ʔa-...-	...	
<i>faʔā</i>	<i>faʔā-a</i>	<i>faʔā-ti</i>	<i>faʔā-ū</i>	<i>ta-fā-u</i>	<i>ʔa-fā-a</i>	<i>ʔa-fā-</i>	<i>fā</i>	M
<i>faʔā</i>	<i>faʔā</i>	<i>faʔal-ti</i>	<i>faʔaw</i>	<i>ta-fā</i>	<i>ʔa-fā</i>	<i>ʔa-fa</i>	<i>fā</i>	I
<i>raʔā</i>	<i>raʔā</i>	<i>raʔayti</i>	<i>raʔaw</i>	<i>tarā</i>	<i>ʔarā</i>	<i>ʔara</i>	<i>rā</i>	S

Table 1–2: Examples of major Korean irregular verb classes compared with regular verbs

Base Form		(-e)		Meaning	Comment
집-	<i>cip-</i>	집어	<i>cip.e</i>	‘pick’	regular
깎-	<i>kip-</i>	기워	<i>ki.we</i>	‘sew’	<i>p</i> -irregular
믿-	<i>mit-</i>	믿어	<i>mit.e</i>	‘believe’	regular
싣-	<i>sit-</i>	실어	<i>sil.e</i>	‘load’	<i>t</i> -irregular
씻-	<i>ssis-</i>	씻어	<i>ssis.e</i>	‘wash’	regular
잇-	<i>is-</i>	이어	<i>i.e</i>	‘link’	<i>s</i> -irregular
낳-	<i>nah-</i>	낳아	<i>nah.a</i>	‘bear’	regular
까맣-	<i>kka.mah-</i>	까매	<i>kka.may</i>	‘be black’	<i>h</i> -irregular
치르-	<i>chi.lu-</i>	치러	<i>chi.le</i>	‘pay’	regular <i>u</i> -ellipsis
이르-	<i>i.lu-</i>	이르러	<i>i.lu.le</i>	‘reach’	<i>le</i> -irregular
흐르-	<i>hu.lu-</i>	흘러	<i>hul.le</i>	‘flow’	<i>lu</i> -irregular

In contrast, some irregularities are bound to particular lexemes or contexts, and cannot be accounted for by general rules. Korean irregular verbs provide examples of such irregularities.

Korean shows exceptional constraints on the selection of grammatical morphemes. It is hard to find irregular inflection in other agglutinative languages: two irregular verbs in Japanese [18], one in Finnish [19]. These languages are abundant with morphological alternations that are formalized by precise phonological rules. Korean additionally features lexically dependent stem alternation. As in many other languages, *i-* ‘be’ and *ha-* ‘do’ have unique irregular endings. Other irregular verbs are classified by the stem final phoneme. Table 1–2 compares major irregular verb classes with regular verbs in the same phonological condition.

1.2.2 Ambiguity

Morphological ambiguity is the possibility that word forms be understood in multiple ways out of the context of their discourse. Words forms that look the same but have distinct functions or meaning are called homonyms.

Ambiguity is present in all aspects of morphological processing and language processing at large. Morphological parsing is not concerned with complete disambiguation of words in their context, however; it can effectively restrict the set of valid interpretations of a given word form [20, 21].

In Korean, homonyms are one of the most problematic objects in morphological analysis because they prevail all around frequent lexical items. Table 1–3 arranges homonyms on the basis of their behavior with different endings. Example 1–8 is an example of homonyms through nouns and verbs.

Table 1–3: Systematic homonyms arise as verbs combined with endings in Korean

	(-ko)		(-e)		(-un)	Meaning
묻고	<i>mwut.ko</i>	묻어	<i>mwut.e</i>	묻은	<i>mwut.un</i>	‘bury’
묻고	<i>mwut.ko</i>	물어	<i>mwul.e</i>	물은	<i>mwul.un</i>	‘ask’
물고	<i>mwul.ko</i>	물어	<i>mwul.e</i>	문	<i>mwun</i>	‘bite’
걷고	<i>ket.ko</i>	걸어	<i>ket.e</i>	걸은	<i>ket.un</i>	‘roll up’
걷고	<i>ket.ko</i>	걸어	<i>kel.e</i>	걸은	<i>kel.un</i>	‘walk’
걸고	<i>kel.ko</i>	걸어	<i>kel.e</i>	건	<i>ken</i>	‘hang’
굽고	<i>kwup.ko</i>	굽어	<i>kwup.e</i>	굽은	<i>kwup.un</i>	‘be bent’
굽고	<i>kwup.ko</i>	구워	<i>kwu.we</i>	구운	<i>kwu.wun</i>	‘bake’
이르고	<i>i.lu.ko</i>	이르러	<i>i.lu.le</i>	이른	<i>i.lun</i>	‘reach’
이르고	<i>i.lu.ko</i>	일러	<i>il.le</i>	이른	<i>i.lun</i>	‘say’

EXAMPLE 1–8: 난 ‘orchid’ ← 난 *nan* ‘orchid’
 난 ‘I’ ← 나 *na* ‘I’ + *-n* (topic)
 난 ‘which flew’ ← 날- *nal-* ‘fly’ + *-n* (relative, past)
 난 ‘which got out’ ← 나- *na-* ‘get out’ + *-n* (relative, past)

We could also consider ambiguity in the senses of the noun *nan*, according to the Standard Korean Language Dictionary: *nan*¹ ‘egg’, *nan*² ‘revolt’, *nan*⁵ ‘section (in newspaper)’, *nan*⁶ ‘orchid’, plus several infrequent readings.

Arabic is a language of rich morphology, both derivational and inflectional. Because Arabic script usually does not encode short vowels and omits yet some other diacritical marks that would record the phonological form exactly, the degree of its morphological ambiguity is considerably increased. In addition, Arabic orthography collapses certain word forms together. The problem of morphological disambiguation of Arabic encompasses not only the resolution of the structural components of words and their actual morphosyntactic properties (i.e., morphological tagging [22, 23, 24]) but also tokenization and normalization [25], lemmatization, stemming, and diacritization [26, 27, 28].

When inflected syntactic words are combined in an utterance, additional phonological and orthographic changes can take place, as shown in Figure 1–1. In Sanskrit, one such euphony rule is known as external *sandhi* [29, 30]. Inverting *sandhi* during tokenization is usually nondeterministic in the sense that it can provide multiple solutions. In any language, tokenization decisions may impose constraints on the morphosyntactic properties of the tokens being reconstructed, which then have to be respected in further processing. The tight coupling between morphology and syntax has inspired proposals for disambiguating them jointly rather than sequentially [4].

Czech is a highly inflected fusional language. Unlike agglutinative languages, inflectional morphemes often represent several functions simultaneously, and there is no particular one-to-one correspondence between their forms and functions. Inflectional **paradigms**

<i>dirāsati</i>	درستي	drAsty	→	<i>dirāsatu ī</i>	دراسة ي	drAsp y
			→	<i>dirāsati ī</i>	دراسة ي	drAsp y
			→	<i>dirāsata ī</i>	دراسة ي	drAsp y
<i>mu'allimīya</i>	معلمي	mElmy	→	<i>mu'allimū ī</i>	معلمو ي	mElmw y
			→	<i>mu'allimī ī</i>	معلمي ي	mElmy y
<i>katabtumūhā</i>	كتبتموها	ktbtmwhA	→	<i>katabtum hā</i>	كتبتمها	ktbtm hA
<i>ʾiġrāʾuhu</i>	إجراؤه	IjrAWh	→	<i>ʾiġrāʾu hu</i>	إجراؤه	IjrA' h
<i>ʾiġrāʾihi</i>	إجرائه	IjrA}h	→	<i>ʾiġrāʾi hu</i>	إجراؤه	IjrA' h
<i>ʾiġrāʾahu</i>	إجراؤه	IjrA'h	→	<i>ʾiġrāʾa hu</i>	إجراؤه	IjrA' h
<i>li-ʾl-ʾasafi</i>	للأسف	l10sf	→	<i>li ʾl-ʾasafi li</i>	ل للأسف	l 1 A10sf

Figure 1–1: Complex tokenization and normalization of euphony in Arabic. Three nominal cases are expressed by the same word form with *dirāsati* ‘my study’ and *mu'allimīya* ‘my teachers’, but the original case endings are distinct. In *katabtumūhā* ‘you-MASC-PL wrote them’, the liaison vowel *ū* is dropped when tokenized. Special attention is needed to normalize some orthographic conventions, such as the interaction of *ʾiġrāʾ* ‘carrying out’ and the cliticized *hu* ‘his’ respecting the case ending or the merge of the definite article of *ʾasaf* ‘regret’ with the preposition *li* ‘for’

(i.e., schemes for finding the form of a lexeme associated with the required properties) in Czech are of numerous kinds, yet they tend to include nonunique forms in them.

Table 1–4 lists the paradigms of several common Czech words. Inflectional paradigms for nouns depend on the grammatical gender and the phonological structure of a lexeme. The individual forms in a paradigm vary with grammatical number and case, which are the free parameters imposed only by the context in which a word is used.

Looking at the morphological variation of the word *stavení* ‘building’, we might wonder why we should distinguish all the cases for it when this lexeme can take only four different forms. Is the detail of the case system appropriate? The answer is yes, because we can find linguistic evidence that leads to this case category abstraction. Just consider other words of the same meaning in place of *stavení* in various contexts. We conclude that there is indeed a case distinction made by the underlying system, but it need not necessarily be expressed clearly and uniquely in the form of words.

The morphological phenomenon that some words or word classes show instances of systematic homonymy is called syncretism. In particular, homonymy can occur due to **neutralization** and **uninflectedness** with respect to some morphosyntactic parameters. These cases of morphological syncretism are distinguished by the ability of the context to demand the morphosyntactic properties in question, as stated by Baerman, Brown, and Corbett [10, p. 32]:

Whereas *neutralization* is about syntactic irrelevance as reflected in morphology, *uninflectedness* is about morphology being unresponsive to a feature that is syntactically relevant.

For example, it seems fine for syntax in Czech or Arabic to request the personal pronoun of the first-person feminine singular, equivalent to ‘I’, despite it being homonymous with

Table 1–4: Morphological paradigms of the Czech words *dům* ‘house’, *budova* ‘building’, *stavba* ‘building’, *stavení* ‘building’. Despite systematic ambiguities in them, the space of inflectional parameters could not be reduced without losing the ability to capture all distinct forms elsewhere: *s* singular, *P* plural number; 1 nominative, 2 genitive, 3 dative, 4 accusative, 5 vocative, 6 locative, 7 instrumental case

	MASCULINE INANIMATE	FEMININE	FEMININE	NEUTER
S1	dům	budova	stavba	stavení
S2	domu	budovy	stavby	stavení
S3	domu	budově	stavbě	stavení
S4	dům	budovu	stavbu	stavení
S5	dome	budovo	stavbo	stavení
S6	domu / domě	budově	stavbě	stavení
S7	domem	budovou	stavbou	stavením
P1	domy	budovy	stavby	stavení
P2	domů	budov	staveb	stavení
P3	domům	budovám	stavbám	stavením
P4	domy	budovy	stavby	stavení
P5	domy	budovy	stavby	stavení
P6	domech	budovách	stavbách	staveních
P7	domy	budovami	stavbami	staveními

the first-person masculine singular. The reason is that for some other values of the person category, the forms of masculine and feminine gender are different, and there exist syntactic dependencies that do take gender into account. It is not the case that the first-person singular pronoun would have no gender nor that it would have both. We just observe uninflectedness here. On the other hand, we might claim that in English or Korean, the gender category is syntactically neutralized if it ever was present, and the nuances between *he* and *she*, *him* and *her*, *his* and *hers* are only semantic.

With the notion of paradigms and syncretism in mind, we should ask what is the minimal set of combinations of morphosyntactic inflectional parameters that covers the inflectional variability in a language. Morphological models that would like to define a joint system of underlying morphosyntactic properties for multiple languages would have to generalize the parameter space accordingly and neutralize any systematically void configurations.

1.2.3 Productivity

Is the inventory of words in a language finite, or is it unlimited? This question leads directly to discerning two fundamental approaches to language, summarized in the distinction between *langue* and *parole* by Ferdinand de Saussure, or in the competence versus performance duality by Noam Chomsky.

In one view, language can be seen as simply a collection of utterances (*parole*) actually pronounced or written (performance). This ideal data set can in practice be approximated by linguistic corpora, which are finite collections of linguistic data that are studied with empirical methods and can be used for comparison when linguistic models are developed.

Yet, if we consider language as a system (langue), we discover in it structural devices like recursion, iteration, or compounding that allow to produce (competence) an infinite set of concrete linguistic utterances. This general potential holds for morphological processes as well and is called morphological productivity [31, 32].

We denote the set of word forms found in a corpus of a language as its vocabulary. The members of this set are word types, whereas every original instance of a word form is a word token.

The distribution of words [33] or other elements of language follows the “80/20 rule,” also known as the law of the vital few. It says that most of the word tokens in a given corpus can be identified with just a couple of word types in its vocabulary, and words from the rest of the vocabulary occur much less commonly if not rarely in the corpus. Furthermore, new, unexpected words will always appear as the collection of linguistic data is enlarged.

In Czech, negation is a productive morphological operation. Verbs, nouns, adjectives, and adverbs can be prefixed with *ne-* to define the complementary lexical concept. In Example 1–9, *budeš* ‘you will be’ is the second-person singular of *být* ‘to be’, and *nebudu* ‘I will not be’ is the first-person singular of *nebýt*, the negated *být*. We could easily have *číst* ‘to read’ and *nečíst* ‘not to read’, or we could create an adverbial phrase like *noviny nenoviny* that would express ‘indifference to newspapers’ in general:

EXAMPLE 1–9: *Budeš číst ty noviny? Budeš je číst? Nebudu je číst.*
 you-will read the newspaper? you-will it read? not-I-will it read.

Example 1–9 has the meaning of Example 1–1 and Example 1–6. The word *noviny* ‘newspaper’ exists only in plural whether it signifies one piece of newspaper or many of them. We can literally translate *noviny* as the plural of *novina* ‘news’ to see the origins of the word as well as the fortunate analogy with English.

It is conceivable to include all negated lexemes into the lexicon and thereby again achieve a finite number of word forms in the vocabulary. Generally, though, the richness of a morphological system of a language can make this approach highly impractical.

Most languages contain words that allow some of their structural components to repeat freely. Consider the prefix *pra-* related to a notion of ‘generation’ in Czech and how it can or cannot be iterated, as shown in Example 1–10:

EXAMPLE 1–10: <i>vnuk</i> ‘grandson’	<i>pravnuke</i> ‘great-grandson’
	<i>prapra...vnuk</i> ‘great-great-...grandson’
<i>les</i> ‘forest’	<i>prales</i> ‘jungle’, ‘virgin forest’
<i>zdroj</i> ‘source’	<i>prazdroj</i> ‘urquell’, ‘original source’
<i>starý</i> ‘old’	<i>prastarý</i> ‘time-honored’, ‘dateless’

In creative language, such as in blogs, chats, and emotive informal communication, iteration is often used to accent intensity of expression. Creativity may, of course, go beyond the rules of productivity itself [32].

Let us give an example where creativity, productivity, and the issue of unknown words meet nicely. According to Wikipedia, the word *googol* is a made-up word denoting the number “one followed by one hundred zeros,” and the name of the company Google is an

inadvertent misspelling thereof. Nonetheless, both of these words successfully entered the lexicon of English where morphological productivity started working, and we now know the verb *to google* and nouns like *googling* or even *googlish* or *googleology* [34].

The original names have been adopted by other languages, too, and their own morphological processes have been triggered. In Czech, one says *googlovat*, *googlit* ‘to google’ or *vygooglovat*, *vygooglit* ‘to google out’, *googlování* ‘googling’, and so on. In Arabic, the names are transcribed as *ǧūǧūl* ‘googol’ and *ǧūǧīl* ‘Google’. The latter one got transformed to the verb *ǧawǧal* ‘to google’ through internal inflection, as if there were a genuine root *ǧ w ǧ l*, and the corresponding noun *ǧawǧalah* ‘googling’ exists as well.

1.3 Morphological Models

There are many possible approaches to designing and implementing morphological models. Over time, computational linguistics has witnessed the development of a number of formalisms and frameworks, in particular grammars of different kinds and expressive power, with which to address whole classes of problems in processing natural as well as formal languages.

Various domain-specific programming languages have been created that allow us to implement the theoretical problem using hopefully intuitive and minimal programming effort. These special-purpose languages usually introduce idiosyncratic notations of programs and are interpreted using some restricted model of computation. The motivation for such approaches may partly lie in the fact that, historically, computational resources were too limited compared to the requirements and complexity of the tasks being solved. Other motivations are theoretical given that finding a simple but accurate and yet generalizing model is the point of scientific abstraction.

There are also many approaches that do not resort to domain-specific programming. They, however, have to take care of the runtime performance and efficiency of the computational model themselves. It is up to the choice of the programming methods and the design style whether such models turn out to be pure, intuitive, adequate, complete, reusable, elegant, or not.

Let us now look at the most prominent types of computational approaches to morphology. Needless to say, this typology is not strictly exclusive in the sense that comprehensive morphological models and their applications can combine various distinct implementational aspects, discussed next.

1.3.1 Dictionary Lookup

Morphological parsing is a process by which word forms of a language are associated with corresponding linguistic descriptions. Morphological systems that specify these associations by merely enumerating them case by case do not offer any generalization means. Likewise for systems in which analyzing a word form is reduced to looking it up verbatim in word

lists, dictionaries, or databases, unless they are constructed by and kept in sync with more sophisticated models of the language.

In this context, a dictionary is understood as a data structure that directly enables obtaining some precomputed results, in our case word analyses. The data structure can be optimized for efficient lookup, and the results can be shared. Lookup operations are relatively simple and usually quick. Dictionaries can be implemented, for instance, as lists, binary search trees, tries, hash tables, and so on.

Because the set of associations between word forms and their desired descriptions is declared by plain enumeration, the coverage of the model is finite and the generative potential of the language is not exploited. Developing as well as verifying the association list is tedious, liable to errors, and likely inefficient and inaccurate unless the data are retrieved automatically from large and reliable linguistic resources.

Despite all that, an enumerative model is often sufficient for the given purpose, deals easily with exceptions, and can implement even complex morphology. For instance, dictionary-based approaches to Korean [35] depend on a large dictionary of all possible combinations of allomorphs and morphological alternations. These approaches do not allow development of reusable morphological rules, though [36].

The word list or dictionary-based approach has been used frequently in various ad hoc implementations for many languages. We could assume that with the availability of immense online data, extracting a high-coverage vocabulary of word forms is feasible these days [37]. The question remains how the associated annotations are constructed and how informative and accurate they are. References to the literature on the unsupervised learning and induction of morphology, which are methods resulting in structured and therefore nonenumerative models, are provided later in this chapter.

1.3.2 Finite-State Morphology

By finite-state morphological models, we mean those in which the specifications written by human programmers are directly compiled into finite-state transducers. The two most popular tools supporting this approach, which have been cited in literature and for which example implementations for multiple languages are available online, include XFST (Xerox Finite-State Tool) [9] and LexTools [11].⁵

Finite-state transducers are computational devices extending the power of finite-state automata. They consist of a finite set of nodes connected by directed edges labeled with pairs of input and output symbols. In such a network or graph, nodes are also called states, while edges are called arcs. Traversing the network from the set of initial states to the set of final states along the arcs is equivalent to reading the sequences of encountered input symbols and writing the sequences of corresponding output symbols.

The set of possible sequences accepted by the transducer defines the input language; the set of possible sequences emitted by the transducer defines the output language. For example, a finite-state transducer could translate the infinite regular language consisting of the words *vnuk*, *pravnuuk*, *prapravnuuk*, ... to the matching words in the infinite regular language defined by *grandson*, *great-grandson*, *great-great-grandson*, ...

