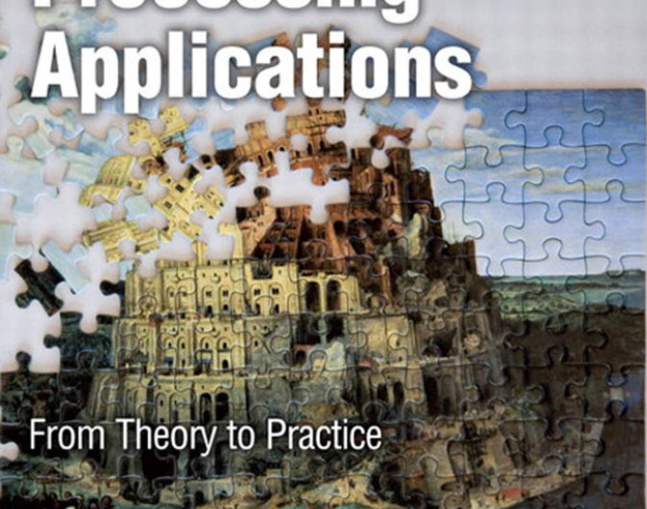
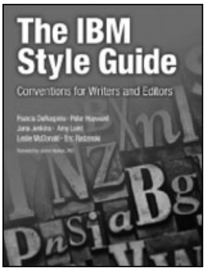
Multilingual Natural Language Processing Applications



From Theory to Practice

Edited by Daniel M. Bikel and Imed Zitouni

Related Books of Interest



The IBM Style Guide

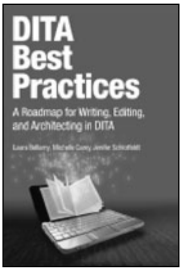
Conventions for Writers and Editors

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
A Roadmap for Writing, Editing, and Architecting in DITA
Laura Bellamy, Michelle Carey, Jenifer Schlotfeldt

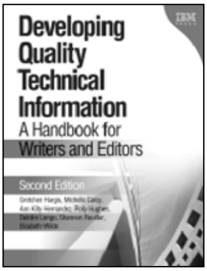
DITA Best Practices

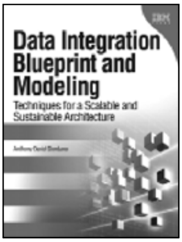
By Laura Bellamy, Michelle Carey, and Jenifer Schlotfeldt

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

Search Engine Marketing, Inc.
Driving Search Traffic to Your Company's Web Site
Second Edition

Search Engine Marketing, Inc.

By Mike Moran and Bill Hunt

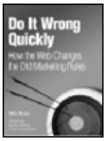
ISBN: 0-13-606868-5

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

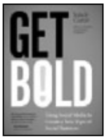


Do It Wrong Quickly

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

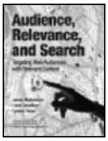
ISBN: 0-13-261831-1

The Social Factor

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

Azua

ISBN: 0-13-701890-8

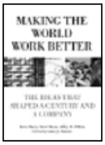


Audience, Relevance, and Search

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

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.

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

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

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

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 information 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 interfactes, and foundational problems in information theory, and he coedited *Image Databases: Search and Retrieval of Digital Imagery* (Wiley, 2002).

Jennifer Chu-Carroll (jencc@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 Labratories. 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).

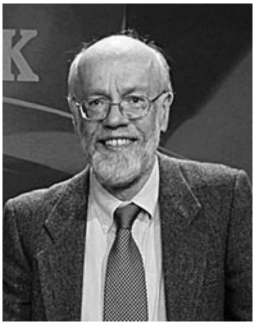
Benoit Favre (benoit.favre@lif.univ-mrs.fr) is an associate professor at Aix-Marseille Université, Marseille, France. He is a researcher in the field of natural language understanding. His research interests are in speech and text understanding with a focus on machine learning approaches. He received 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.

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.robertopieraccini.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 Conversational 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 Autonomus 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-adjoining 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-adjoining 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 WestLawNext, 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 interpretating 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, AAAI 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.te[2]
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: Viděl as ho? Neviděla jsem ho. Neviděla jsem nikoho.
saw+you-are him? not-saw I-am him. not-saw I-am no-one.

Here, *viděl as* 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 *Nikohos 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 morphophonemic. 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-	*po-ass-*	봤-	*pwass-*	'have seen'
(b)	가지었-	*ka.ci-ess-*	가졌-	*ka.cyess-*	'have taken'
(c)	하였-	*ha-yess-*	했-	*hayss-*	'have done'
(d)	되었-	*toy-ess-*	됐-	*twayss-*	'have become'
(e)	놓았-	*noh-ass-*	놨-	*nwass-*	'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.