5. See <http://www.fsmbook.com/> and <http://compling.ai.uiuc.edu/catms/> respectively.

The role of finite-state transducers is to capture and compute **regular relations** on sets [38, 9, 11].⁶ That is, transducers specify relations between the input and output languages. In fact, it is possible to invert the domain and the range of a relation, that is, exchange the input and the output. In finite-state computational morphology, it is common to refer to the input word forms as **surface strings** and to the output descriptions as **lexical strings**, if the transducer is used for morphological analysis, or vice versa, if it is used for morphological generation.

The linguistic descriptions we would like to give to the word forms and their components can be rather arbitrary and are obviously dependent on the language processed as well as on the morphological theory followed. In English, a finite-state transducer could analyze the surface string *children* into the lexical string *child* [+plural], for instance, or generate *women* from *woman* [+plural]. For other examples of possible input and output strings, consider Example 1–8 or Figure 1–1.

Relations on languages can also be viewed as functions. Let us have a relation \mathcal{R} , and let us denote by $[\Sigma]$ the set of all sequences over some set of symbols Σ , so that the domain and the range of \mathcal{R} are subsets of $[\Sigma]$. We can then consider \mathcal{R} as a function mapping an input string into a set of output strings, formally denoted by this type signature, where $[\Sigma]$ equals *String*:

$$\mathcal{R} :: [\Sigma] \rightarrow \{[\Sigma]\} \qquad \mathcal{R} :: \textit{String} \rightarrow \{\textit{String}\} \qquad (1.1)$$

Finite-state transducers have been studied extensively for their formal algebraic properties and have proven to be suitable models for miscellaneous problems [9]. Their applications encoding the surface rather than lexical string associations as **rewrite rules** of phonology and morphology have been around since the two-level morphology model [39], further presented in *Computational Approaches to Morphology and Syntax* [11] and *Morphology and Computation* [40].

Morphological operations and processes in human languages can, in the overwhelming number of cases and to a sufficient degree, be expressed in finite-state terms. Beesley and Karttunen [9] stress concatenation of transducers as the method for factoring surface and lexical languages into simpler models and propose a somewhat unsystematic **compile-replace** transducer operation for handling nonconcatenative phenomena in morphology. Roark and Sproat [11], however, argue that building morphological models in general using transducer composition, which is pure, is a more universal approach.

A theoretical limitation of finite-state models of morphology is the problem of capturing **reduplication** of words or their elements (e.g., to express plurality) found in several human languages. A formal language that contains only words of the form λ^{1+k} , where λ is some arbitrary sequence of symbols from an alphabet and $k \in \{1, 2, \dots\}$ is an arbitrary natural number indicating how many times λ is repeated after itself, is not a regular language, not even a context-free language. General reduplication of strings of unbounded length is thus not a regular-language operation. Coping with this problem in the framework of finite-state transducers is discussed by Roark and Sproat [11].

6. Regular relations and regular languages are restricted in their structure by the limited memory of the device (i.e., the finite set of configurations in which it can occur). Unlike with regular languages, intersection of regular relations can in general yield nonregular results [38].

Finite-state technology can be applied to the morphological modeling of isolating and agglutinative languages in a quite straightforward manner. Korean finite-state models are discussed by Kim et al. [41], Lee and Rim [42], and Han [43], to mention a few. For treatments of nonconcatenative morphology using finite-state frameworks, see especially Kay [44], Beesley [45], Kiraz [46], and Habash, Rambow, and Kiraz [47]. For comparison with finite-state models of the rich morphology of Czech, compare Skoumalová [48] and Sedláček and Smrž [49].

Implementing a refined finite-state morphological model requires careful fine-tuning of its lexicons, rewrite rules, and other components, while extending the code can lead to unexpected interactions in it, as noted by Oazer [50]. Convenient specification languages like those mentioned previously are needed because encoding the finite-state transducers directly would be extremely arduous, error prone, and unintelligible.

Finite-state tools are available in most general-purpose programming languages in the form of support for regular expression matching and substitution. While these may not be the ultimate choice for building full-fledged morphological analyzers or generators of a natural language, they are very suitable for developing tokenizers and morphological guessers capable of suggesting at least some structure for words that are formed correctly but cannot be identified with concrete lexemes during full morphological parsing [9].

1.3.3 Unification-Based Morphology

Unification-based approaches to morphology have been inspired by advances in various formal linguistic frameworks aiming at enabling complete grammatical descriptions of human languages, especially head-driven phrase structure grammar (HPSG) [51], and by development of languages for lexical knowledge representation, especially DATR [52]. The concepts and methods of these formalisms are often closely connected to those of logic programming. In the excellent thesis by Erjavec [53], the scientific context is discussed extensively and profoundly; refer also to the monographs by Carpenter [54] and Shieber [55].

In finite-state morphological models, both surface and lexical forms are by themselves unstructured strings of atomic symbols. In higher-level approaches, linguistic information is expressed by more appropriate data structures that can include complex values or can be recursively nested if needed. Morphological parsing \mathcal{P} thus associates linear forms ϕ with alternatives of structured content ψ , cf. (1.1):

$$\mathcal{P} :: \phi \rightarrow \{\psi\} \qquad \mathcal{P} :: \text{form} \rightarrow \{\text{content}\} \qquad (1.2)$$

Erjavec [53] argues that for morphological modeling, word forms are best captured by regular expressions, while the linguistic content is best described through **typed feature structures**. Feature structures can be viewed as directed acyclic graphs. A node in a feature structure comprises a set of attributes whose values can be feature structures again. Nodes are associated with types, and atomic values are attributeless nodes distinguished by their type. Instead of unique instances of values everywhere, references can be used to establish value instance identity. Feature structures are usually displayed as attribute-value matrices or as nested symbolic expressions.

Unification is the key operation by which feature structures can be merged into a more informative feature structure. Unification of feature structures can also fail, which means

that the information in them is mutually incompatible. Depending on the flavor of the processing logic, unification can be monotonic (i.e., information-preserving), or it can allow inheritance of default values and their overriding. In either case, information in a model can be efficiently shared and reused by means of inheritance hierarchies defined on the feature structure types.

Morphological models of this kind are typically formulated as logic programs, and unification is used to solve the system of constraints imposed by the model. Advantages of this approach include better abstraction possibilities for developing a morphological grammar as well as elimination of redundant information from it.

However, morphological models implemented in DATR can, under certain assumptions, be converted to finite-state machines and are thus formally equivalent to them in the range of morphological phenomena they can describe [11]. Interestingly, one-level phonology [56] formulating phonological constraints as logic expressions can be compiled into finite-state automata, which can then be intersected with morphological transducers to exclude any disturbing phonologically invalid surface strings [cf. 57, 53]

Unification-based models have been implemented for Russian [58], Czech [59], Slovene [53], Persian [60], Hebrew [61], Arabic [62, 63], and other languages. Some rely on DATR; some adopt, adapt, or develop other unification engines.

1.3.4 Functional Morphology

This group of morphological models includes not only the ones following the methodology of functional morphology [64], but even those related to it, such as morphological resource grammars of Grammatical Framework [65]. Functional morphology defines its models using principles of functional programming and type theory. It treats morphological operations and processes as pure mathematical functions and organizes the linguistic as well as abstract elements of a model into distinct types of values and type classes.

Though functional morphology is not limited to modeling particular types of morphologies in human languages, it is especially useful for fusional morphologies. Linguistic notions like paradigms, rules and exceptions, grammatical categories and parameters, lexemes, morphemes, and morphs can be represented intuitively and succinctly in this approach. Designing a morphological system in an accurate and elegant way is encouraged by the computational setting, which supports logical decoupling of subproblems and reinforces the semantic structure of a program by strong type checking.

Functional morphology implementations are intended to be reused as programming libraries capable of handling the complete morphology of a language and to be incorporated into various kinds of applications. Morphological parsing is just one usage of the system, the others being morphological generation, lexicon browsing, and so on. Next to parsing (1.2), we can describe inflection \mathcal{I} , derivation \mathcal{D} , and lookup \mathcal{L} as functions of these generic types:

$$\mathcal{I} :: \textit{lexeme} \rightarrow \{\textit{parameter}\} \rightarrow \{\textit{form}\} \quad (1.3)$$

$$\mathcal{D} :: \textit{lexeme} \rightarrow \{\textit{parameter}\} \rightarrow \{\textit{lexeme}\} \quad (1.4)$$

$$\mathcal{L} :: \textit{content} \rightarrow \{\textit{lexeme}\} \quad (1.5)$$

A functional morphology model can be compiled into finite-state transducers if needed, but can also be used interactively in an interpreted mode, for instance. Computation within a model may exploit lazy evaluation and employ alternative methods of efficient parsing, lookup, and so on [see 66, 12].

Many functional morphology implementations are embedded in a general-purpose programming language, which gives programmers more freedom with advanced programming techniques and allows them to develop full-featured, real-world applications for their models. The Zen toolkit for Sanskrit morphology [67, 68] is written in OCaml. It influenced the functional morphology framework [64] in Haskell, with which morphologies of Latin, Swedish, Spanish, Urdu [69], and other languages have been implemented.

In Haskell, in particular, developers can take advantage of its syntactic flexibility and design their own notation for the functional constructs that model the given problem. The notation then constitutes a so-called domain-specific embedded language, which makes programming even more fun. Figure 1–2 illustrates how the ElixirFM implementation of Arabic morphology [12, 17] captures the structure of words and defines the lexicon. Despite the entries being most informative, their format is simply similar to that found in printed dictionaries. Operators like `>`, `<`, `<<` and labels like `verb` are just infix functions; patterns and affixes like `FaCY`, `FCI`, `At` are data constructors.

> "d r y" < [<i>d r y</i> دري
FaCY	'verb'	["know", "notice"]	<i>faʿā</i>
'imperf'	FCI,		<i>fī</i>
FACY	'verb'	["flatter", "deceive"],	<i>fāʿā</i>
HaFCY	'verb'	["inform", "let know"],	<i>ʿafā</i>
IA > "'a" >> FCI <<< "ly"	'adj'	["agnostic"],	<i>lā-ʾa-fī-īy</i>
FiCAL < aT	'noun'	["knowledge", "knowing"],	<i>fīʾāl-ah</i>
MuFACY < aT	'noun'	["flattery"]	<i>mufāʿā-āt</i>
'plural'	MuFACY < At,		<i>fāʿī</i>
FACI	'adj'	["aware", "knowing"]]	
know, notice	I (<i>i</i>)	<i>darā</i> درى	دراية <i>dirāyah</i>
flatter, deceive	III	<i>dārā</i> دارى	مداراة <i>mudārāh</i>
inform, let know	IV	<i>ʾadrā</i> أدري	(مداريات <i>mudārayāt</i>)
agnostic		<i>lā-ʾadrīy</i> لأدري	دار <i>dārin</i>

Figure 1–2: Excerpt from the ElixirFM lexicon and a layout generated from it. The source code of entries nested under the *d r y* root is shown in monospace font. Note the custom notation and the economy yet informativeness of the declaration

Even without the options provided by general-purpose programming languages, functional morphology models achieve high levels of abstraction. Morphological grammars in Grammatical Framework [65] can be extended with descriptions of the syntax and semantics of a language. Grammatical Framework itself supports multilinguality, and models of more than a dozen languages are available in it as open-source software [70, 71].

Grammars in the OpenCCG project [72] can be viewed as functional models, too. Their formalism discerns declarations of features, categories, and families that provide type-system-like means for representing structured values and inheritance hierarchies on them. The grammars leverage heavily the functionality to define parametrized macros to minimize redundancy in the model and make required generalizations. Expansion of macros in the source code has effects similar to inlining of functions. The original text of the grammar is reduced to associations between word forms and their morphosyntactic and lexical properties.

1.3.5 Morphology Induction

We have focused on finding the structure of words in diverse languages supposing we know what we are looking for. We have not considered the problem of discovering and inducing word structure without the human insight (i.e., in an unsupervised or semi-supervised manner). The motivation for such approaches lies in the fact that for many languages, linguistic expertise might be unavailable or limited, and implementations adequate to a purpose may not exist at all. Automated acquisition of morphological and lexical information, even if not perfect, can be reused for bootstrapping and improving the classical morphological models, too.

Let us skim over the directions of research in this domain. In the studies by Hammarström [73] and Goldsmith [74], the literature on unsupervised learning of morphology is reviewed in detail. Hammarström divides the numerous approaches into three main groups. Some works compare and cluster words based on their similarity according to miscellaneous metrics [75, 76, 77, 78]; others try to identify the prominent features of word forms distinguishing them from the unrelated ones. Most of the published approaches cast morphology induction as the problem of word boundary and morpheme boundary detection, sometimes acquiring also lexicons and paradigms [79, 80, 81, 82, 83].⁷

There are several challenging issues about deducing word structure just from the forms and their context. They are caused by ambiguity [76] and irregularity [75] in morphology, as well as by orthographic and phonological alternations [85] and nonlinear morphological processes [86, 87].

In order to improve the chances of statistical inference, parallel learning of morphologies for multiple languages is proposed by Snyder and Barzilay [88], resulting in discovery of abstract morphemes. The discriminative log-linear model of Poon, Cherry, and Toutanova [89] enhances its generalization options by employing overlapping contextual features when making segmentation decisions [cf. 90].

7. Compare these with a semisupervised approach to word hyphenation [84].

1.4 Summary

In this chapter, we learned that morphology can be looked at from opposing viewpoints: one that tries to find the structural components from which words are built versus a more syntax-driven perspective wherein the functions of words are the focus of the study. Another distinction can be made between analytic and generative aspects of morphology or can consider man-made morphological frameworks versus systems for unsupervised induction of morphology. Yet other kinds of issues are raised about how well and how easily the morphological models can be implemented.

We described morphological parsing as the formal process recovering structured information from a linear sequence of symbols, where ambiguity is present and where multiple interpretations should be expected.

We explored interesting morphological phenomena in different types of languages and mentioned several hints in respect to multilingual processing and model development.

With Korean as a language where agglutination moderated by phonological rules is the dominant morphological process, we saw that a viable model of word decomposition can work at the morphemes level, regardless of whether they are lexical or grammatical.

In Czech and Arabic as fusional languages with intricate systems of inflectional and derivational parameters and lexically dependent word stem variation, such factorization is not useful. Morphology is better described via paradigms associating the possible forms of lexemes with their corresponding properties.

We discussed various options for implementing either of these models using modern programming techniques.

Acknowledgment

We would like to thank Petr Novák for his valuable comments on an earlier draft of this chapter.

Bibliography

- [1] M. Liberman, “Morphology.” Linguistics 001, Lecture 7, University of Pennsylvania, 2009. http://www.ling.upenn.edu/courses/Fall_2009/ling001/morphology.html.
- [2] M. Haspelmath, “The indeterminacy of word segmentation and the nature of morphology and syntax,” *Folia Linguistica*, vol. 45, 2011.
- [3] H. Kučera and W. N. Francis, *Computational Analysis of Present-Day American English*. Providence, RI: Brown University Press, 1967.
- [4] S. B. Cohen and N. A. Smith, “Joint morphological and syntactic disambiguation,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 208–217, 2007.

- [5] T. Nakagawa, “Chinese and Japanese word segmentation using word-level and character-level information,” in *Proceedings of 20th International Conference on Computational Linguistics*, pp. 466–472, 2004.
- [6] H. Shin and H. You, “Hybrid n -gram probability estimation in morphologically rich languages,” in *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, 2009.
- [7] D. Z. Hakkani-Tür, K. Oflazer, and G. Tür, “Statistical morphological disambiguation for agglutinative languages,” in *Proceedings of the 18th Conference on Computational Linguistics*, pp. 285–291, 2000.
- [8] G. T. Stump, *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge Studies in Linguistics, New York: Cambridge University Press, 2001.
- [9] K. R. Beesley and L. Karttunen, *Finite State Morphology*. CSLI Studies in Computational Linguistics, Stanford, CA: CSLI Publications, 2003.
- [10] M. Baerman, D. Brown, and G. G. Corbett, *The Syntax-Morphology Interface. A Study of Syncretism*. Cambridge Studies in Linguistics, New York: Cambridge University Press, 2006.
- [11] B. Roark and R. Sproat, *Computational Approaches to Morphology and Syntax*. Oxford Surveys in Syntax and Morphology, New York: Oxford University Press, 2007.
- [12] O. Smrž, “Functional Arabic morphology. Formal system and implementation,” PhD thesis, Charles University in Prague, 2007.
- [13] H. Eifring and R. Theil, *Linguistics for Students of Asian and African Languages*. Universitetet i Oslo, 2005.
- [14] B. Bickel and J. Nichols, “Fusion of selected inflectional formatives & exponence of selected inflectional formatives,” in *The World Atlas of Language Structures Online* (M. Haspelmath, M. S. Dryer, D. Gil, and B. Comrie, eds.), ch. 20 & 21, Munich: Max Planck Digital Library, 2008.
- [15] W. Fischer, *A Grammar of Classical Arabic*. Trans. Jonathan Rodgers. Yale Language Series, New Haven, CT: Yale University Press, 2002.
- [16] K. C. Ryding, *A Reference Grammar of Modern Standard Arabic*. New York: Cambridge University Press, 2005.
- [17] O. Smrž and V. Bielický, “ElixirFM.” Functional Arabic Morphology, SourceForge.net, 2010. <http://sourceforge.net/projects/elixir-fin/>.
- [18] T. Kamei, R. Kōno, and E. Chino, eds., *The Sanseido Encyclopedia of Linguistics, Volume 6 Terms* (in Japanese). Sanseido, 1996.
- [19] F. Karlsson, *Finnish Grammar*. Helsinki: Werner Söderström Osakenyhtiö, 1987.
- [20] J. Hajič and B. Hladká, “Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset,” in *Proceedings of COLING-ACL 1998*, pp. 483–490, 1998.

- [21] J. Hajič, “Morphological tagging: Data vs. dictionaries,” in *Proceedings of NAACL-ANLP 2000*, pp. 94–101, 2000.
- [22] N. Habash and O. Rambow, “Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop,” in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pp. 573–580, 2005.
- [23] N. A. Smith, D. A. Smith, and R. W. Tromble, “Context-based morphological disambiguation with random fields,” in *Proceedings of HLT/EMNLP 2005*, pp. 475–482, 2005.
- [24] J. Hajič, O. Smrž, T. Buckwalter, and H. Jin, “Feature-based tagger of approximations of functional Arabic morphology,” in *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pp. 53–64, 2005.
- [25] T. Buckwalter, “Issues in Arabic orthography and morphology analysis,” in *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pp. 31–34, 2004.
- [26] R. Nelken and S. M. Shieber, “Arabic diacritization using finite-state transducers,” in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pp. 79–86, 2005.
- [27] I. Zitouni, J. S. Sorensen, and R. Sarikaya, “Maximum entropy based restoration of Arabic diacritics,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 577–584, 2006.
- [28] N. Habash and O. Rambow, “Arabic diacritization through full morphological tagging,” in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pp. 53–56, 2007.
- [29] G. Huet, “Lexicon-directed segmentation and tagging of Sanskrit,” in *Proceedings of the XIIth World Sanskrit Conference*, pp. 307–325, 2003.
- [30] G. Huet, “Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor,” in *Sanskrit Computational Linguistics: First and Second International Symposia* (G. Huet, A. Kulkarni, and P. Scharf, eds.), vol. 5402 of *LNAI*, pp. 162–199, Berlin: Springer, 2009.
- [31] F. Katamba and J. Stonham, *Morphology*. Basingstoke: Palgrave Macmillan, 2006.
- [32] L. Bauer, *Morphological Productivity*, Cambridge Studies in Linguistics. New York: Cambridge University Press, 2001.
- [33] R. H. Baayen, *Word Frequency Distributions*, Text, Speech and Language Technology. Boston: Kluwer Academic Publishers, 2001.
- [34] A. Kilgarriff, “Googleology is bad science,” *Computational Linguistics*, vol. 33, no. 1, pp. 147–151, 2007.