Example 1–6: `hl stqrO h*h AljrA}d?`[3] هل ستقرأ هذه الجرائد؟
hal sa-taqraʾu hāḏihi ʾl-ǧarāʾida?
whether will+you-read this the-newspapers?

`hl stqrWhA? ln OqrOhA.` هل ستقرؤها؟ لن أقرأها.
hal sa-taqraʾuhā? lan ʾaqraʾahā.
whether will+you-read+it? not-will I-read+it.

[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 *hāḏihi ’l-ǧarāʾida* refers to ‘these newspapers’. While *sa-taqraʾu* ‘you will read’ combines the future marker *sa-* with the imperfective second-person masculine singular verb *taqraʾu* in the indicative mood and active voice, *sa-taqraʾuhā* ‘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 *taqraʾu* ‘you-MASC-SG read’ belongs is *qaraʾ*, 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 ʾ*, 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 *fʿal* and concatenate it, under rules of morphophonemic changes, with further prefixes and suffixes. The structure of *taqraʾu* is thus parsed into the template *ta-fʿal-u* and the invariant root.

The word *al-ǧarāʾida* ‘the newspapers’ in the accusative case and definite state is another example of internal inflection. Its structure follows the template *al-faʿāʾil-a* with the root *ǧ r d*. This word is the plural of *ǧarīdah* ‘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 *ǧarīdah* ‘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, *ǧadīd* becomes *ǧadīdah* ‘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: `hl rOyth? lm Orh. lm Or OHdA.` هل رأيته؟ لم أره. لم أر أحدا.
hal raʾaytihi? lam ʾarahu. lam ʾara ʾaḥadan.
whether you-saw+him? not-did I-see+him. not-did I-see anyone.

In Example 1–7, *raʾayti* is the second-person feminine singular perfective verb in active voice, member of the *raʾā* 'to see' lexeme of the *r ʾ y* root. The prototypical, regularized pattern for this citation form is *faʿal*, as we saw with *qaraʾ* in Example 1–6. Alternatively, we could assume the pattern of *raʾā* to be *faʿā*, thereby asserting in a compact way that the final root consonant and its vocalic context are subject to the particular phonological change, resulting in *raʾā* like *faʿā* instead of *raʾay* like *faʿal*. 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 *qaraʾ* 'to read' and *raʾā* '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	
qaraʾ	*qaraʾa*	*qaraʾti*	*qaraʾū*	*taqraʾu*	*ʾaqraʾa*	*ʾaqraʾ*	*qraʾ*	S
faʿal	*faʿal-a*	*faʿal-ti*	*faʿal-ū*	*ta-fʿal-u*	*ʾa-fʿal-a*	*ʾa-fʿal*	*fʿal*	I
faʿal	*faʿal-a*	*faʿal-ti*	*faʿal-ū*	*ta-fʿal-u*	*ʾa-fʿal-a*	*ʾa-fʿal-*	*fʿal*	M
...	***...-a***	***...-ti***	***...-ū***	***ta-...-u***	***ʾa-...-a***	***ʾa-...-***	***...***	
faʿā	*faʿā-a*	*faʿā-ti*	*faʿā-ū*	*ta-fā-u*	*ʾa-fā-a*	*ʾa-fā-*	*fā*	M
faʿā	*faʿā*	*faʿal-ti*	*faʿ-aw*	*ta-fā*	*ʾa-fā*	*ʾa-fa*	*fā*	I
raʾā	*raʾā*	*raʾayti*	*raʾaw*	*tarā*	*ʾarā*	*ʾara*	***rā***	S

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

Base Form		(-*e*)		Meaning	Comment
집-	*cip-*	집어	*cip.e*	‘pick’	regular
깁-	*kip-*	기워	*ki.we*	‘sew’	*p*-irregular
믿-	*mit-*	믿어	*mit.e*	‘believe’	regular
싣-	*sit-*	실어	*sil.e*	‘load’	*t*-irregular
씻-	*ssis-*	씻어	*ssis.e*	‘wash’	regular
잇-	*is-*	이어	*i.e*	‘link’	*s*-irregular
낳-	*nah-*	낳아	*nah.a*	‘bear’	regular
까맣-	*kka.mah-*	까매	*kka.may*	‘be black’	*h*-irregular
치르-	*chi.lu-*	치러	*chi.le*	‘pay’	regular *u*-ellipsis
이르-	*i.lu-*	이르러	*i.lu.le*	‘reach’	*le*-irregular
흐르-	*hu.lu-*	흘러	*hul.le*	‘flow’	*lu*-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	***mwut.ko***	묻어	*mwut.e*	묻은	*mwut.un*	‘bury’
묻고	***mwut.ko***	물어	***mwul.e***	물은	*mwul.un*	‘ask’
물고	*mwul.ko*	물어	***mwul.e***	문	*mwun*	‘bite’
걷고	***ket.ko***	걷어	*ket.e*	걷은	*ket.un*	‘roll up’
걷고	***ket.ko***	걸어	***kel.e***	걸은	*kel.un*	‘walk’
걸고	*kel.ko*	걸어	***kel.e***	건	*ken*	‘hang’
굽고	***kwup.ko***	굽어	*kwup.e*	굽은	*kwup.un*	‘be bent’
굽고	***kwup.ko***	구워	*kwu.we*	구운	*kwu.wun*	‘bake’
이르고	***i.lu.ko***	이르러	*i.lu.le*	이른	***i.lun***	‘reach’
이르고	***i.lu.ko***	일러	*il.le*	이른	***i.lun***	‘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**