- [35] H.-C. Kwon and Y.-S. Chae, "A dictionary-based morphological analysis," in *Proceedings of Natural Language Processing Pacific Rim Symposium*, pp. 178–185, 1991.
- [36] D.-B. Kim, K.-S. Choi, and K.-H. Lee, "A computational model of Korean morphological analysis: A prediction-based approach," *Journal of East Asian Linguistics*, vol. 5, no. 2, pp. 183–215, 1996.
- [37] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.
- [38] R. M. Kaplan and M. Kay, "Regular models of phonological rule systems," *Computational Linguistics*, vol. 20, no. 3, pp. 331–378, 1994.
- [39] K. Koskeniemi, "Two-level morphology: A general computational model for word form recognition and production," PhD thesis, University of Helsinki, 1983.
- [40] R. Sproat, *Morphology and Computation*. ACL–MIT Press Series in Natural Language Processing. Cambridge, MA: MIT Press, 1992.
- [41] D.-B. Kim, S.-J. Lee, K.-S. Choi, and G.-C. Kim, "A two-level morphological analysis of Korean," in *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 535–539, 1994.
- [42] S.-Z. Lee and H.-C. Rim, "Korean morphology with elementary two-level rules and rule features," in *Proceedings of the Pacific Association for Computational Linguistics*, pp. 182–187, 1997.
- [43] N.-R. Han, "Klex: A finite-state transducer lexicon of Korean," in *Finite-state Methods and Natural Language Processing: 5th International Workshop, FSMNLP 2005*, pp. 67–77, Springer, 2006.
- [44] M. Kay, "Nonconcatenative finite-state morphology," in *Proceedings of the Third Conference of the European Chapter of the ACL (EACL-87)*, pp. 2–10, ACL, 1987.
- [45] K. R. Beesley, "Arabic morphology using only finite-state operations," in *COLING-ACL'98 Proceedings of the Workshop on Computational Approaches to Semitic languages*, pp. 50–57, 1998.
- [46] G. A. Kiraz, *Computational Nonlinear Morphology with Emphasis on Semitic Languages*. Studies in Natural Language Processing, Cambridge: Cambridge University Press, 2001.
- [47] N. Habash, O. Rambow, and G. Kiraz, "Morphological analysis and generation for Arabic dialects," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pp. 17–24, 2005.
- [48] H. Skoumalová, "A Czech morphological lexicon," in *Proceedings of the Third Meeting of the ACL Special Interest Group in Computational Phonology*, pp. 41–47, 1997.
- [49] R. Sedláček and P. Smrž, "A new Czech morphological analyser ajka," in *Text, Speech and Dialogue*, vol. 2166, pp. 100–107, Berlin: Springer, 2001.

- [50] K. Oflazer, “Computational morphology.” ESSLLI 2006 European Summer School in Logic, Language, and Information, 2006.
- [51] C. Pollard and I. A. Sag, *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press, 1994.
- [52] R. Evans and G. Gazdar, “DATR: A language for lexical knowledge representation,” *Computational Linguistics*, vol. 22, no. 2, pp. 167–216, 1996.
- [53] T. Erjavec, “Unification, inheritance, and paradigms in the morphology of natural languages,” PhD thesis, University of Ljubljana, 1996.
- [54] B. Carpenter, *The Logic of Typed Feature Structures*. Cambridge Tracts in Theoretical Computer Science 32, New York: Cambridge University Press, 1992.
- [55] S. M. Shieber, *Constraint-Based Grammar Formalisms: Parsing and Type Inference for Natural and Computer Languages*. Cambridge, MA: MIT Press, 1992.
- [56] S. Bird and T. M. Ellison, “One-level phonology: Autosegmental representations and rules as finite automata,” *Computational Linguistics*, vol. 20, no. 1, pp. 55–90, 1994.
- [57] S. Bird and P. Blackburn, “A logical approach to Arabic phonology,” in *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 89–94, 1991.
- [58] G. G. Corbett and N. M. Fraser, “Network morphology: A DATR account of Russian nominal inflection,” *Journal of Linguistics*, vol. 29, pp. 113–142, 1993.
- [59] J. Hajič, “Unification morphology grammar. Software system for multilanguage morphological analysis,” PhD thesis, Charles University in Prague, 1994.
- [60] K. Megerdooian, “Unification-based Persian morphology,” in *Proceedings of CICLing 2000*, 2000.
- [61] R. Finkel and G. Stump, “Generating Hebrew verb morphology by default inheritance hierarchies,” in *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pp. 9–18, 2002.
- [62] S. R. Al-Najem, “Inheritance-based approach to Arabic verbal root-and-pattern morphology,” in *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (A. Soudi, A. van den Bosch, and G. Neumann, eds.), vol. 38, pp. 67–88, Berlin: Springer, 2007.
- [63] S. Köprü and J. Miller, “A unification based approach to the morphological analysis and generation of Arabic,” in *CAASL-3: Third Workshop on Computational Approaches to Arabic Script-based Languages*, 2009.
- [64] M. Forsberg and A. Ranta, “Functional morphology,” in *Proceedings of the 9th ACM SIGPLAN International Conference on Functional Programming, ICFP 2004*, pp. 213–223, 2004.
- [65] A. Ranta, “Grammatical Framework: A type-theoretical grammar formalism,” *Journal of Functional Programming*, vol. 14, no. 2, pp. 145–189, 2004.

- [66] P. Ljunglöf, “Pure functional parsing. An advanced tutorial,” Licenciate thesis, Göteborg University & Chalmers University of Technology, 2002.
- [67] G. Huet, “The Zen computational linguistics toolkit,” ESSLLI 2002 European Summer School in Logic, Language, and Information, 2002.
- [68] G. Huet, “A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger,” *Journal of Functional Programming*, vol. 15, no. 4, pp. 573–614, 2005.
- [69] M. Humayoun, H. Hammarström, and A. Ranta, “Urdu morphology, orthography and lexicon extraction,” in *CAASL-2: Second Workshop on Computational Approaches to Arabic Script-based Languages*, pp. 59–66, 2007.
- [70] A. Dada and A. Ranta, “Implementing an open source Arabic resource grammar in GF,” in *Perspectives on Arabic Linguistics* (M. A. Mughazy, ed.), vol. XX, pp. 209–231, John Benjamins, 2007.
- [71] A. Ranta, “Grammatical Framework.” Programming Language for Multilingual Grammar Applications, <http://www.grammaticalframework.org/>, 2010.
- [72] J. Baldridge, S. Chatterjee, A. Palmer, and B. Wing, “DotCCG and VisCCG: Wiki and programming paradigms for improved grammar engineering with OpenCCG,” in *Proceedings of the Workshop on Grammar Engineering Across Frameworks*, 2007.
- [73] H. Hammarström, “Unsupervised learning of morphology and the languages of the world,” PhD thesis, Chalmers University of Technology and University of Gothenburg, 2009.
- [74] J. A. Goldsmith, “Segmentation and morphology,” in *Computational Linguistics and Natural Language Processing Handbook* (A. Clark, C. Fox, and S. Lappin, eds.), pp. 364–393, Chichester: Wiley-Blackwell, 2010.
- [75] D. Yarowsky and R. Wicentowski, “Minimally supervised morphological analysis by multimodal alignment,” in *Proceedings of the 38th Meeting of the Association for Computational Linguistics*, pp. 207–216, 2000.
- [76] P. Schone and D. Jurafsky, “Knowledge-free induction of inflectional morphologies,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 183–191, 2001.
- [77] S. Neuvel and S. A. Fulop, “Unsupervised learning of morphology without morphemes,” in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pp. 31–40, 2002.
- [78] N. Hathout, “Acquisition of the morphological structure of the lexicon based on lexical similarity and formal analogy,” in *Coling 2008: Proceedings of the 3rd Textgraphs Workshop on Graph-based Algorithms for Natural Language Processing*, pp. 1–8, 2008.
- [79] J. Goldsmith, “Unsupervised learning of the morphology of a natural language,” *Computational Linguistics*, vol. 27, no. 2, pp. 153–198, 2001.

- [80] H. Johnson and J. Martin, “Unsupervised learning of morphology for English and Inuktitut,” in *Companion Volume of the Proceedings of the Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics 2003: Short Papers*, pp. 43–45, 2003.
- [81] M. Creutz and K. Lagus, “Induction of a simple morphology for highly-inflecting languages,” in *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology*, pp. 43–51, 2004.
- [82] M. Creutz and K. Lagus, “Unsupervised models for morpheme segmentation and morphology learning,” *ACM Transactions on Speech and Language Processing*, vol. 4, no. 1, pp. 1–34, 2007.
- [83] C. Monson, J. Carbonell, A. Lavie, and L. Levin, “ParaMor: Minimally supervised induction of paradigm structure and morphological analysis,” in *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, pp. 117–125, 2007.
- [84] F. M. Liang, “Word Hy-phen-a-tion by Com-put-er,” PhD thesis, Stanford University, 1983.
- [85] V. Demberg, “A language-independent unsupervised model for morphological segmentation,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 920–927, 2007.
- [86] A. Clark, “Supervised and unsupervised learning of Arabic morphology,” in *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (A. Soudi, A. van den Bosch, and G. Neumann, eds.), vol. 38, pp. 181–200, Berlin: Springer, 2007.
- [87] A. Xanthos, *Apprentissage automatique de la morphologie: le cas des structures racine-schème*. Sciences pour la communication, Bern: Peter Lang, 2008.
- [88] B. Snyder and R. Barzilay, “Unsupervised multilingual learning for morphological segmentation,” in *Proceedings of ACL-08: HLT*, pp. 737–745, 2008.
- [89] H. Poon, C. Cherry, and K. Toutanova, “Unsupervised morphological segmentation with log-linear models,” in *Proceedings of Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 209–217, 2009.
- [90] S. Della Pietra, V. Della Pietra, and J. Lafferty, “Inducing features of random fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

Index

- . (period), sentence segmentation markers, 30
- “” (Quotation marks), sentence segmentation markers, 30
- ! (Exclamation point), as sentence segmentation marker, 30
- ? (Question mark), sentence segmentation markers, 30
- 80/20 rule (vital few), 14

- a priori models, in document retrieval, 377
- Abbreviations, punctuation marks in, 30
- Absity parser, rule-based semantic parsing, 122
- Abstracts
 - in automatic summarization, 397
 - defined, 400
- Accumulative vector space model, for document retrieval, 374–375
- Accuracy, in QA, 462
- ACE. See Automatic content extraction (ACE)
- Acquis corpus
 - for evaluating IR systems, 390
 - for machine translation, 358
- Adequacy, of translation, 334
- Adjunctive arguments, PropBank verb predicates, 119–120
- AER (Alignment-error rate), 343
- AEs (Analysis engines), UIMA, 527
- Agglutinative languages
 - finite-state technology applied to, 18
 - linear decomposition of words, 192
 - morphological typology and, 7
 - parsing issues related to morphology, 90–91
- Aggregate processor, combining NLP engines, 523
- Aggregation architectures, for NLP. See also Natural language processing (NLP), combining engines for
 - GATE, 529–530
 - InfoSphere Streams, 530–531
 - overview of, 527
 - UIMA, 527–529
- Aggregation models, for MLIR, 385
- Agreement feature, of coreference models, 301
- Air Travel Information System (ATIS)
 - as resource for meaning representation, 148
 - rule-based systems for semantic parsing, 150
 - supervised systems for semantic parsing, 150–151
- Algorithms. See by individual types
- Alignment-error rate (AER), 343
- Alignment, in RTE
 - implementing, 233–236
 - latent alignment inference, 247–248
 - learning alignment independently of entailment, 244–245
 - leveraging multiple alignments, 245
 - modeling, 226
- Allomorphs, 6
- “almost-parsing” language model, 181
- Ambiguity
 - disambiguation problem in morphology, 91
 - in interpretation of expressions, 10–13
 - issues with morphology induction, 21
 - PCFGs and, 80–83
 - resolution in parsing, 80
 - sentence segmentation markers and, 30
 - structural, 99
 - in syntactic analysis, 61
 - types of, 8
 - word sense and. See Disambiguation systems, word sense
- Analysis engines (AEs), UIMA, 527
- Analysis, in RTE framework
 - annotators, 219
 - improving, 248–249
 - multiview representation of, 220–222
 - overview of, 220
- Analysis stage, of summarization system
 - building a summarization system and, 421
 - overview of, 400

- Anaphora resolution. See also Coreference resolution
 - automatic summarization and, 398
 - cohesion of, 401
 - multilingual automatic summarization and, 410
 - QA architectures and, 438–439
 - zero anaphora resolution, 249, 444
- Anchored speech recognition, 490
- Anchors, in SSTK, 246
- Annotation/annotation guidelines
 - entity detection and, 293
 - in GALE, 478
 - Penn Treebank and, 87–88
 - phrase structure trees and, 68–69
 - QA architectures and, 439–440
 - in RTE, 219, 222–224
 - snippet processing and, 485
 - for treebanks, 62
 - of utterances based on rule-based grammars, 502–503
 - of utterances in spoken dialog systems, 513
- Answers, in QA
 - candidate answer extraction. See Candidate answer extraction, in QA
 - candidate answer generation. See Candidate answer generation, in QA
 - evaluating correctness of, 461–462
 - scores for, 450–453, 458–459
 - scoring component for, 435
 - type classification of, 440–442
- Arabic
 - ambiguity in, 11–12
 - corpora for relation extraction, 317
 - distillation, 479, 490–491
 - EDT and, 286
 - ElixirFM lexicon, 20
 - encoding and script, 368
 - English-to-Arabic machine translation, 114
 - as fusional language, 8
 - GALE IOD and, 532, 534–536
 - IR and, 371
 - irregularity in, 8–9
 - language modeling, 189–191, 193
 - mention detection experiments, 294–296
 - morphemes in, 6
 - morphological analysis of, 191
 - multilingual issues in predicate-argument structures, 146–147
 - polarity analysis of words and phrases, 269
 - productivity/creativity in, 15
 - regional dialects not in written form, 195
 - RTE in, 218
 - stem-matching features for capturing morphological similarities, 301
 - TALES case study, 538
 - tokens in, 4
 - translingual summarization, 398–399, 424–426
 - unification-based models, 19
- Architectures
 - aggregation architectures for NLP, 527–529
 - for question answering (QA), 435–437
 - of spoken dialog systems, 505
 - system architectures for distillation, 488
 - system architectures for semantic parsing, 101–102
 - types of EDT architectures, 286–287
- Arguments
 - consistency of argument identification, 323
 - event extraction and, 321–322
 - in GALE distillation initiative, 475
 - in RTE systems, 220
- Arguments, predicate-argument recognition
 - argument sequence information, 137–138
 - classification and identification, 139–140
 - core and adjunctive, 119
 - disallowing overlaps, 137
 - discontiguous, 121
 - identification and classification, 123
 - noun arguments, 144–146
- ART (artifact) relation class, 312
- ASCII
 - as encoding scheme, 368
 - parsing issues related, 89
- Asian Federation of Natural Language Processing, 218
- Asian languages. See also by individual Asian languages
 - multilingual IR and, 366, 390
 - QA and, 434, 437, 455, 460–461, 466
- Ask.com, 435
- ASR (automatic speech recognition)
 - sentence boundary annotation, 29
 - sentence segmentation markers, 31
- ASSERT (Automatic Statistical SEMantic Role Tagger), 147, 447

- ATIS. See Air Travel Information System (ATIS)
- Atomic events, summarization and, 418
- Attribute features, in coreference models, 301
- Automatic content extraction (ACE)
 - coreference resolution experiments, 302–303
 - event extraction and, 320–321
 - mention detection and, 287, 294
 - relation extraction and, 311–312
 - in Rosetta Consortium distillation system, 480–481
- Automatic speech recognition (ASR)
 - sentence boundary annotation, 29
 - sentence segmentation markers, 31
- Automatic Statistical SEmantic Role Tagger (ASSERT), 147, 447
- Automatic summarization
 - bibliography, 427–432
 - coherence and cohesion in, 401–404
 - extraction and modification processes in, 399–400
 - graph-based approaches, 401
 - history of, 398–399
 - introduction to, 397–398
 - learning how to summarize, 406–409
 - LexPageRank, 406
 - multilingual. See Multilingual automatic summarization
 - stages of, 400
 - summary, 426–427
 - surface-based features used in, 400–401
 - TextRank, 404–406
- Automatic Summary Evaluation based on n -gram graphs (AutoSummENG), 419–420

- Babel Fish
 - crosslingual question answering and, 455
 - Systran, 331
- Backend services, of spoken dialog system, 500
- Backoff smoothing techniques
 - generalized backoff strategy, 183–184
 - in language model estimation, 172
 - nonnormalized form, 175
 - parallel backoff, 184
- Backus-Naur form, of context-free grammar, 59
- BananaSplit, IR preprocessing and, 392

- Base phrase chunks, 132–133
- BASEBALL system, in history of QA systems, 434
- Basic Elements (BE)
 - automatic evaluation of summarization, 417–419
 - metrics in, 420
- Bayes rule, for sentence or topic segmentation, 39–40
- Bayes theorem, maximum-likelihood estimation and, 376
- Bayesian parameter estimation, 173–174
- Bayesian topic-based language models, 186–187
- BBN, event extraction and, 322
- BE (Basic Elements)
 - automatic evaluation of summarization, 417–419
 - metrics in, 420
- BE with Transformations for Evaluation (BEwTE), 419–420
- Beam search
 - machine translation and, 346
 - reducing search space using, 290–291
- Bell tree, for coreference resolution, 297–298
- Bengali. See Indian languages
- Berkeley word aligner, in machine translation, 357
- Bibliographic summaries, in automatic summarization, 397
- Bilingual latent semantic analysis (bLSA), 197–198
- Binary classifier, in event matching, 323–324
- Binary conditional model, for probability of mention links, 297–300
- BLEU
 - machine translation metrics, 334, 336
 - mention detection experiments and, 295
 - ROUGE compared with, 415–416
- Block comparison method, for topic segmentation, 38
- bLSA (bilingual latent semantic analysis), 197–198
- BLUE (Boeing Language Understanding Engine), 242–244
- BM25 model, in document retrieval, 375
- BNC (British National Corpus), 118
- Boeing Language Understanding Engine (BLUE), 242–244

- Boolean models
 - for document representation in monolingual IR, 372
 - for document retrieval, 374
- Boolean named entity flags, in PSG, 126
- Bootstrapping
 - building subjectivity lexicons, 266–267
 - corpus-based approach to subjectivity and sentiment analysis, 269
 - dictionary-based approach to subjectivity and sentiment analysis, 273
 - ranking approaches to subjectivity and sentiment analysis, 275–276
 - semisupervised approach to relation extraction, 318
- Boundary classification problems
 - overview of, 33
 - sentence boundaries. See Sentence boundary detection
 - topic boundaries. See Topic segmentation
- British National Corpus (BNC), 118
- Brown Corpus, as resource for semantic parsing, 104
- Buckwalter Morphological Analyzer, 191

- C-ASSERT, software programs for semantic role labeling, 147
- Call-flow
 - localization of, 514
 - strategy of dialog manager, 504
 - voice user interface (VUI) and, 505–506
- Call routing, natural language and, 510
- Canadian Hansards
 - corpora for IR, 391
 - corpora for machine translation, 358
- Candidate answer extraction, in QA
 - answer scores, 450–453
 - combining evidence, 453–454
 - structural matching, 446–448
 - from structured sources, 449–450
 - surface patterns, 448–449
 - type-based, 446
 - from unstructured sources, 445
- Candidate answer generation, in QA
 - components in QA architectures, 435
 - overview of, 443
- Candidate boundaries, processing stages of segmentation tasks, 48

- Canonization, deferred in RTE multiview representation, 222
- Capitalization (Uppercase), sentence segmentation markers, 30
- CAS (Common analysis structure), UIMA, 527, 536
- Cascading systems, types of EDT architectures, 286–287
- Case
 - parsing issues related to, 88
 - sentence segmentation markers, 30
- Catalan, 109
- Categorical ambiguity, word sense and, 104
- Cause-and-effect relations, causal reasoning and, 250
- CCG (Combinatory Categorical Grammar), 129–130
- CFGs. See Context-free grammar (CFGs)
- Character *n*-gram models, 370
- Chart decoding, tree-based models for machine translation, 351–352
- Chart parsing, worst-case parsing algorithm for CFGs, 74–79
- Charts, IXIR distillation system, 488–489
- CHILL (Constructive Heuristics Induction for Language Learning), 151
- Chinese
 - anaphora frequency in, 444
 - challenges of sentence and topic segmentation, 30
 - corpora for relation extraction, 317
 - corpus-based approach to subjectivity and sentiment analysis, 274–275
 - crosslingual language modeling, 197–198
 - data sets related to summarization, 424–426
 - dictionary-based approach to subjectivity and sentiment analysis, 272–273
 - distillation, 479, 490–491
 - EDT and, 286
 - HowNet lexicon for, 105
 - human assessment of word meaning, 333
 - IR and, 366, 390
 - isolating (analytic) languages, 7
 - as isolating or analytic language, 7
 - language modeling in without word segmentation, 193–194
 - lingPipe for word segmentation, 423
 - machine translation and, 322, 354, 358
 - mention detection experiments, 294–296

- multilingual issues in predicate-argument structures, 146–147
- phrase structure treebank, 70
- polarity analysis of words and phrases, 269
- preprocessing best practices in IR, 372
- QA and, 461, 464
- QA architectures and, 437–438
- resources for semantic parsing, 122
- RTE in, 218
- scripts not using whitespace, 369
- subjectivity and sentiment analysis, 259–260
- TALES case study, 538
- translingual summarization, 399, 410
- word segmentation and parsing, 89–90
- word segmentation in, 4–5
- word sense annotation in, 104
- Chomsky, Noam, 13, 98–99
- Chunk-based systems, 132–133
- Chunks
 - defined, 292
 - meaning chunks in semantic parsing, 97
- CIDR algorithm, for multilingual summarization, 411
- Citations
 - evaluation in distillation, 493
 - in GALE distillation initiative, 477
- CKY algorithm, worst-case parsing for CFGs, 76–78
- Class-based language models, 178–179
- Classes
 - language modeling using morphological categories, 193
 - of relations, 311
- Classification
 - of arguments, 123, 139–140
 - data-driven, 287–289
 - dynamic class context in PSG, 128
 - event extraction and, 321–322
 - overcoming independence assumption, 137–138
 - paradigms, 133–137
 - problems related to sentence boundaries. See Sentence boundary detection
 - problems related to topic boundaries. See Topic segmentation
 - relation extraction and, 312–316
- Classification tag lattice (trellis), searching for mentions, 289
- Classifiers
 - in event matching, 323–324
 - localization of grammars and, 516
 - maximum entropy classifiers, 37, 39–40
 - in mention detection, 292–293
 - pipeline of, 321
 - in relation extraction, 313, 316–317
 - in subjectivity and sentiment analysis, 270–272, 274
 - Type classifier in QA systems, 440–442
 - in word disambiguation, 110
- CLASSIFY function, 313
- ClearTK tool, for building summarization system, 423
- CLIR. See Crosslingual information retrieval (CLIR)
- Clitics
 - Czech example, 5
 - defined, 4
- Co-occurrence, of words between languages, 337–338
- Coarse to fine parsing, 77–78
- Code switchers
 - impact on sentence segmentation, 31
 - multilingual language modeling and, 195–196
- COGEX, for answer scores in QA, 451
- Coherence, sentence-sentence connections and, 402
- Cohesion, anaphora resolution and, 401–402
- Collection language, in CLIR, 365
- Combination hypothesis, combining classifiers to boost performance, 293
- Combinatory Categorical Grammar (CCG), 129–130
- Common analysis structure (CAS), UIMA, 527, 536
- Communicator program, for meaning representation, 148–150
- Comparators, RTE, 219, 222–223
- Competence vs. performance, Chomsky on, 13
- Compile/replace transducer (Beesley and Karttunen), 17
- Componentization of design, for NLP aggregation, 524–525
- Components of words
 - lexemes, 5
 - morphemes, 5–7
 - morphological typology and, 7–8

- Compound slitting
 - BananaSplit tool, 392
 - normalization for fusional languages, 371
- Computational efficiency
 - desired attributes of NLP aggregation, 525–526
 - in GALE IOD, 537
 - in GATE, 530
 - in InfoSphere Streams, 530–531
 - in UIMA, 528
- Computational Natural Language Learning (CoNLL), 132
- Concatenative languages, 8
- Concept space, interlingual document representations, 381
- Conceptual density, as measure of semantic similarity, 112
- Conditional probability, MaxEnt formula for, 316
- Conditional random fields (CRFs)
 - in discriminative parsing model, 84
 - machine learning and, 342
 - measuring token frequency, 369
 - mention detection and, 287
 - relation extraction and, 316
 - sentence or topic segmentation and, 39–40
- Confidence weighted score (CWS), in QA, 463
- CoNLL (Computational Natural Language Learning), 132
- Constituents
 - atomic events and, 418
 - in PSG, 127
- Constituents, in RTE
 - comparing annotation constituents, 222–224
 - multiview representation of analysis and, 220
 - numerical quantities (NUM), 221, 233
- Constraint-based language models, 177
- Constructive Heuristics Induction for Language Learning (CHILL), 151
- Content Analysis Toolkit (Tika), for preprocessing IR documents, 392
- Content word, in PSG, 125–126
- Context, as measure of semantic similarity, 112
- Context-dependent process, in GALE IOD, 536–537
- Context features, of Rosetta Consortium distillation system, 486
- Context-free grammar (CFGs)
 - for analysis of natural language syntax, 60–61
 - dependency graphs in syntax analysis, 65–67
 - rules of syntax, 59
 - shift-reduce parsing, 72–73
 - worst-case parsing algorithm, 74–78
- Contextual subjectivity analysis, 261
- Contradiction, in textual entailment, 211
- Conversational speech, sentence segmentation in, 31
- Core arguments, PropBank verb predicates, 119
- Coreference resolution. See also Anaphora resolution
 - automatic summarization and, 398
 - Bell tree for, 297–298
 - experiments in, 302–303
 - information extraction and, 100, 285–286
 - MaxEnt model applied to, 300–301
 - models for, 298–300
 - overview of, 295–296
 - as relation extraction system, 311
 - in RTE, 212, 227
- Corpora
 - for distillation, 480–483
 - for document-level annotations, 274
 - Europarl (European Parliament), 295, 345
 - for IR systems, 390–391
 - for machine translation (MT), 358
 - for relation extraction, 317
 - for semantic parsing, 104–105
 - for sentence-level annotations, 271–272
 - for subjectivity and sentiment analysis, 262–263, 274–275
 - for summarization, 406, 425
 - for word/phrase-level annotations, 267–269
- Coverage rate criteria, in language model evaluation, 170
- Cranfield paradigm, 387
- Creativity/productivity, and the unknown word problem, 13–15
- CRFs. See Conditional random fields (CRFs)
- Cross-Language Evaluation Forum (CLEF)
 - applying to RTE to non-English languages, 218
 - IR and, 377, 390
 - QA and, 434, 454, 460–464

- Cross-language mention propagation, 293, 295
- Cross-lingual projections, 275
- Crossdocument coreference (XDC), in Rosetta Consortium distillation system, 482–483
- Crossdocument Structure Theory Bank (CSTBank), 425
- Crossdocument structure theory (CST), 425
- Crosslingual distillation, 490–491
- Crosslingual information retrieval (CLIR)
 - best practices, 382
 - interlingual document representations, 381–382
 - machine translation, 380–381
 - overview of, 365, 378
 - translation-based approaches, 378–380
- Crosslingual language modeling, 196–198
- Crosslingual question answering, 454–455
- Crosslingual summarization, 398
- CST (Crossdocument structure theory), 425
- CSTBank (Crossdocument Structure Theory Bank), 425
- Cube pruning, decoding phrase-based models, 347–348
- CWS (Confidence weighted score), in QA, 463
- Cyrillic alphabet, 371
- Czech
 - ambiguity in, 11–13
 - dependency graphs in syntax analysis, 62–65
 - dependency parsing in, 79
 - finite-state models, 18
 - as fusional language, 8
 - language modeling, 193
 - morphological richness of, 355
 - negation indicated by inflection, 5
 - parsing issues related to morphology, 91
 - productivity/creativity in, 14–15
 - syntactic features used in sentence and topic segmentation, 43
 - unification-based models, 19
- DAMSL (Dialog Act Markup in Several Layers), 31
- Data-driven
 - machine translation, 331
 - mention detection, 287–289
- Data formats, challenges in NLP aggregation, 524
- Data-manipulation capabilities
 - desired attributes of NLP aggregation, 526
 - in GATE, 530
 - in InfoSphere Streams, 531
 - in UIMA, 528–529
- Data reorganization, speech-to-text (STT) and, 535–536
- Data sets
 - for evaluating IR systems, 389–391
 - for multilingual automatic summarization, 425–426
- Data types
 - GALE Type System (GTS), 534–535
 - usage conventions for NLP aggregation, 540–541
- Databases
 - of entity relations and events, 309–310
 - relational, 449
- DATR, unification-based morphology and, 18–19
- DBpedia, 449
- de Saussure, Ferdinand, 13
- Decision trees, for sentence or topic segmentation, 39–40
- Decoding phrase-based models
 - cube pruning approach, 347–348
 - overview of, 345–347
- Deep representation, in semantic interpretation, 101
- Deep semantic parsing
 - coverage in, 102
 - overview of, 98
- Defense Advanced Research Projects Agency (DARPA)
 - GALE distillation initiative, 475–476
 - GALE IOD case study. See Interoperability Demo (IOD), GALE case study
 - Topic Detection and Tracking (TDT) program, 32–33
- Definitional questions, QA and, 433
- Deletions metrics, machine translation, 335
- Dependencies
 - global similarity in RTE and, 247
 - high-level features in event matching, 324–326
- Dependency graphs
 - phrase structure trees compared with, 69–70
 - in syntactic analysis, 63–67
 - in treebank construction, 62

- Dependency parsing
 - implementing RTE and, 227
 - Minipar and Stanford Parser, 456
 - MST algorithm for, 79–80
 - shift-reduce parsing algorithm for, 73
 - structural matching and, 447
 - tree edit distance based on, 240–241
 - worst-case parsing algorithm for CFGs, 78
- Dependency trees
 - non projective, 65–67
 - overview of, 130–132
 - patterns used in relation extraction, 318
 - projective, 64–65
- Derivation, parsing and, 71–72
- Devanagari, preprocessing best practices in IR, 371
- Dialog Act Markup in Several Layers (DAMSL), 31
- Dialog manager
 - directing speech generation, 499–500
 - overview of, 504–505
- Dialog module (DM)
 - call-flow localization and, 514
 - voice user interface and, 507–508
- Dialogs
 - forms of, 509–510
 - rules of, 499–500
- Dictionary-based approach, in subjectivity and sentiment analysis
 - document-level annotations, 272–273
 - sentence-level annotations, 270–271
 - word/phrase-level annotations, 264–267
- Dictionary-based morphology, 15–16
- Dictionary-based translations
 - applying to CLIR, 380
 - crosslingual modeling and, 197
- Directed dialogs, 509
- Directed graphs, 79–80
- Dirichlet distribution
 - Hierarchical Dirichlet process (HDP), 187
 - language models and, 174
 - Latent Dirichlet allocation (LDA) model, 186
- DIRT (Discovery of inference rules from text), 242
- Disambiguation systems, word sense
 - overview of, 105
 - rule-based, 105–109
 - semantic parsing and, 152–153
 - semi-supervised, 114–116
 - software programs for, 116–117
 - supervised, 109–112
 - unsupervised, 112–114
- Discontiguous arguments, PropBank verb predicates, 121
- Discourse commitments (beliefs), RTE system based on, 239–240
- Discourse connectives, relating sentences by, 29
- Discourse features
 - relating sentences by discourse connectives, 29
 - in sentence and topic segmentation, 44
- Discourse segmentation. *See* Topic segmentation
- Discourse structure
 - automatic summarization and, 398, 410
 - RTE applications and, 249
- Discovery of inference rules from text (DIRT), 242
- Discriminative language models
 - modeling using morphological categories, 192–193
 - modeling without word segmentation, 194
 - overview of, 179–180
- Discriminative local classification methods, for sentence/topic boundary detection, 36–38
- Discriminative parsing models
 - morphological information in, 91–92
 - overview of, 84–87
- Discriminative sequence classification methods
 - complexity of, 40–41
 - overview of, 34
 - performance of, 41
 - for sentence/topic boundary detection, 38–39
- Distance-based reordering model, in machine translation, 344
- Distance, features of coreference models, 301
- Distillation
 - bibliography, 495–497
 - crosslingual, 490–491
 - document and corpus preparation, 480–483
 - evaluation and metrics, 491–494
 - example, 476–477
 - indexing and, 483

- introduction to, 475–476
- multimodal, 490
- query answers and, 483–487
- redundancy reduction, 489–490
- relevance and redundancy and, 477–479
- relevance detection, 488–489
- Rosetta Consortium system, 479–480
- summary, 495
- system architectures for, 488
- DM (Dialog module)
 - call-flow localization and, 514
 - voice user interface and, 507–508
- Document-level annotations, for subjectivity and sentiment analysis
 - corpus-based, 274
 - dictionary-based, 272–273
 - overview of, 272
- Document retrieval system, INDRI, 323
- Document structure
 - bibliography, 49–56
 - comparing segmentation methods, 40–41
 - discourse features of segmentation methods, 44
 - discriminative local classification method for segmentation, 36–38
 - discriminative sequence classification method for segmentation, 38–39
 - discussion, 48–49
 - extensions for global modeling sentence segmentation, 40
 - features of segmentation methods, 41–42
 - generative sequence classification method for segmentation, 34–36
 - hybrid methods for segmentation, 39–40
 - introduction to, 29–30
 - lexical features of segmentation methods, 42–43
 - methods for detecting probable sentence or topic boundaries, 33–34
 - performance of segmentation methods, 41
 - processing stages of segmentation tasks, 48
 - prosodic features for segmentation, 45–48
 - sentence boundary detection (segmentation), 30–32
 - speech-related features for segmentation, 45
 - summary, 49
 - syntactic features of segmentation methods, 43–44
 - topic boundary detection (segmentation), 32–33
 - typographical and structural features for segmentation, 44–45
- Document Understanding Conference (DUC), 404, 424
- Documents, in distillation systems
 - indexing, 483
 - preparing, 480–483
 - retrieving, 483–484
- Documents, in IR
 - interlingual representation, 381–382
 - monolingual representation, 372–373
 - preprocessing, 366–367
 - a priori models, 377
 - reducing MLIR to CLIR, 383–384
 - syntax and encoding, 367–368
 - translating entire collection, 379
- Documents, QA searches, 444
- Domain dependent scope, for semantic parsing, 102
- Domain independent scope, for semantic parsing, 102
- Dominance relations, 325
- DSO Corpus, of Sense-Tagged English, 104
- DUC (Document Understanding Conference), 404, 424
- Dutch
 - IR and, 390–391
 - normalization and, 371
 - QA and, 439, 444, 461
 - RTE in, 218
- Edit distance, features of coreference models, 301
- Edit Distance Textual Entailment Suite (EDITS), 240–241
- EDT. See Entity detection and tracking (EDT)
- Elaborative summaries, in automatic summarization, 397
- ElixirFM lexicon, 20
- Ellipsis, linguistic supports for cohesion, 401
- EM algorithm. See Expectation-maximization (EM) algorithm
- Encoding
 - of documents in information retrieval, 368
 - parsing issues related to, 89

English

- call-flow localization and, 514
 - co-occurrence of words between languages, 337–339
 - corpora for relation extraction, 317
 - corpus-based approach to subjectivity and sentiment analysis, 271–272
 - crosslingual language modeling, 197–198
 - dependency graphs in syntax analysis, 65
 - discourse parsers for, 403
 - distillation, 479, 490–491
 - finite-state transducer applied to English example, 17
 - GALE IOD and, 532, 534–536
 - IR and, 390
 - as isolating or analytic language, 7
 - machine translation and, 322, 354, 358
 - manually annotated corpora for, 274
 - mention detection, 287
 - mention detection experiments, 294–296
 - multilingual issues in predicate-argument structures, 146–147
 - normalization and, 370
 - phrase structure trees in syntax analysis, 62
 - polarity analysis of words and phrases, 269
 - productivity/creativity in, 14–15
 - QA and, 444, 461
 - QA architectures and, 437
 - RTE in, 218
 - sentence segmentation markers, 30
 - subjectivity and sentiment analysis, 259–260, 262
 - as SVO language, 356
 - TALES case study, 538
 - tokenization and, 410
 - translingual summarization, 398–399, 424–426
 - word order and, 356
 - WordNet and, 109
- Enrichment, in RTE
 - implementing, 228–231
 - modeling, 225
- Ensemble clustering methods, in relation extraction, 317–318
- Entities
 - classifiers, 292–293
 - entity-based relation extraction, 314–315
 - events. See Events
 - relations. See Relations
 - resolution in semantic interpretation, 100
- Entity detection and tracking (EDT)
 - Bell tree for, 297–298
 - bibliography, 303–307
 - combining entity and relation detection, 320
 - coreference models, 298–300
 - coreference resolution, 295–296
 - data-driven classification, 287–289
 - experiments in coreference resolution, 302–303
 - experiments in mention detection, 294–295
 - features for mention detection, 291–294
 - introduction to, 285–287
 - MaxEnt model applied to, 300–301
 - mention detection task, 287
 - searching for mentions, 289–291
 - summary, 303
- Equivalent terms, in GALE distillation initiative, 475
- Errors
 - machine translation, 335–337, 343, 349
 - parsing, 141–144
 - sentence and topic segmentation, 41
- ESA (Explicit semantic analysis), for interlingual document representation, 382
- Europarl (European Parliament) corpus
 - evaluating co-occurrence of word between languages, 337
 - for IR systems, 391
 - for machine translation, 358
 - phrase translation tables, 345
- European Language Resources Association, 218
- European languages. See also by individual languages
 - crosslingual question answering and, 455
 - QA architectures and, 437
 - whitespace use in, 369
- European Parliament Plenary Speech corpus, 295
- EVALITA, applying to RTE to non-English languages, 218
- Evaluation, in automatic summarization
 - automated evaluation methodologies, 415–418
 - manual evaluation methodologies, 413–415