dirāsatī	دراستي	`drAsty`	→	*dirāsatu ī*	دراسة ي	`drAsp y`
			→	*dirāsati ī*	دراسة ي	`drAsp y`
			→	*dirāsata ī*	دراسة ي	`drAsp y`
muʿallimīya	معلمي	`mElmy`	→	*muʿallimū ī*	معلمو ي	`mElmw y`
			→	*muʿallimī ī*	معلمي ي	`mElmy y`
katabtumūhā	كتبتموها	`ktbtmwhA`	→	*katabtum hā*	كتبتم ها	`ktbtm hA`
ʾiǧrāʾuhu	إجراؤه	`IjrAWh`	→	*ʾiǧrāʾu hu*	إجراء ه	`IjrA' h`
ʾiǧrāʾihi	إجرائه	`IjrA}h`	→	*ʾiǧrāʾi hu*	إجراء ه	`IjrA' h`
ʾiǧrāʾahu	إجراءه	`IjrA'h`	→	*ʾiǧrāʾa hu*	إجراء ه	`IjrA' h`
li-'l-ʾasafi	للأسف	`llOsf`	→	*li 'l-ʾasafi li*	ل الأسف	`l AlOsf`

Figure 1–1: Complex tokenization and normalization of euphony in Arabic. Three nominal cases are expressed by the same word form with *dirāsatī* 'my study' and *muʿallimīya* '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 *ʾiǧrāʾ* '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:	*vnuk* 'grandson'	*pravnuk* 'great-grandson'
		prapra...vnuk 'great-great-...grandson'
	les 'forest'	*prales* 'jungle', 'virgin forest'
	zdroj 'source'	*prazdroj* 'urquell', 'original source'
	starý 'old'	*prastarý* '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 *ǧūǧil* '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*, *pravnuk*, *prapravnuk*, ... 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*:

$$\mathcal{R} \ :: \ [\Sigma] \to \{[\Sigma]\} \qquad\qquad \mathcal{R} \ :: \ \mathit{String} \to \{\mathit{String}\} \tag{1.1}$$

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 Sedlâĉek 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):

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

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:

$$\mathcal{I} :: lexeme \rightarrow \{parameter\} \rightarrow \{form\} \tag{1.3}$$

$$\mathcal{D} :: lexeme \rightarrow \{parameter\} \rightarrow \{lexeme\} \tag{1.4}$$

$$\mathcal{L} :: content \rightarrow \{lexeme\} \tag{1.5}$$

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" <| [                                                      d r y دري
FaCY                    `verb`  [ "know", "notice" ]                       faʿā
     `imperf`  FCI,                                                         fʿī
FACY                    `verb`  [ "flatter", "deceive" ],                  fāʿā
HaFCY                   `verb`  [ "inform", "let know" ],                  ʾafʿā
lA >| "'a" >>| FCI |<< "ly"    `adj`  [ "agnostic" ],              lā-ʾa-fʿī-īy
FiCAL |< aT             `noun`  [ "knowledge", "knowing" ],            fiʿāl-ah
MuFACY |< aT            `noun`  [ "flattery" ]                       mufāʿā-ah
     `plural`  MuFACY |< At,                                          mufāʿā-āt
FACI                    `adj`   [ "aware", "knowing" ] ]                   fāʿī
```

know, notice	I (*i*)	*darā* درى	knowledge, knowing	*dirāyah* دراية
flatter, deceive	III	*dārā* دارى	flattery	*mudārāh* مداراة
inform, let know	IV	*ʾadrā* أدرى		(*mudārayāt* مداريات)
agnostic		*lā-ʾadrīy* لاأدري	aware, knowing	*dārin* دار

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/elixer-fm/.

[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. Koskenniemi, "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 trancducer 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. Megerdoomian, "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, "Acquistion 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 Inuktikut," 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