- overview of, 412–413
- recent developments in, 418–419
- Evaluation, in distillation
 - citation checking, 493
 - GALE and, 492
 - metrics, 493–494
 - overview of, 491–492
 - relevance and redundancy and, 492–493
- Evaluation, in IR
 - best practices, 391
 - data sets for, 389–390
 - experimental setup for, 387
 - measures in, 388–389
 - overview of, 386–387
 - relevance assessment, 387–388
 - trec-eval tool for, 393
- Evaluation, in MT
 - automatic evaluation, 334–335
 - human assessment, 332–334
 - meaning and, 332
 - metrics for, 335–337
- Evaluation, in QA
 - answer correctness, 461–462
 - performance metrics, 462–464
 - tasks, 460–461
- Evaluation, in RTE
 - general model and, 224
 - improving, 251–252
 - performance evaluation, 213–214
- Evaluation, of aggregated NLP, 541
- Evaluative summaries, in automatic summarization, 397
- Events. *See also* Entities
 - extraction, 320–322
 - future directions in extraction, 326
 - matching, 323–326
 - moving beyond sentence processing, 323
 - overview of, 320
 - resolution in semantic interpretation, 100
- Exceptions
 - challenges in NLP aggregation, 524
 - functional morphology models and, 19
- Exclamation point (!), as sentence segmentation marker, 30
- Existence** classifier, in relation extraction, 313
- Expansion documents, query expansion and, 377
- Expansion rules, features of predicate-argument structures, 145
- Expectation-maximization (EM) algorithm
 - split-merge over trees using, 83
 - symmetrization and, 340–341
 - word alignment between languages and, 339–340
- Experiments
 - in coreference resolution, 302–303
 - in mention detection, 294–295
 - setting up for IR evaluation, 387
- Explicit semantic analysis (ESA), for interlingual document representation, 382
- eXtended WordNet (XWN), 451
- Extraction
 - in automatic summarization, 399–400
 - as classification problem, 312–313
 - of events, 320–322, 326
 - of relations, 310–311
- Extraction, in QA
 - candidate extraction from structured sources, 449–450
 - candidate extraction from unstructured sources, 445–449
 - candidate extraction techniques in QA, 443
- Extracts
 - in automatic summarization, 397
 - defined, 400
- Extrinsic evaluation, of summarization, 412
- F-measure, in mention detection, 294
- Factoid QA systems
 - answer correctness, 461
 - answer scores and, 450–453
 - baseline, 443
 - candidate extraction or generation and, 435
 - challenges in, 464–465
 - crosslingual question answering and, 454
 - evaluation tasks, 460–461
 - extracting using high-level searches, 445
 - extracting using structural matching, 446
 - MURAX and, 434
 - performance metrics, 462–463
 - questions, 433
 - type classification of, 440
- Factoids, in manual evaluation of summarization, 413
- Factored (cascaded) model, 313
- Factored language models (FLM)
 - machine translation and, 355

- Factored language models (*continued*)
 - morphological categories in, 193
 - overview of, 183–184
- Feature extractors
 - building summarization systems, 423
 - distillation and, 485–486
 - summarization and, 406
- Features
 - in mention detection system, 291–294
 - typed feature structures and unification, 18–19
 - in word disambiguation system, 110–112
- Features, in sentence or topic segmentation
 - defined, 33
 - discourse features, 44
 - lexical features, 42–43
 - overview of, 41–42
 - predictions based on, 29
 - prosodic features, 45–48
 - speech-related features, 45
 - syntactic features, 43–44
 - typographical and structural features, 44–45
- Fertility, word alignment and, 340
- Files types, document syntax and, 367–368
- Finite-state morphology, 16–18
- Finite-state transducers, 16–17, 20
- Finnish
 - as agglutinative language, 7
 - IR and, 390–391
 - irregular verbs, 10
 - language modeling, 189–191
 - parsing issues related to morphology, 91
 - summarization and, 399
- FIRE (Forum for Information Retrieval Evaluation), 390
- Flexible, distributed componentization
 - desired attributes of NLP aggregation, 524–525
 - in GATE, 530
 - in InfoSphere Streams, 530
 - in UIMA, 528
- FLM. See Factored language models (FLM)
- Fluency, of translation, 334
- Forum for Information Retrieval Evaluation (FIRE), 390
- FraCaS corpus, applying natural logic to RTE, 246
- Frame elements
 - in PSG, 126
 - semantic frames in FrameNet, 118
- FrameNet
 - limitation of, 122–123
 - resources, 122
 - resources for predicate-argument recognition, 118–122
- Freebase, 449
- French
 - automatic speech recognition (ASR), 179
 - dictionary-based approach to subjectivity and sentiment analysis, 267
 - human assessment of translation English to, 332–333
 - IR and, 378, 390–391
 - language modeling, 188
 - localization of spoken dialog systems, 513
 - machine translation and, 350, 353–354, 358
 - phrase structure trees in syntax analysis, 62
 - polarity analysis of words and phrases, 269
 - QA and, 454, 461
 - RTE in, 217–218
 - translingual summarization, 398
 - word segmentation and, 90
 - WordNet and, 109
- Functional morphology, 19–21
- Functions, viewing language relations as, 17
- Fusional languages
 - functional morphology models and, 19
 - morphological typology and, 8
 - normalization and, 371
 - preprocessing best practices in IR, 371
- GALE. See Global Autonomous Language Exploitation (GALE)
- GALE Type System (GTS), 534–535
- GATE. See General Architecture for Text Engineering (GATE)
- Gazetteer, features of mention detection system, 293
- GEN-AFF (general-affiliation), relation class, 312
- Gender
 - ambiguity resolution, 13
 - multilingual approaches to grammatical gender, 398

- General Architecture for Text Engineering (GATE)
 attributes of, 530
 history of summarization systems, 399
 overview of, 529–530
 summarization frameworks, 422
- General Inquirer, subjectivity and sentiment analysis lexicon, 262
- Generalized backoff strategy, in FLM, 183–184
- Generative parsing models, 83–84
- Generative sequence classification methods
 complexity of, 40
 overview of, 34
 performance of, 41
 for sentence/topic boundary detection, 34–36
- Geometric vector space model, for document retrieval, 375
- GeoQuery
 resources for meaning representation, 149
 supervised systems for semantic parsing, 151
- German
 co-occurrence of words between languages, 337–339
 dictionary-based approach to subjectivity and sentiment analysis, 265–266, 273
 discourse parsers for, 403
 as fusional language, 8
 IR and, 390–392
 language modeling, 189
 mention detection, 287
 morphological richness of, 354–355
 normalization, 370–371
 OOV rate in, 191
 phrase-based model for decoding, 345
 polarity analysis of words and phrases, 269
 QA and, 461
 RTE in, 218
 subjectivity and sentiment analysis, 259, 276
 summarization and, 398, 403–404, 420
 WordNet and, 109
- Germanic languages, language modeling for, 189
- GetService process, of voice user interface (VUI), 506–507
- Giza, machine translation program, 423
- GIZA toolkit, for machine translation, 357
- Global Autonomous Language Exploitation (GALE)
 distillation initiative of DARPA, 475–476
 evaluation in distillation, 492
 Interoperability Demo case study. See Interoperability Demo (IOD), GALE case study
 metrics for evaluating distillation, 494
 relevance and redundancy in, 477–479
- Global linear model, discriminative approach to learning, 84
- Good-Turing
 machine translation and, 345
 smoothing techniques in language model estimation, 172
- Google, 435
- Google Translate, 331, 455
- Grammars
 Combinatory Categorical Grammar (CCG), 129–130
 context-free. See Context-free grammar (CFGs)
 head-driven phrase structure grammar (HPSG), 18
 localization of, 514, 516–517
 morphological resource grammars, 19, 21
 phrase structure. See Phrase Structure Grammar (PSG)
 probabilistic context-free. See Probabilistic context-free grammars (PCFGs)
 rule-based grammars in speech recognition, 501–503
 Tree-Adjoining Grammar (TAG), 130
 voice user interface (VUI), 508–509
- Grammatical Framework, 19, 21
- Graph-based approaches, to automatic summarization
 applying RST to summarization, 402–404
 coherence and cohesion and, 401–402
 LexPageRank, 406
 overview of, 401
 TextRank, 404–406
- Graph generation, in RTE
 implementing, 231–232
 modeling, 226
- Graphemes, 4
- Greedy best-fit decoding, in mention detection, 322
- Groups, aligning views in RTE, 233

- Grow-diag-final method, for word alignment, 341
- GTS (GALE Type System), 534–535
- Gujarati. See Indian languages
- HDP (Hierarchical Dirichlet process), 187
- Head-driven phrase structure grammar (HPSG), 18
- Head word
 - dependency trees and, 131
 - in Phrase Structure Grammar (PSG), 124
- Headlines, typographical and structural features for sentence and topic segmentation, 44–45
- Hebrew
 - encoding and script, 368
 - preprocessing best practices in IR, 371
 - tokens in, 4
 - unification-based models, 19
- HELM (hidden event language model)
 - applied to sentence segmentation, 36
 - methods for sentence or topic segmentation, 40
- Hidden event language model (HELM)
 - applied to sentence segmentation, 36
 - methods for sentence or topic segmentation, 40
- Hidden Markov model (HMM)
 - applied to topic and sentence segmentation, 34–36
 - measuring token frequency, 369
 - mention detection and, 287
 - methods for sentence or topic segmentation, 39
 - word alignment between languages and, 340
- Hierarchical Dirichlet process (HDP), 187
- Hierarchical phrase-based models, in machine translation, 350–351
- Hierarchical phrase pairs, in machine translation, 351
- High-level features, in event matching, 324
- Hindi. See also Indian languages
 - IR and, 390
 - resources for semantic parsing, 122
 - translingual summarization, 399
- History, conditional context of probability, 83
- HMM. See Hidden Markov model (HMM)
- Homonymy
 - in Korean, 10
 - word sense ambiguities and, 104
- HowNet
 - dictionary-based approach to subjectivity and sentiment analysis, 272–273
 - semantic parsing resource, 105
- HTML Parser, preprocessing IR documents, 392
- Hunalign tool, for machine translation, 357
- Hungarian
 - dependency graphs in syntax analysis, 65
 - IR and, 390
 - morphological richness of, 355
- Hybrid methods, for segmentation, 39–40
- Hypergraphs, worst-case parsing algorithm for CFGs, 74–79
- Hypernyms, 442
- Hyponymy, 310
- Hypotheses, machine translation and, 346
- IBM Models, for machine translation, 338–341
- Identification, of arguments, 123, 139–140
- IDF. See Inverse document frequency (IDF)
- IE. See Information extraction (IE)
- ILP (Integer linear programming), 247
- Implementation process, in RTE
 - alignment, 233–236
 - enrichment, 228–231
 - graph generation, 231–232
 - inference, 236–238
 - overview of, 227
 - preprocessing, 227–228
 - training, 238
- IMS (It Makes Sense), program for word sense disambiguation, 117
- Independence assumption
 - document retrieval and, 372
 - overcoming in predicate-argument structure, 137–138
- Indexes
 - of documents in distillation system, 483
 - for IR generally, 366
 - latent semantic indexing (LSI), 381
 - for monolingual IR, 373–374
 - for multilingual IR, 383–384
 - phrase indices, 366, 369–370
 - positional indices, 366
 - translating MLIR queries, 384

- Indian languages, IR and. See also Hindi, 390
- INDRI document retrieval system, 323
- Inexact retrieval models, for monolingual information retrieval, 374
- InfAP metrics, for IR performance, 389
- Inference, textual. See Textual inference
- Inflectional paradigms
 - in Czech, 11–12
 - in morphologically rich languages, 189
- Information context, as measure of semantic similarity, 112
- Information extraction (IE). See also Entity detection and tracking (EDT)
 - defined, 285
 - entity and event resolution and, 100
- Information retrieval (IR)
 - bibliography, 394–396
 - crosslingual. See Crosslingual information retrieval (CLIR)
 - data sets used in evaluation of, 389–391
 - distillation compared with, 475
 - document preprocessing for, 366–367
 - document syntax and encoding, 367–368
 - evaluation in, 386–387, 391
 - introduction to, 366
 - key word searches in, 433
 - measures in, 388–389
 - monolingual. See Monolingual information retrieval
 - multilingual. See Multilingual information retrieval (MLIR)
 - normalization and, 370–371
 - preprocessing best practices, 371–372
 - redundancy problem and, 488
 - relevance assessment, 387–388
 - summary, 393
 - tokenization and, 369–370
 - tools, software, and resources, 391–393
 - translingual, 491
- Informative summaries, in automatic summarization, 401–404
- InfoSphere Streams, 530–531
- Insertion metric, in machine translation, 335
- Integer linear programming (ILP), 247
- Interactive voice response (IVR), 505, 511
- Interoperability Demo (IOD), GALE case study
 - computational efficiency, 537
 - flexible application building with, 537
 - functional description, 532–534
 - implementing, 534–537
 - overview of, 531–532
- Interoperability, in aggregated NLP, 540
- Interpolation, language model adaptation and, 176
- Intrinsic evaluation, of summarization, 412
- Inverse document frequency (IDF)
 - answer scores in QA and, 450–451
 - document representation in monolingual IR, 373
 - relationship questions and, 488
 - searching over unstructured sources, 445
- Inverted indexes, for monolingual information retrieval, 373–374
- IOD case study. See Interoperability Demo (IOD), GALE case study
- IR. See Information retrieval (IR)
- Irregularity
 - defined, 8
 - issues with morphology induction, 21
 - in linguistic models, 8–10
- IRSTLM toolkit, for machine translation, 357
- Isolating (analytic) languages
 - finite-state technology applied to, 18
 - morphological typology and, 7
- It Makes Sense (IMS), program for word sense disambiguation, 117
- Italian
 - dependency graphs in syntax analysis, 65
 - IR and, 390–391
 - normalization and, 371
 - polarity analysis of words and phrases, 269
 - QA and, 461
 - RTE in, 218
 - summarization and, 399
 - WordNet and, 109
- IVR (interactive voice response), 505, 511
- IXIR distillation system, 488–489
- Japanese
 - as agglutinative language, 7
 - anaphora frequency in, 444
 - call-flow localization and, 514
 - crosslingual QA, 455
 - discourse parsers for, 403
 - EDT and, 286
 - GeoQuery corpus translated into, 149
 - IR and, 390

- Japanese (*continued*)
 - irregular verbs, 10
 - language modeling, 193–194
 - polarity analysis of words and phrases, 269
 - preprocessing best practices in IR, 371–372
 - QA architectures and, 437–438, 461, 464
 - semantic parsing, 122, 151
 - subjectivity and sentiment analysis, 259, 267–271
 - word order and, 356
 - word segmentation in, 4–5
- JAVELIN system, for QA, 437
- Joint inference, NLP and, 320
- Joint systems
 - optimization vs. interoperability in
 - aggregated NLP, 540
 - types of EDT architectures, 286
- Joshua machine translation program, 357, 423
- JRC-Acquis corpus
 - for evaluating IR systems, 390
 - for machine translation, 358

- KBP (Knowledge Base Population), of Text Analysis Conferences (TAC), 481–482
- Kernel functions, SVM mapping and, 317
- Kernel methods, for relation extraction, 319
- Keyword searches
 - in IR, 433
 - searching over unstructured sources, 443–445
- KL-ONE system, for predicate-argument recognition, 122
- Kneser-Ney smoothing technique, in language model estimation, 172
- Knowledge Base Population (KBP), of Text Analysis Conferences (TAC), 481–482
- Korean
 - as agglutinative language, 7
 - ambiguity in, 10–11
 - dictionary-based approach in, 16
 - EDT and, 286
 - encoding and script, 368
 - finite-state models, 18
 - gender, 13
 - generative parsing model, 92
 - IR and, 390
 - irregular verbs, 10
 - language modeling, 190
 - language modeling using subword units, 192
 - morphemes in, 6–7
 - polarity analysis of words and phrases, 269
 - preprocessing best practices in IR, 371–372
 - resources for semantic parsing, 122
 - word segmentation in, 4–5
- KRISPER program, for rule-based semantic parsing, 151

- Language identification, in MLIR, 383
- Language models
 - adaptation, 176–178
 - Bayesian parameter estimation, 173–174
 - Bayesian topic-based, 186–187
 - bibliography, 199–208
 - class-based, 178–179
 - crosslingual, 196–198
 - discriminative, 179–180
 - for document retrieval, 375–376
 - evaluation of, 170–171
 - factored, 183–184
 - introduction to, 169
 - language-specific problems, 188–189
 - large-scale models, 174–176
 - MaxEnt, 181–183
 - maximum-likelihood estimation and smoothing, 171–173
 - morphological categories in, 192–193
 - for morphologically rich languages, 189–191
 - multilingual, 195–196
 - n*-gram approximation, 170
 - neural network, 187–188
 - spoken vs. written languages and, 194–195
 - subword unit selection, 191–192
 - summary, 198
 - syntax-based, 180–181
 - tree-based, 185–186
 - types of, 178
 - variable-length, 179
 - word segmentation and, 193–194
- The Language Understanding Annotated Corpus, 425
- Langue and parole (de Saussure), 13
- Latent Dirichlet allocation (LDA) model, 186
- Latent semantic analysis (LSA)
 - bilingual (bLSA), 197–198
 - language model adaptation and, 176–177
 - probabilistic (PLSA), 176–177

- Latent semantic indexing (LSI), 381
- Latin
 - as fusional language, 8
 - morphologies of, 20
 - preprocessing best practices in IR, 371
 - transliteration of scripts to, 368
- Latvian
 - IR and, 390
 - summarization and, 399
- LDA (Latent Dirichlet allocation) model, 186
- LDC. See Linguistic Data Consortium (LDC)
- LDOCE (Longman Dictionary of Contemporary English), 104
- LEA. See Lexical entailment algorithm (LEA)
- Learning, discriminative approach to, 84
- Lemmas
 - defined, 5
 - machine translation metrics and, 336
 - mapping terms to, 370
- Lemmatizers
 - mapping terms to lemmas, 370
 - preprocessing best practices in IR, 371
- Lemur IR framework, 392
- Lesk algorithm, 105–106
- Lexemes
 - functional morphology models and, 19
 - overview of, 5
- Lexical chains, in topic segmentation, 38, 43
- Lexical choice, in machine translation, 354–355
- Lexical collocation, 401
- Lexical entailment algorithm (LEA)
 - alignment stage of RTE model, 236
 - enrichment stage of RTE model, 228–231
 - inference stage of RTE model, 237
 - preprocessing stage in RTE model, 227–228
 - training stage of RTE model, 238
- Lexical features
 - context as, 110
 - in coreference models, 301
 - in event matching, 324
 - in mention detection, 292
 - of relation extraction systems, 314
 - in sentence and topic segmentation, 42–43
- Lexical matching, 212–213
- Lexical ontologies, relation extraction and, 310
- Lexical strings, 17, 18
- Lexicon, of languages
 - building, 265–266
 - dictionary-based approach to subjectivity and sentiment analysis, 270, 273
 - ElixirFM lexicon of Arabic, 20
 - sets of lexemes constituting, 5
 - subjectivity and sentiment analysis with, 262, 275–276
- LexPageRank, approach to automatic summarization, 406, 411
- LexTools, for finite-state morphology, 16
- Linear model interpolation, for smoothing language model estimates, 173
- LinearRank algorithm, learning summarization, 408
- lingPipe tool, for summarization, 423
- Linguistic challenges, in MT
 - lexical choice, 354–355
 - morphology and, 355
 - word order and, 356
- Linguistic Data Consortium (LDC)
 - corpora for machine translation, 358
 - evaluating co-occurrence of word between languages, 337
 - history of summarization systems, 399
 - OntoNotes corpus, 104
 - on sentence segmentation markers in conversational speech, 31
 - summarization frameworks, 422
- List questions
 - extension to, 453
 - QA and, 433
- Local collocations, features of supervised systems, 110–111
- Localization, of spoken dialog systems
 - call-flow localization, 514
 - localization of grammars, 516–517
 - overview of, 513–514
 - prompt localization, 514–516
 - testing, 519–520
 - training, 517–519
- Log-linear models, phrase-based models for MT, 348–349
- Logic-based representation, applying to RTE, 242–244
- Logographic scripts, preprocessing best practices in IR, 371
- Long-distance dependencies, syntax-based language models for, 180–181
- Longman Dictionary of Contemporary English (LDOCE), 104

- Lookup operations, dictionaries and, 16
- Loudness, prosodic cues, 45–47
- Low-level features, in event matching, 324
- Lucene
 - document indexing with, 483
 - document retrieval with, 483–484
 - IR frameworks, 392
- LUNAR QA system, 434

- Machine learning. *See also* Conditional random fields (CRFs)
 - event extraction and, 322
 - measuring token frequency, 369
 - summarization and, 406–409
 - word alignment as learning problem, 341–343
- Machine translation (MT)
 - alignment models, 340
 - automatic evaluation, 334–335
 - bibliography, 360–363
 - chart decoding, 351–352
 - CLIR applied to, 380–381
 - co-occurrence of words and, 337–338
 - coping with model size, 349–350
 - corpora for, 358
 - crosslingual QA and, 454
 - cube pruning approach to decoding, 347–348
 - data reorganization and, 536
 - data resources for, 356–357
 - decoding phrase-based models, 345–347
 - expectation maximization (EM) algorithm, 339–340
 - future directions, 358–359
 - in GALE IOD, 532–533
 - hierarchical phrase-based models, 350–351
 - history and current state of, 331–332
 - human assessment and, 332–334
 - IBM Model 1, 338–339
 - lexical choice, 354–355
 - linguistic choices, 354
 - log-linear models and parameter tuning, 348–349
 - meaning evaluation, 332
 - metrics, 335–337
 - morphology and, 355
 - multilingual automatic summarization and, 410
 - overview of, 331
 - paraphrasing and, 59
 - phrase-based models, 343–344
 - programs for, 423
 - RTE applied to, 217–218
 - in RTTS, 538
 - sentences as processing unit in, 29
 - statistical. *See* Statistical machine translation (SMT)
 - summary, 359
 - symmetrization, 340–341
 - syntactic models, 352–354
 - systems for, 357–358
 - in TALES, 538
 - tools for, 356–357, 392
 - training issues, 197
 - training phrase-based models, 344–345
 - translation-based approach to CLIR, 378–380
 - tree-based models, 350
 - word alignment and, 337, 341–343
 - word order and, 356
- MAP (maximum a posteriori)
 - Bayesian parameter estimation and, 173–174
 - language model adaptation and, 177–178
- MAP (Mean average precision), metrics for IR systems, 389
- Marathi, 390
- Margin infused relaxed algorithm (MIRA)
 - methods for sentence or topic segmentation, 39
 - unsupervised approaches to machine learning, 342
- Markov model. *See also* Hidden Markov model (HMM), 34–36
- Matches, machine translation metrics, 335
- Matching events, 323–326
- Mate retrieval setup, relevance assessment and, 388
- MaxEnt model
 - applied to distillation, 480
 - classifiers for relation extraction, 316–317
 - classifiers for sentence or topic segmentation, 37, 39–40
 - coreference resolution with, 300–301
 - language model adaptation and, 177
 - memory-based learning compared with, 322
 - mention detection, 287–289

- modeling using morphological categories, 193
- modeling without word segmentation, 194
- overview of, 181–183
- subjectivity and sentiment analysis with, 274
- unsupervised approaches to machine learning, 342
- Maximal marginal relevance (MMR), in automatic summarization, 399
- Maximum a posteriori (MAP)
 - Bayesian parameter estimation and, 173–174
 - language model adaptation and, 177–178
- Maximum-likelihood estimation
 - Bayesian parameter estimation and, 173–174
 - as parameter estimation language model, 171–173
 - used with document models in information retrieval, 375–376
- MEAD system, for automatic summarization, 410–411, 423
- Mean average precision (MAP), metrics for IR systems, 389
- Mean reciprocal rank (MRR), metrics for QA systems, 462–463
- Meaning chunks, semantic parsing and, 97
- Meaning of words. See Word meaning
- Meaning representation
 - Air Travel Information System (ATIS), 148
 - Communicator program, 148–149
 - GeoQuery, 149
 - overview of, 147–148
 - RoboCup, 149
 - rule-based systems for, 150
 - semantic interpretation and, 101
 - software programs for, 151
 - summary, 153–154
 - supervised systems for, 150–151
- Measures. See Metrics
- Media Resource Control Protocol (MRCP), 504
- Meeting Recorder Dialog Act (MRDA), 31
- Memory-based learning, 322
- MEMT (multi-engine machine translation), in GALE IOD, 532–533
- Mention detection
 - Bell tree and, 297
 - computing probability of mention links, 297–300
 - data-driven classification, 287–289
 - experiments in, 294–295
 - features for, 291–294
 - greedy best-fit decoding, 322
 - MaxEnt model applied to entity-mention relationships, 301
 - mention-matching features in event matching, 324
 - overview of, 287
 - problems in information extraction, 285–286
 - in Rosetta Consortium distillation system, 480–481
 - searching for mentions, 289–291
- Mention-synchronous process, 297
- Mentions
 - entity relations and, 310–311
 - named, nominal, prenominal, 287
- Meronymy, 310
- MERT (minimum error rate training), 349
- METEOR, metrics for machine translation, 336
- METONYMY class, ACE, 312
- Metrics
 - distillation, 491–494
 - graph generation and, 231
 - IR, 388
 - machine translation, 335–337
 - magnitude of RTE metrics, 233
 - for multilingual automatic summarization, 419–420
 - QA, 462–464
 - RTE annotation constituents, 222–224
- Microsoft, history of QA systems and, 435
- Minimum error rate training (MERT), 349
- Minimum spanning trees (MSTs), 79–80
- Minipar
 - dependency parsing with, 456
 - rule-based dependency parser, 131–132
- MIRA (margin infused relaxed algorithm)
 - methods for sentence or topic segmentation, 39
 - unsupervised approaches to machine learning, 342
- Mixed initiative dialogs, in spoken dialog systems, 509

- MLIR. See Multilingual information retrieval (MLIR)
- MLIS-MUSI summarization system, 399
- MMR (maximal marginal relevance), in automatic summarization, 399
- Models, information retrieval
 - monolingual, 374–376
 - selection best practices, 377–378
- Models, word alignment
 - EM algorithm, 339–340
 - IBM Model 1, 338–339
 - improvements on IBM Model 1, 340
- Modern Standard Arabic (MSA), 189–191
- Modification processes, in automatic summarization, 399–400
- Modifier word, dependency trees and, 131
- Monolingual information retrieval. See also Information retrieval (IR)
 - document a priori models, 377
 - document representation, 372–373
 - index structures, 373–374
 - model selection best practices, 377–378
 - models for, 374–376
 - overview of, 372
 - query expansion technique, 376–377
- Monotonicity
 - applying natural logic to RTE, 246
 - defined, 224
- Morfessor package, for identifying morphemes, 191–192
- Morphemes
 - abstract in morphology induction, 21
 - automatic algorithms for identifying, 191–192
 - defined, 4
 - examples of, 6–7
 - functional morphology models and, 19
 - Japanese text segmented into, 438
 - language modeling for morphologically rich languages, 189
 - overview of, 5–6
 - parsing issues related to, 90–91
 - typology and, 7–8
- Morphological models
 - automating (morphology induction), 21
 - dictionary-based, 15–16
 - finite-state, 16–18
 - functional, 19–21
 - overview of, 15
 - unification-based, 18–19
- Morphological parsing
 - ambiguity and, 10–13
 - dictionary lookup and, 15
 - discovery of word structure by, 3
 - irregularity and, 8–10
 - issues and challenges, 8
- Morphology
 - categories in language models, 192–193
 - compared with syntax and phonology and orthography, 3
 - induction, 21
 - language models for morphologically rich languages, 189–191
 - linguistic challenges in machine translation, 355
 - parsing issues related to, 90–92
 - typology, 7–8
- Morphs (segments)
 - data-sparseness problem and, 286
 - defined, 5
 - functional morphology models and, 19
 - not all morphs can be assumed to be morphemes, 7
 - typology and, 8
- Moses system
 - grow-diag-final method, 341
 - machine translation, 357, 423
- MPQA corpus
 - manually annotated corpora for English, 274
 - subjectivity and sentiment analysis, 263, 272
- MRCP (Media Resource Control Protocol), 504
- MRDA (Meeting Recorder Dialog Act), 31
- MRR (Mean reciprocal rank), metrics for QA systems, 462–463
- MSA (Modern Standard Arabic), 189–191
- MSE (Multilingual Summarization Evaluation), 399, 425
- MSTs (minimum spanning trees), 79–80
- Multext Dataset, corpora for evaluating IR systems, 390
- Multi-engine machine translation (MEMT), in GALE IOD, 532–533
- Multilingual automatic summarization
 - automated evaluation methodologies, 415–418

- building a summarization system, 420–421, 423–424
- challenges in, 409–410
- competitions related to, 424–425
- data sets for, 425–426
- devices/tools for, 423
- evaluating quality of summaries, 412–413
- frameworks summarization system can be implemented in, 422–423
- manual evaluation methodologies, 413–415
- metrics for, 419–420
- recent developments, 418–419
- systems for, 410–412
- Multilingual information retrieval (MLIR)
 - aggregation models, 385
 - best practices, 385–386
 - defined, 382
 - index construction, 383–384
 - language identification, 383
 - overview of, 365
 - query translation, 384
- Multilingual language modeling, 195–196
- Multilingual Summarization Evaluation (MSE), 399, 425
- Multimodal distillation, 490
- Multiple reference translations, 336
- Multiple views, overcoming parsing errors, 142–144
- MURAX, 434

- n*-gram
 - localization of grammars and, 516
 - trigrams, 502–503
- n*-gram approximation
 - language model evaluation and, 170–171
 - language-specific modeling problems, 188–189
 - maximum-likelihood estimation, 171–172
 - smoothing techniques in language model estimation, 172
 - statistical language models using, 170
 - subword units used with, 192
- n*-gram models. See also Phrase indices
 - AutoSummENG graph, 419
 - character models, 370
 - defined, 369–370
 - document representation in monolingual IR, 372–373
- Naïve Bayes
 - classifiers for relation extraction, 316
 - subjectivity and sentiment analysis, 274
- Named entity recognition (NER)
 - aligning views in RTE, 233
 - automatic summarization and, 398
 - candidate answer generation and, 449
 - challenges in RTE, 212
 - enrichment stage of RTE model, 229–230
 - features of supervised systems, 112
 - graph generation stage of RTE model, 231
 - impact on searches, 444
 - implementing RTE and, 227
 - information extraction and, 100
 - mention detection related to, 287
 - in PSG, 125–126
 - QA architectures and, 439
 - in Rosetta Consortium distillation system, 480
 - in RTE, 221
- National Institute of Standards and Technology (NIST)
 - BLEU score, 295
 - relation extraction and, 311
 - summarization frameworks, 422
 - textual entailment and, 211, 213
- Natural language
 - call routing, 510
 - parsing, 57–59
- Natural language generation (NLG), 503–504
- Natural language processing (NLP)
 - applications of syntactic parsers, 59
 - applying to non-English languages, 218
 - distillation and. See Distillation
 - extraction of document structure as aid in, 29
 - joint inference, 320
 - machine translation and, 331
 - minimum spanning trees (MST) and, 79
 - multiview representation of analysis, 220–222
 - packages for, 253
 - problems in information extraction, 286
 - relation extraction and, 310
 - RTE applied to NLP problems, 214
 - RTE as subfield of. See Recognizing textual entailment (RTE)
 - syntactic analysis of natural language, 57
 - textual inference, 209

- Natural language processing (NLP),
 - combining engines for aggregation architectures, 527
 - bibliography, 548–549
 - computational efficiency, 525–526
 - data-manipulation capacity, 526
 - flexible, distributed componentization, 524–525
 - GALE Interoperability Demo case study, 531–537
 - General Architecture for Text Engineering (GATE), 529–530
 - InfoSphere Streams, 530–531
 - introduction to, 523–524
 - lessons learned, 540–542
 - robust processing, 526–527
 - RTTS case study, 538–540
 - summary, 542
 - TALES case study, 538
 - Unstructured Information Management Architecture (UIMA), 527–529, 542–547
- Natural Language Toolkit (NLTK), 422
- Natural language understanding (NLU), 209
- Natural logic-based representation, applying to RTE, 245–246
- NDCG (Normalized discounting cumulative gain), 389
- NER. See Named entity recognition (NER)
- Neural network language models (NNLMs)
 - language modeling using morphological categories, 193
 - overview of, 187–188
- Neural networks, approach to machine learning, 342
- Neutralization, homonyms and, 12
- The New York Times Annotated Corpus, 425
- NewsBlaster, for automatic summarization, 411–412
- NII Test Collection for IR Systems (NTCIR)
 - answer scores in QA and, 453
 - data sets for evaluating IR systems, 390
 - evaluation of QA, 460–464
 - history of QA systems and, 434
- NIST. See National Institute of Standards and Technology (NIST)
- NLG (natural language generation), 503–504
- NLP. See Natural language processing (NLP)
- NLTK (Natural Language Toolkit), 422
- NNLMs (neural network language models)
 - language modeling using morphological categories, 193
 - overview of, 187–188
- NOMinalization LEXicon (NOMLEX), 121
- Non projective dependency trees, 65–66
- Nonlinear languages, morphological typology and, 8
- Normalization
 - Arabic, 12
 - overview of, 370–371
 - tokens and, 4
 - Z-score normalization, 385
- Normalized discounting cumulative gain (NDCG), 389
- Norwegian, 461
- Noun arguments, 144–146
- Noun head, of prepositional phrases in PSB, 127
- NTCIR. See NII Test Collection for IR Systems (NTCIR)
- Numerical quantities (NUM) constituents, in RTE, 221, 233
- Objective word senses, 261
- OCR (Optical character recognition), 31
- One vs. All (OVA) approach, 136–137
- OntoNotes corpus, 104
- OOV (out of vocabulary)
 - coverage rates in language models, 170
 - morphologically rich languages and, 189–190
- OOV rate
 - in Germanic languages, 191
 - inventorying morphemes and, 192
 - language modeling without word segmentation, 194
- Open-domain QA systems, 434
- Open Standard by the Organization for the Advancement of Structured Information Standards (OASIS), 527
- OpenCCG project, 21
- openNLP, 423
- Opinion questions, QA and, 433
- OpinionFinder
 - as rule-based system, 263
 - subjectivity and sentiment analysis, 271–272, 275–276
 - subjectivity and sentiment analysis lexicon, 262

- Optical character recognition (OCR), 31
- OPUS project, corpora for machine translation, 358
- Ordinal constituent position, in PSG, 127
- ORG-AFF (organization-affiliation) class, 311–312
- Orthography
 - Arabic, 11
 - issues with morphology induction, 21
- Out of vocabulary (OOV)
 - coverage rates in language models, 170
 - morphologically rich languages and, 189–190
- PageRank
 - automatic summarization, 401
 - LexPageRank compared with, 406
 - TextRank compared with, 404
- Paradigms
 - classification, 133–137
 - functional morphology models and, 19
 - inflectional paradigms in Czech, 11–12
 - inflectional paradigms in morphologically rich languages, 189
- ParaEval
 - automatic evaluation of summarization, 418
 - metrics in, 420
- Paragraphs, sentences forming, 29
- Parallel backoff, 184
- Parameter estimation language models
 - Bayesian parameter estimation, 173–174
 - large-scale models, 174–176
 - maximum-likelihood estimation and smoothing, 171–173
- Parameter tuning, 348–349
- Parameters, functional morphology models and, 19
- Paraphrasing, parsing natural language and, 58–59
- Parasitic gap recovery, in RTE, 249
- parole and langue (de Saussure), 13
- Parsing
 - algorithms for, 70–72
 - ambiguity resolution in, 80
 - defined, 97
 - dependency parsing, 79–80
 - discriminative models, 84–87
 - generative models, 83–84
 - hypergraphs and chart parsing, 74–79
 - natural language, 57–59
 - semantic parsing. *See* semantic parsing
 - sentences as processing unit in, 29
 - shift-reduce parsing, 72–73
- Part of speech (POS)
 - class-based language models and, 178
 - features of supervised systems, 110
 - implementing RTE and, 227
 - natural language grammars and, 60
 - in PSG, 125–127
 - QA architectures and, 439
 - in Rosetta Consortium distillation system, 480
 - for sentence segmentation, 43
 - syntactic analysis of natural language, 57–58
- PART-WHOLE relation class, 311
- Partial order method, for ranking sentences, 407
- Particle language model, subword units in, 192
- Partition function, in MaxEnt formula, 316
- PASCAL. *See* Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL)
- Path
 - in CCG, 130
 - in PSG, 124, 128–129
 - in TAG, 130
 - for verb sense disambiguation, 112
- Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL)
 - evaluating textual entailment, 213
 - RTE challenge, 451–452
 - textual entailment and, 211
- Pauses, prosodic cues, 45–47
- Peer surveys, in evaluation of summarization, 412
- Penn Treebank
 - dependency trees and, 130–132
 - parsing issues and, 87–89
 - performance degradation and, 147
 - phrase structure trees in, 68, 70
 - PropBank and, 123
- PER (Position-independent error rate), 335
- PER-SOC (personal-social) relation class, 311
- Performance
 - of aggregated NLP, 541

- Performance (*continued*)
 - combining classifiers to boost (Combination hypothesis), 293
 - competence vs. performance (Chomsky), 13
 - of document segmentation methods, 41
 - evaluating IR, 389
 - evaluating QA, 462–464
 - evaluating RTE, 213–214
 - feature performance in predicate-argument structure, 138–140
 - Penn Treebank, 147
- Period (.), sentence segmentation markers, 30
- Perplexity
 - criteria in language model evaluation, 170–171
 - inventorying morphemes and, 192
 - language modeling using morphological categories, 193
 - language modeling without word segmentation, 194
- Persian
 - IR and, 390
 - unification-based models, 19
- Phoenix, 150
- Phonemes, 4
- Phonology
 - compared with morphology and syntax and orthography, 3
 - issues with morphology induction, 21
- Phrasal verb collocations, in PSG, 126
- Phrase-based models, for MT
 - coping with model size, 349–350
 - cube pruning approach to decoding, 347–348
 - decoding, 345–347
 - hierarchical phrase-based models, 350–351
 - log-linear models and parameter tuning, 348–349
 - overview of, 343–344
 - training, 344–345
- Phrase feature, in PSG, 124
- Phrase indices, tokenization and, 366, 369–370
- Phrase-level annotations, for subjectivity and sentiment analysis
 - corpus-based, 267–269
 - dictionary-based, 264–267
 - overview of, 264
- Phrase Structure Grammar (PSG), 124–129
- Phrase structure trees
 - examples of, 68–70
 - morphological information in, 91
 - in syntactic analysis, 67
 - treebank construction and, 62
- Phrases
 - early approaches to summarization and, 400
 - types in CCG, 129–130
- PHYS (physical) relation class, 311
- Pipeline approach, to event extraction, 320–321
- Pitch, prosodic cues, 45–47
- Pivot language, translation-based approach to CLIR, 379–380
- Polarity
 - corpus-based approach to subjectivity and sentiment analysis, 269
 - relationship to monotonicity, 246
 - word sense classified by, 261
- Polysemy, 104
- Portuguese
 - IR and, 390–391
 - QA and, 461
 - RTE in, 218
- POS. See Part of speech (POS)
- Position-independent error rate (PER), 335
- Positional features, approaches to summarization and, 401
- Positional indices, tokens and, 366
- Posting lists, term relationships in document retrieval, 373–374
- Pre-reordering, word order in machine translation, 356
- Preboundary lengthening, in sentence segmentation, 47
- Precision, IR evaluation measure, 388
- Predicate-argument structure
 - base phrase chunks, 132–133
 - classification paradigms, 133–137
 - Combinatory Categorical Grammar (CCG), 129–130
 - dependency trees, 130–132
 - feature performance, salience, and selection, 138–140
 - FrameNet resources, 118–119
 - multilingual issues, 146–147
 - noun arguments, 144–146
 - other resources, 121–122
 - overcoming parsing errors, 141–144

- overcoming the independence assumption, 137–138
- Phrase Structure Grammar (PSG), 124–129
- PropBank resources, 119–121
- robustness across genres, 147
- semantic interpretation and, 100
- semantic parsing. *See* Predicate-argument structure
- semantic role labeling, 118
- sizing training data, 140–141
- software programs for, 147
- structural matching and, 447–448
- summary, 153
- syntactic representation, 123–124
- systems, 122–123
- Tree-Adjoining Grammar, 130
- Predicate context, in PSG, 129
- Predicate feature, in Phrase Structure Grammar (PSG), 124
- Prepositional phrase adjunct, features of supervised systems, 111
- Preprocessing, in IR
 - best practices, 371–372
 - documents for information retrieval, 366–367
 - tools for, 392
- Preprocessing, in RTE
 - implementing, 227–228
 - modeling, 224–225
- Preprocessing queries, 483
- Preterminals. *See* Part of speech (POS)
- Previous role, in PSG, 126
- PRF (Pseudo relevance feedback)
 - as alternative to query expansion, 445
 - overview of, 377
- Private states. *See also* Subjectivity and sentiment analysis, 260
- Probabilistic context-free grammars (PCFGs)
 - for ambiguity resolution, 80–83
 - dependency graphs in syntax analysis, 66–67
 - generative parsing models, 83–84
 - parsing techniques, 78
- Probabilistic latent semantic analysis (PLSA), 176–177
- Probabilistic models
 - document a priori models, 377
 - for document retrieval, 375
- Probability
 - history of, 83
 - MaxEnt formula for conditional probability, 316
- Productivity/creativity, and the unknown word problem, 13–15
- Projective dependency trees
 - overview of, 64–65
 - worst-case parsing algorithm for CFGs, 78
- Projectivity
 - in dependency analysis, 64
 - non projective dependency trees, 65–67
 - projective dependency trees, 64–65
- Prompt localization, spoken dialog systems, 514–516
- PropBank
 - annotation of, 447
 - dependency trees and, 130–132
 - limitation of, 122
 - Penn Treebank and, 123
 - as resource for predicate-argument recognition, 119–122
 - tagging text with arguments, 124
- Prosody
 - defined, 45
 - sentence and topic segmentation, 45–48
- Pseudo relevance feedback (PRF)
 - as alternative to query expansion, 445
 - overview of, 377
- PSG (Phrase Structure Grammar), 124–129
- Publications, resources for RTE, 252
- Punctuation
 - in PSG, 129
 - typographical and structural features for sentence and topic segmentation, 44–45
- PUNDIT, 122
- Pushdown automaton, in CFGs, 72
- Pyramid, for manual evaluation of summarization, 413–415
- QA. *See* Question answering (QA)
- QUALM QA system, 434
- Queries
 - evaluation in distillation, 492
 - preprocessing, 483
 - QA architectures and, 439
 - searching unstructured sources, 443–445
 - translating CLIR queries, 379
 - translating MLIR queries, 384

- Query answering distillation system
 - document retrieval, 483–484
 - overview of, 483
 - planning stage, 487
 - preprocessing queries, 483
 - snippet filtering, 484
 - snippet processing, 485–487
- Query expansion
 - applying to CLIR queries, 380
 - for improving information retrieval, 376–377
 - searching over unstructured sources, 445
- Query generation, in QA architectures, 435
- Query language, in CLIR, 365
- Question analysis, in QA, 435, 440–443
- Question answering (QA)
 - answer scores, 450–453
 - architectures, 435–437
 - bibliography, 467–473
 - candidate extraction from structured sources, 449–450
 - candidate extraction from unstructured sources, 445–449
 - case study, 455–460
 - challenges in, 464–465
 - crosslingual, 454–455
 - evaluating answer correctness, 461–462
 - evaluation tasks, 460–461
 - introduction to and history of, 433–435
 - IR compared with, 366
 - performance metrics, 462–464
 - question analysis, 440–443
 - RTE applied to, 215
 - searching over unstructured sources, 443–445
 - source acquisition and preprocessing, 437–440
 - summary, 465–467
- Question mark (?), sentence segmentation markers, 30
- Questions, in GALE distillation initiative, 475
- Quotation marks (“”), sentence segmentation markers, 30

- R summarization frameworks, 422
- RandLM toolkit, for machine translation, 357
- Random forest language models (RFLMs)
 - modeling using morphological categories, 193
 - tree-based modeling, 185–186
- Ranks methods, for sentences, 407
- RDF (Resource Description Framework), 450
- Real-Time Translation Services (RTTS), 538–540
- Realization stage, of summarization systems
 - building a summarization system and, 421
 - overview of, 400
- Recall, IR evaluation measures, 388
- Recall-Oriented Understudy for Gisting Evaluation (ROUGE)
 - automatic evaluation of summarization, 415–418
 - metrics in, 420
- Recognizing textual entailment (RTE)
 - alignment, 233–236
 - analysis, 220
 - answer scoring and, 464
 - applications of, 214
 - bibliography, 254–258
 - case studies, 238–239
 - challenge of, 212–213
 - comparing constituents in, 222–224
 - developing knowledge resources for, 249–251
 - discourse commitments extraction case study, 239–240
 - enrichment, 228–231
 - evaluating performance of, 213–214
 - framework for, 219
 - general model for, 224–227
 - graph generation, 231–232
 - implementation of, 227
 - improving analytics, 248–249
 - improving evaluation, 251–252
 - inference, 236–238
 - introduction to, 209–210
 - investing/applying to new problems, 249
 - latent alignment inference, 247–248
 - learning alignment independently of entailment, 244–245
 - leveraging multiple alignments, 245
 - limited dependency context for global similarity, 247
 - logical representation and inference, 242–244
 - machine translation, 217–218
 - multiview representation, 220–222
 - natural logic and, 245–246
 - in non-English languages, 218–219

- PASCAL challenge, 451
- preprocessing, 227–228
- problem definition, 210–212
- QA and, 215, 433–434
- requirements for RTE framework, 219–220
- resources for, 252–253
- searching for relations, 215–217
- summary, 253–254
- Syntactic Semantic Tree Kernels (SSTKs), 246–247
- training, 238
- transformation-based approaches to, 241–242
- tree edit distance case study, 240–241
- Recombination, machine translation and, 346
- Recursive transition networks (RTNs), 150
- Redundancy, in distillation
 - detecting, 492–493
 - overview of, 477–479
 - reducing, 489–490
- Redundancy, in IR, 488
- Reduplication of words, limits of finite-state models, 17
- Reference summaries, 412, 419
- Regular expressions
 - surface patterns for extracting candidate answers, 449
 - in type-based candidate extraction, 446
- Regular relations, finite-state transducers capturing and computing, 17
- Related terms, in GALE distillation initiative, 475
- Relation extraction systems
 - classification approach, 312–313
 - coreference resolution as, 311
 - features of classification-based systems, 313–316
 - kernel methods for, 319
 - overview of, 310
 - supervised and unsupervised, 317–319
- Relational databases, 449
- Relations
 - bibliography, 327–330
 - classifiers for, 316
 - combining entity and relation detection, 320
 - between constituents in RTE, 220
 - detection in Rosetta Consortium distillation system, 480–482
 - extracting, 310–313
 - features of classification-based extractors, 313–316
 - introduction to, 309–310
 - kernel methods for extracting, 319
 - recognition impacting searches, 444
 - summary, 326–327
 - supervised and unsupervised approaches to extracting, 317–319
 - transitive closure of, 324–326
 - types of, 311–312
- Relationship questions, QA and, 433, 488
- Relevance, feedback and query expansion, 376–377
- Relevance, in distillation
 - analysis of, 492–493
 - detecting, 488–489
 - examples of irrelevant answers, 477
 - overview of, 477–479
 - redundancy reduction and, 488–490
- Relevance, in IR
 - assessment, 387–388
 - evaluation, 386
- Remote operation, challenges in NLP aggregation, 524
- Resource Description Framework (RDF), 450
- Resources, for RTE
 - developing knowledge resources, 249–251
 - overview of, 252–253
- Restricted domains, history of QA systems, 434
- Result pooling, relevance assessment and, 387
- Rewrite rules (in phonology and morphology), 17
- RFLMs (Random forest language models)
 - modeling using morphological categories, 193
 - tree-based modeling, 185–186
- Rhetorical structure theory (RST), applying to summarization, 401–404
- RoboCup, for meaning representation, 149
- Robust processing
 - desired attributes of NLP aggregation, 526–527
 - in GATE, 529
 - in InfoSphere Streams, 531
 - in UIMA, 529
- Robust risk minimization (RRM), mention detection and, 287

Roget's Thesaurus

- semantic parsing, 104
 - word sense disambiguation, 106–107
- Role extractors, classifiers for relation extraction, 316
- Romanian
- approaches to subjectivity and sentiment analysis, 276–277
 - corpus-based approach to subjectivity and sentiment analysis, 271–272
 - cross-lingual projections, 275
 - dictionary-based approach to subjectivity and sentiment analysis, 264–266, 270
 - IR and, 390
 - QA and, 461
 - subjectivity and sentiment analysis, 259
 - summarization and, 399
- Romanization, transliteration of scripts to Latin (Roman) alphabet, 368
- Rosetta Consortium system
- document and corpus preparation, 480–483
 - indexing and, 483
 - overview of, 479–480
 - query answers and, 483–487
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation)
- automatic evaluation of summarization, 415–418
 - metrics in, 420
- RRM (robust risk minimization), mention detection and, 287
- RST (rhetorical structure theory), applying to summarization, 401–404
- RTNs (recursive transition networks), 150
- RTTS (Real-Time Translation Services), 538–540
- Rule-based grammars, in speech recognition, 501–502
- Rule-based sentence segmentation, 31–32
- Rule-based systems
- dictionary-based approach to subjectivity and sentiment analysis, 270
 - for meaning representation, 150
 - statistical models compared with, 292
 - subjectivity and sentiment analysis, 267
 - word and phrase-level annotations in subjectivity and sentiment analysis, 263
 - for word sense disambiguation, 105–109

- Rules, functional morphology models and, 19
- Russian
- language modeling using subword units, 192
 - parsing issues related to morphology, 91
 - unification-based models, 19

SALAAM algorithms, 114–115

SALSA project, for predicate-argument recognition, 122

Sanskrit

- ambiguity in, 11
 - as fusional language, 8
 - Zen toolkit for morphology of, 20
- SAPT (semantically augmented parse tree), 151

Scalable entailment relation recognition (SERR), 215–217

SCGIS (Sequential conditional generalized iterative scaling), 289

Scores

- ranking answers in QA, 435, 450–453, 458–459
- ranking sentences, 407
- sentence relevance in distillation systems, 485–486

Scripts

- preprocessing best practices in IR, 371–372
- transliteration and direction of, 368

SCUs (summarization content units), in

Pyramid method, 414–415

Search component, in QA architectures, 435

Searches

- broadening to overcome parsing errors, 144
- in mention detection, 289–291
- over unstructured sources in QA, 443–445
- QA architectures and, 439
- QA vs. IR, 433
- reducing search space using beam search, 290–291
- for relations, 215–217

SEE (Summary Evaluation Environment), 413

Seeds, unsupervised systems and, 112

Segmentation

- in aggregated NLP, 540
- sentence boundaries. See Sentence boundary detection
- topic boundaries. See Topic segmentation

Semantic concordance (SEMCOR) corpus, WordNet, 104

- Semantic interpretation
 - entity and event resolution, 100
 - meaning representation, 101
 - overview of, 98–99
 - predicate-argument structure and, 100
 - structural ambiguity and, 99
 - word sense and, 99–100
- Semantic parsing
 - Air Travel Information System (ATIS), 148
 - bibliography, 154–167
 - Communicator program, 148–149
 - corpora for, 104–105
 - entity and event resolution, 100
 - GeoQuery, 149
 - introduction to, 97–98
 - meaning representation, 101, 147–148
 - as part of semantic interpretation, 98–99
 - predicate-argument structure. *See*
 - Predicate-argument structure
 - resource availability for disambiguation of
 - word sense, 104–105
 - RoboCup, 149
 - rule-based systems, 105–109, 150
 - semi-supervised systems, 114–116
 - software programs for, 116–117, 151
 - structural ambiguity and, 99
 - summary, 151
 - supervised systems, 109–112, 150–151
 - system paradigms, 101–102
 - unsupervised systems, 112–114
 - word sense and, 99–100, 102–105
- Semantic role labeling (SRL). *See also*
 - Predicate-argument structure
 - challenges in RTE and, 212
 - combining dependency parsing with, 132
 - implementing RTE and, 227
 - overcoming independence assumption, 137–138
 - predicate-argument structure training, 447
 - in Rosetta Consortium distillation system, 480
 - in RTE, 221
 - sentences as processing unit in, 29
 - for shallow semantic parsing, 118
- Semantically augmented parse tree (SAPT), 151
- Semantics
 - defined, 97
 - explicit semantic analysis (ESA), 382
 - features of classification-based relation
 - extraction systems, 315–316
 - finding entity relations, 310
 - latent semantic indexing (LSI), 381
 - QA and, 439–440
 - structural matching and, 446–447
 - topic detection and, 33
- SEMCOR (semantic concordance) corpus,
 - WordNet, 104
- SEMEVAL, 263
- Semi-supervised systems, for word sense
 - disambiguation, 114–116
- Semistructured data, candidate extraction
 - from, 449–450
- SemKer system, applying syntactic tree
 - kernels to RTE, 246
- Sense induction, unsupervised systems and, 112
- SENSEVAL, for word sense disambiguation, 105–107
- Sentence boundary detection
 - comparing segmentation methods, 40–41
 - detecting probable sentence or topic
 - boundaries, 33–34
 - discourse features, 44
 - discriminative local classification method
 - for, 36–38
 - discriminative sequence classification
 - method for, 38–39
 - extensions for global modeling, 40
 - features of segmentation methods, 41–42
 - generative sequence classification method, 34–36
 - hybrid methods, 39–40
 - implementing RTE and, 227
 - introduction to, 29
 - lexical features, 42–43
 - overview of, 30–32
 - performance of, 41
 - processing stages of, 48
 - prosodic features, 45–48
 - speech-related features, 45
 - syntactic features, 43–44
 - typographical and structural features, 44–45
- Sentence-level annotations, for subjectivity
 - and sentiment analysis
 - corpus-based approach, 271–272
 - dictionary-based approach, 270–271
 - overview of, 269

- Sentence splitters, tools for building
 - summarization systems, 423
- Sentences
 - coherence of sentence-sentence connections, 402
 - extracting within-sentence relations, 310
 - methods for learning rank of, 407
 - parasitic gap recovery, 249
 - processing for event extraction, 323
 - relevance in distillation systems, 485–486
 - units in sentence segmentation, 33
 - unsupervised approaches to selection, 489
- Sentential complement, features of supervised systems, 111
- Sentential forms, parsing and, 71–72
- Sentiment analysis. See Subjectivity and sentiment analysis
- SentiWordNet, 262
- Sequential conditional generalized iterative scaling (SCGIS), 289
- SERR (scalable entailment relation recognition), 215–217
- Shallow semantic parsing
 - coverage in semantic parsing, 102
 - overview of, 98
 - semantic role labeling for, 118
 - structural matching and, 447
- Shalmaneser program, for semantic role labeling, 147
- Shift-reduce parsing, 72–73
- SHRDLU QA system, 434
- SIGHAN, Chinese word segmentation, 194
- SIGLEX (Special Group on LEXicon), 103
- Similarity enablement, relation extraction and, 310
- Slovene unification-based model, 19
- SLU (statistical language understanding)
 - continuous improvement cycle in dialog systems, 512–513
 - generations of dialog systems, 511–512
- Smoothing techniques
 - Laplace smoothing, 174
 - machine translation and, 345
 - n*-gram approximation, 172–173
- SMT. See Statistical machine translation (SMT)
- Snippets, in distillation
 - crosslingual distillation and, 491
 - evaluation, 492–493
 - filtering, 484
 - main and supporting, 477–478
 - multimodal distillation and, 490
 - planning and, 487
 - processing, 485–487
- Snowball Stemmer, 392
- Software programs
 - for meaning representation, 151
 - for predicate-argument structure, 147
 - for semantic parsing, 116–117
- Sort expansion, machine translation phrase decoding, 347–348
- Sources, in QA
 - acquiring, 437–440
 - candidate extraction from structured, 449–450
 - candidate extraction from unstructured, 445–449
 - searching over unstructured, 443–445
- Spanish
 - code switching example, 31, 195–196
 - corpus-based approach to subjectivity and sentiment analysis, 272
 - discriminative approach to parsing, 91–92
 - GeoQuery corpus translated into, 149
 - IR and, 390–391
 - localization of spoken dialog systems, 513–514, 517–520
 - mention detection experiments, 294–296
 - morphologies of, 20
 - polarity analysis of words and phrases, 269
 - QA and, 461
 - resources for semantic parsing, 122
 - RTE in, 218
 - semantic parser for, 151
 - summarization and, 398
 - TAC and, 424
 - TALES case study, 538
 - WordNet and, 109
- Special Group on LEXicon (SIGLEX), 103
- Speech
 - discourse features in topic or sentence segmentation, 44
 - lexical features in sentence segmentation, 42
 - prosodic features for sentence or topic segmentation, 45–48
 - sentence segmentation accuracy, 41

- Speech generation
 - dialog manager directing, 499–500
 - spoken dialog systems and, 503–504
- Speech recognition
 - anchored speech recognition, 490
 - automatic speech recognition (ASR), 29, 31
 - language modeling using subword units, 192
 - MaxEnt model applied to, 181–183
 - Morfessor package applied to, 191–192
 - neural network language models applied to, 188
 - rule-based grammars in, 501–502
 - spoken dialog systems and, 500–503
- Speech Recognition Grammar Specification (SRGS), 501–502
- Speech-to-text (STT)
 - data reorganization and, 535–536
 - in GALE IOD, 532–533
 - NLP and, 523–524
 - in RTTS, 538
- Split-head concept, in parsing, 78
- Spoken dialog systems
 - architecture of, 505
 - bibliography, 521–522
 - call-flow localization, 514
 - continuous improvement cycle in, 512–513
 - dialog manager, 504–505
 - forms of dialogs, 509–510
 - functional diagram of, 499–500
 - generations of, 510–512
 - introduction to, 499
 - localization of, 513–514
 - localization of grammars, 516–517
 - natural language call routing, 510
 - prompt localization, 514–516
 - speech generation, 503–504
 - speech recognition and understanding, 500–503
 - summary, 520–521
 - testing, 519–520
 - training, 517–519
 - transcription and annotation of utterances, 513
 - voice user interface (VUI), 505–509
- Spoken languages, vs. written languages and language models, 194–195
- SRGS (Speech Recognition Grammar Specification), 501–502
- SRILM (Stanford Research Institute Language Modeling)
 - overview of, 184
 - SRILM toolkit for machine translation, 357
- SRL. See Semantic role labeling (SRL)
- SSI (Structural semantic interconnections)
 - algorithm, 107–109
- SSTKs (Syntactic Semantic Tree Kernels), 246–247
- Stacks, of hypotheses in machine translation, 346
- Stanford Parser, dependency parsing with, 456
- Stanford Research Institute Language Modeling (SRILM)
 - overview of, 184
 - SRILM toolkit for machine translation, 357
- START QA system, 435–436
- Static knowledge, in textual entailment, 210
- Statistical language models
 - n*-gram approximation, 170–171
 - overview of, 169
 - rule-based systems compared with, 292
 - spoken vs. written languages and, 194–195
 - translation with, 331
- Statistical language understanding (SLU)
 - continuous improvement cycle in dialog systems, 512–513
 - generations of dialog systems, 511–512
- Statistical machine translation (SMT)
 - applying to CLIR, 381
 - cross-language mention propagation, 293–294
 - evaluating co-occurrence of words, 337–338
 - mention detection experiments, 293–294
- Stemmers
 - mapping terms to stems, 370
 - preprocessing best practices in IR, 371
 - Snowball Stemmer, 392
- Stems, mapping terms to, 370
- Stop-words, removing in normalization, 371
- Structural ambiguity, 99
- Structural features
 - of classification-based relation extraction systems, 314
 - sentence and topic segmentation, 44–45
- Structural matching, for candidate extraction in QA, 446–448

- Structural semantic interconnections (SSI)
 - algorithm, 107–109
- Structure
 - of documents. See Document structure
 - of words. See Word structure
- Structured data
 - candidate extraction from structured sources, 449–450
 - candidate extraction from unstructured sources, 445–449
- Structured knowledge, 434
- Structured language model, 181
- Structured queries, 444
- STT (Speech-to-text). See Speech-to-text (STT)
- Subcategorization
 - in PSG, 125
 - in TAG, 130
 - for verb sense disambiguation, 112
- Subclasses, of relations, 311
- Subject/object presence, features of supervised systems, 111
- Subject, object, verb (SOV) word order, 356
- Subjectivity, 260
- Subjectivity analysis, 260
- Subjectivity and sentiment analysis
 - applied to English, 262
 - bibliography, 278–281
 - comparing approaches to, 276–277
 - corpora for, 262–263
 - definitions, 260–261
 - document-level annotations, 272–274
 - introduction to, 259–260
 - lexicons and, 262
 - ranking approaches to, 274–276
 - sentence-level annotations, 269, 270–272
 - summary, 277
 - tools for, 263–264
 - word and phrase level annotations, 264–269
- Substitution, linguistic supports for cohesion, 401
- Subword units, selecting for language models, 191–192
- SUMMA
 - history of summarization systems, 399
 - for multilingual automatic summarization, 411
 - summarization frameworks, 423
- SUMMARIST, 398
- Summarization, automatic. See Automatic summarization
- Summarization content units (SCUs), in Pyramid method, 414–415
- Summary Evaluation Environment (SEE), 413
- SummBank
 - history of summarization systems, 399
 - summarization data set, 425
- Supertags, in TAG, 130
- Supervised systems
 - for meaning representation, 150–151
 - for relation extraction, 317–319
 - for sentence segmentation, 37
 - for word sense disambiguation, 109–112
- Support vector machines (SVMs)
 - classifiers for relation extraction, 316–317
 - corpus-based approach to subjectivity and sentiment analysis, 272, 274
 - mention detection and, 287
 - methods for sentence or topic segmentation, 37–39
 - training and test software, 135–137
 - unsupervised approaches to machine learning, 342
- Surface-based features, in automatic summarization, 400–401
- Surface patterns, for candidate extraction in QA, 448–449
- Surface strings
 - input words in input/output language relations, 17
 - unification-based morphology and, 18
- SVMs. See Support vector machines (SVMs)
- SVO (subject, verb, object) word order, 356
- Swedish
 - IR and, 390–391
 - morphologies of, 20
 - semantic parsing and, 122
 - summarization and, 399
- SwiRL program, for semantic role labeling, 147
- Syllabic scripts, 371
- Symmetrization, word alignment and, 340–341
- Syncretism, 8
- Synonyms
 - answers in QA systems and, 442
 - machine translation metrics and, 336

- Syntactic features
 - of classification-based relation extraction systems, 315
 - of coreference models, 301
 - of mention detection system, 292
 - in sentence and topic segmentation, 43–44
- Syntactic models, for machine translation, 352–354
- Syntactic pattern, in PSG, 126
- Syntactic relations, features of supervised systems, 111
- Syntactic representation, in
 - predicate-argument structure, 123–124
- Syntactic roles, in TAG, 130
- Syntactic Semantic Tree Kernels (SSTKs), 246–247
- Syntactic Structures* (Chomsky), 98–99
- Syntax
 - ambiguity resolution, 80
 - bibliography, 92–95
 - compared with morphology and phonology and orthography, 3
 - context-free grammar (CFGs) and, 59–61
 - dependency graphs for analysis of, 63–67
 - discriminative parsing models, 84–87
 - of documents in IR, 367–368
 - generative parsing models, 83–84
 - introduction to, 57
 - minimum spanning trees and dependency parsing, 79–80
 - morphology and, 90–92
 - parsing algorithms for, 70–72
 - parsing natural language, 57–59
 - phrase structure trees for analysis of, 67–70
 - probabilistic context-free grammars, 80–83
 - QA and, 439–440
 - shift-reduce parsing, 72–73
 - structural matching and, 446–447
 - summary, 92
 - tokenization, case, and encoding and, 87–89
 - treebanks data-driven approach to, 61–63
 - word segmentation and, 89–90
 - worst-case parsing algorithm for CFGs, 74–79
- Syntax-based language models, 180–181
- Synthetic languages, morphological typology and, 7
- System architectures
 - for distillation, 488
 - for semantic parsing, 101–102
- System paradigms, for semantic parsing, 101–102
- Systran’s Babelfish program, 331
- TAC. See Text Analysis Conferences (TAC)
- TAG (Tree-Adjoining Grammar), 130
- TALES (Translingual Automated Language Exploitation System), 538
- Tamil
 - as agglutinative language, 7
 - IR and, 390
- Task-based evaluation, of translation, 334
- TBL (transformation-based learning), for sentence segmentation, 37
- TDT (Topic Detection and Tracking) program, 32–33, 42, 425–426
- Telugu, 390
- Templates, in GALE distillation initiative, 475
- Temporal cue words, in PSG, 127–128
- TER (Translation-error rate), 337
- Term-document matrix, document representation in monolingual IR, 373
- Term frequency-inverse document frequency (TF-IDF)
 - multilingual automatic summarization and, 411
 - QA scoring and, 450–451
 - unsupervised approaches to sentence selection, 489
- Term frequency (TF)
 - TF document model, 373
 - unsupervised approaches to sentence selection, 489
- Terms
 - applying RTE to unknown, 217
 - early approaches to summarization and, 400
 - in GALE distillation initiative, 475
 - mapping term vectors to topic vectors, 381
 - mapping to lemmas, 370
 - posting lists, 373–374
- Terrier IR framework, 392
- Text Analysis Conferences (TAC)
 - competitions related to summarization, 424
 - data sets related to summarization, 425

- Text Analysis Conferences (TAC) (*continued*)
 - evaluation of QA systems, 460–464
 - history of QA systems, 434
 - Knowledge Base Population (KBP), 481–482
 - learning summarization, 408
- Text REtrieval Conference (TREC)
 - data sets for evaluating IR systems, 389–390
 - evaluation of QA systems, 460–464
 - history of QA systems, 434
 - redundancy reduction, 489
- Text Tiling method (Hearst)
 - sentence segmentation, 42
 - topic segmentation, 37–38
- Text-to-speech (TTS)
 - architecture of spoken dialog systems, 505
 - history of dialog managers, 504
 - localization of grammars and, 514
 - in RTTS, 538
 - speech generation, 503–504
- TextRank, graphical approaches to automatic summarization, 404–406
- Textual entailment. *See also* Recognizing textual entailment (RTE)
 - contradiction in, 211
 - defined, 210
 - entailment pairs, 210
- Textual inference
 - implementing, 236–238
 - latent alignment inference, 247–248
 - modeling, 226–227
 - NLP and, 209
 - RTE and, 242–244
- TF-IDF (term frequency-inverse document frequency)
 - multilingual automatic summarization and, 411
 - QA scoring and, 450–451
 - unsupervised approaches to sentence selection, 489
- TF (term frequency)
 - TF document model, 373
 - unsupervised approaches to sentence selection, 489
- Thai
 - as isolating or analytic language, 7
 - word segmentation in, 4–5
- Thot program, for machine translation, 423
- Tika (Content Analysis Toolkit), for preprocessing IR documents, 392
- TinySVM software, for SVM training and testing, 135–136
- Token streams, 372–373
- Tokenization
 - Arabic, 12
 - character *n*-gram models and, 370
 - multilingual automatic summarization and, 410
 - normalization and, 370–371
 - parsing issues related to, 87–88
 - phrase indices and, 369–370
 - in Rosetta Consortium distillation system, 480
 - word segmentation and, 369
- Tokenizers, tools for building summarization systems, 423
- Tokens
 - lexical features in sentence segmentation, 42–43
 - mapping between scripts (normalization), 370–371
 - MLIR indexes and, 384
 - output from information retrieval, 366
 - processing stages of segmentation tasks, 48
 - in sentence segmentation, 30
 - translating MLIR queries, 384
 - in word structure, 4–5
- Top-k models, for monolingual information retrieval, 374
- Topic-dependent language model adaptation, 176
- Topic Detection and Tracking (TDT) program, 32–33, 42, 425–426
- Topic or domain, features of supervised systems, 111
- Topic segmentation
 - comparing segmentation methods, 40–41
 - discourse features, 44
 - discriminative local classification method, 36–38
 - discriminative sequence classification method, 38–39
 - extensions for global modeling, 40
 - features of, 41–42
 - generative sequence classification method, 34–36
 - hybrid methods, 39–40

- introduction to, 29
- lexical features, 42–43
- methods for detecting probable topic boundaries, 33–34
- overview of, 32–33
- performance of, 41
- processing stages of segmentation tasks, 48
- prosodic features, 45–48
- speech-related features, 45
- syntactic features, 43–44
- typographical and structural features, 44–45
- Topics, mapping term vectors to topic vectors, 381
- Traces nodes, Treebanks, 120–121
- Training
 - issues related to machine translation (MT), 197
 - minimum error rate training (MERT), 349
 - phrase-based models, 344–345
 - predicate-argument structure, 140–141, 447
 - recognizing textual entailment (RTE), 238
 - in RTE, 238
 - spoken dialog systems, 517–519
 - stage of RTE model, 238
 - support vector machines (SVMs), 135–137
- Transcription
 - of utterances based on rule-based grammars, 502–503
 - of utterances in spoken dialog systems, 513
- Transducers, finite-state, 16–17
- Transformation-based approaches, applying to RTE, 241–242
- Transformation-based learning (TBL), for sentence segmentation, 37
- Transformation stage, of summarization systems, 400, 421
- Transitive closure, of relations, 324–326
- Translation
 - human assessment of word meaning, 333–334
 - by machines. See Machine translation (MT)
 - translation-based approach to CLIR, 378–380
- Translation-error rate (TER), 337
- Translingual Automated Language Exploitation System (TALES), 538
- Translingual information retrieval, 491
- Translingual summarization. See also Automatic summarization, 398
- Transliteration, mapping text between scripts, 368
- TREC. See Text REtrieval Conference (TREC)
- trec-eval, evaluation of IR systems, 393
- Tree-Adjoining Grammar (TAG), 130
- Tree-based language models, 185–186
- Tree-based models, for MT
 - chart decoding, 351–352
 - hierarchical phrase-based models, 350–351
 - linguistic choices and, 354
 - overview of, 350
 - syntactic models, 352–354
- Tree edit distance, applying to RTE, 240–241
- Treebanks
 - data-driven approach to syntactic analysis, 61–63
 - dependency graphs in syntax analysis, 63–67
 - phrase structure trees in syntax analysis, 67–70
 - traces nodes marked as arguments in PropBank, 120–121
 - worst-case parsing algorithm for CFGs, 77
- Trigger models, dynamic self-adapting language models, 176–177
- Triggers
 - consistency of, 323
 - finding event triggers, 321–322
- Trigrams, 502–503
- Troponymy, 310
- Tuning sets, 348
- Turkish
 - dependency graphs in syntax analysis, 62, 65
 - GeoQuery corpus translated into, 149
 - language modeling for morphologically rich languages, 189–191
 - language modeling using morphological categories, 192–193
 - machine translation and, 354
 - morphological richness of, 355
 - parsing issues related to morphology, 90–91
 - semantic parser for, 151
 - syntactic features used in sentence and topic segmentation, 43

- Type-based candidate extraction, in QA, 446, 451
- Type classifier
 - answers in QA systems, 440–442
 - in relation extraction, 313
- Type system, GALE Type System (GTS), 534–535
- Typed feature structures, unification-based morphology and, 18–19
- Typographical features, sentence and topic segmentation, 44–45
- Typology, morphological, 7–8

- UCC (UIMA Component Container), 537
- UIMA. See Unstructured Information Management Architecture (UIMA)
- Understanding, spoken dialog systems and, 500–503
- Unicode (UTF-8/UTF-16)
 - encoding and script, 368
 - parsing issues related to encoding systems, 89
- Unification-based morphology, 18–19
- Unigram models (Yamron), 35–36
- Uninflectedness, homonyms and, 12
- Units of thought, interlingual document representations, 381
- Unknown terms, applying RTE to, 217
- Unknown word problem, 8, 13–15
- Unstructured data, candidate extraction from, 445–449
- Unstructured Information Management Architecture (UIMA)
 - attributes of, 528–529
 - GALE IOD and, 535, 537
 - overview of, 527–528
 - RTTS and, 538–540
 - sample code, 542–547
 - summarization frameworks, 422
 - UIMA Component Container (UCC), 537
- Unstructured text, history of QA systems and, 434
- Unsupervised adaptation, language model adaptation and, 177
- Unsupervised systems
 - machine learning, 342
 - relation extraction, 317–319
 - sentence selection, 489
 - subjectivity and sentiment analysis, 264
 - word sense disambiguation, 112–114
- Update summarization, in automatic summarization, 397
- Uppercase (capitalization), sentence segmentation markers, 30
- UTF-8/UTF-16 (Unicode)
 - encoding and script, 368
 - parsing issues related to encoding systems, 89
- Utterances, in spoken dialog systems
 - rule-based approach to transcription and annotation, 502–503
 - transcription and annotation of, 513

- Variable-length language models, 179
- Vector space model
 - document representation in monolingual IR, 372–373
 - for document retrieval, 374–375
- Verb clustering, in PSG, 125
- Verb sense, in PSG, 126–127
- Verb, subject, object (VSO) word order, 356
- VerbNet, resources for predicate-argument recognition, 121
- Verbs
 - features of predicate-argument structures, 145
 - relation extraction and, 310
- Vietnamese
 - as isolating or analytic language, 7
 - NER task in, 287
- Views
 - in GALE IOD, 534
 - RTE systems, 220
- Vital few (80/20 rule), 14
- Viterbi algorithm
 - applied to Rosetta Consortium distillation system, 480
 - methods for sentence or topic segmentation, 39–40
 - searching for mentions, 291
- Vocabulary
 - indexing IR output, 366
 - language models and, 169
 - in morphologically rich languages, 190
 - productivity/creativity and, 14
 - topic segmentation methods, 38

- Voice Extensible Markup Language. See VoiceXML (Voice Extensible Markup Language)
- Voice feature, in PSG, 124
- Voice of sentence, features of supervised systems, 111
- Voice quality, prosodic modeling and, 47
- Voice user interface (VUI)
 - call-flow, 505–506
 - dialog module (DM) of, 507–508
 - getService process of, 506–507
 - grammars of, 508–509
 - VUI completeness principle, 509–510
- VoiceXML (Voice Extensible Markup Language)
 - architecture of spoken dialog systems, 505
 - generations of dialog systems, 511–512
 - history of dialog managers, 504
- VUI. See Voice user interface (VUI)

- W3C (World Wide Web Consortium), 504
- WASP program, for rule-based semantic parsing systems, 151
- Web 2.0, accelerating need for crosslingual retrieval, 365
- WER (word-error rate), machine translation metrics and, 336–337
- Whitespace
 - preprocessing best practices in IR, 371
 - in word separation, 369
- Wikipedia
 - answer scores in QA and, 452
 - for automatic word sense disambiguation, 115–116
 - crosslingual question answering and, 455
 - as example of explicit semantic analysis, 382
 - predominance of English in, 438
- WikiRelate! program, for word sense disambiguation, 117
- Wiktionary
 - crosslingual question answering and, 455
 - as example of explicit semantic analysis, 382
- Witten-Bell smoothing technique, in language model estimation, 172
- Wolfram Alpha QA system, 435
- Word alignment, cross-language mention propagation, 293
- Word alignment, in MT
 - alignment models, 340
 - Berkeley word aligner, 357
 - co-occurrence of words between languages, 337–338
 - EM algorithm, 339–340
 - IBM Model 1, 338–339
 - as machine learning problem, 341–343
 - overview of, 337
 - symmetrization, 340–341
- Word boundary detection, 227
- Word-error rate (WER), machine translation metrics and, 336–337
- Word lists. See Dictionary-based morphology
- Word meaning
 - automatic evaluation, 334–335
 - evaluation of, 332
 - human assessment of, 332–334
- Word order, 356
- Word/phrase-level annotations, for subjectivity and sentiment analysis
 - corpus-based approach, 267–269
 - dictionary-based approach, 264–267
 - overview of, 264
- Word segmentation
 - in Chinese, Japanese, Thai, and Korean writing systems, 4–5
 - languages lacking, 193–194
 - phrase indices based on, 369–370
 - preprocessing best practices in IR, 371
 - syntax and, 89–90
 - tokenization and, 369
- Word sense
 - classifying according to subjectivity and polarity, 261
 - disambiguation, 105, 152–153
 - overview of, 102–104
 - resources, 104–105
 - rule-based systems, 105–109
 - semantic interpretation and, 99–100
 - semi-supervised systems, 114–116
 - software programs for, 116–117
 - supervised systems, 109–112
 - unsupervised systems, 112–114
- Word sequence, 169
- Word structure
 - ambiguity in interpretation of expressions, 10–13

- Word structure (*continued*)
- automated morphology (morphology induction), 21
 - bibliography, 22–28
 - dictionary-based morphology, 15–16
 - finite-state morphology, 16–18
 - functional morphology, 19–21
 - introduction to, 3–4
 - irregularity in linguistic models, 8–10
 - issues and challenges, 8
 - lexemes, 5
 - morphemes, 5–7
 - morphological models, 15
 - morphological typology, 7–8
 - productivity/creativity and the unknown word problem, 13–15
 - summary, 22
 - tokens and, 4–5
 - unification-based morphology, 18–19
 - units in sentence segmentation, 33
- WordNet
- classifying word sense according to subjectivity and polarity, 261
 - eXtended WordNet (XWN), 451
 - features of supervised systems, 112
 - hierarchical concept information in, 109
 - QA answer scores and, 452
 - as resource for domain-specific information, 122
 - RTE applied to machine translation, 218
 - SEMCOR (semantic concordance) corpus, 104–105
 - subjectivity and sentiment analysis lexicons, 262
 - synonyms, 336
 - word sense disambiguation and, 117
- World Wide Web Consortium (W3C), 504
- Written languages, vs. spoken languages in language modeling, 194–195
- WSJ, 147
- XDC (Crossdocument coreference), in Rosetta Consortium distillation system, 482–483
- Xerox Finite-State Tool (XFST), 16
- XWN (eXtended WordNet), 451
- YamCha software, for SVM training and testing, 135–136
- Yarowsky algorithm, for word sense disambiguation, 114–116
- Z-score normalization, for MLIR aggregation, 385
- Zen toolkit for morphology, applying to Sanskrit, 20
- Zero anaphora resolution, 249, 444