Helsinki University of Technology
Dissertations in Computer and Information Science
Espoo 2006 Report D13

INDUCTION OF THE MORPHOLOGY OF NATURAL LANGUAGE: UNSUPERVISED MORPHEME SEGMENTATION WITH APPLICATION TO AUTOMATIC SPEECH RECOGNITION

Mathias Creutz

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 15th of June, 2006, at 12 o'clock noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
P.O.Box 5400
FIN-02015 HUT
FINLAND

Distribution:
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O.Box 5400
FIN-02015 HUT
FINLAND
Tel. +358-9-451 3272
Fax +358-9-451 3277
http://www.cis.hut.fi

Available in pdf format at http://lib.hut.fi/Diss/2006/isbn9512282119/

ISBN 951-22-8210-0 (printed version)
ISBN 951-22-8211-9 (electronic version)
ISSN 1459-7020
Otamedia Oy
Espoo 2006

Creutz, M. (2006): **Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition**. Doctoral thesis, Helsinki University of Technology, Dissertations in Computer and Information Science, Report D13, Espoo, Finland.
Keywords: morpheme segmentation, morphology induction, unsupervised learning, probabilistic models, concatenative morphology, agglutinative languages, unlimited vocabulary speech recognition, Finnish, English

ABSTRACT

In order to develop computer applications that successfully process natural language data (text and speech), one needs good models of the vocabulary and grammar of as many languages as possible. According to standard linguistic theory, words consist of morphemes, which are the smallest individually meaningful elements in a language. Since an immense number of word forms can be constructed by combining a limited set of morphemes, the capability of understanding and producing new word forms depends on knowing which morphemes are involved (e.g., "water, water+s, water+y, water+less, water+less+ness, sea+water").

Morpheme boundaries are not normally marked in text unless they coincide with word boundaries. The main objective of this thesis is to devise a method that discovers the likely locations of the morpheme boundaries in words of any language. The method proposed, called *Morfessor*, learns a simple model of concatenative morphology (word forming) in an unsupervised manner from plain text. Morfessor is formulated as a Bayesian, probabilistic model. That is, it does not rely on predefined grammatical rules of the language, but makes use of statistical properties of the input text.

Morfessor situates itself between two types of existing unsupervised methods: morphology learning vs. word segmentation algorithms. In contrast to existing morphology learning algorithms, Morfessor can handle words consisting of a varying and possibly high number of morphemes. This is a requirement for coping with highly-inflecting and compounding languages, such as Finnish. In contrast to existing word segmentation methods, Morfessor learns a simple grammar that takes into account sequential dependencies, which improves the quality of the proposed segmentations.

Morfessor is evaluated in two complementary ways in this work: directly by comparing to linguistic reference morpheme segmentations of Finnish and English words and indirectly as a component of a large (or virtually unlimited) vocabulary Finnish speech recognition system. In both cases, Morfessor is shown to outperform state-of-the-art solutions.

The linguistic reference segmentations were produced as part of the current work, based on existing linguistic resources. This has resulted in a morphological gold standard, called *Hutmegs*, containing analyses of a large number of Finnish and English word forms.

Preface

Little did I know, as a boy and teenager, that some of my "hobbies" at that time would materialize in a doctoral thesis. In elementary school, my Finnish teacher once asked me, since I easily remembered the names of the nominal cases in Finnish, whether I was interested in grammar? I found this absurd: Why on earth would anyone be *interested* in grammar? Nonetheless, I guess I was. I was also interested in computer programming. The fascinating thing about computer games was how to make the computer a skillful counterplayer; actually playing the games was less captivating. Then as now, I had an interest for mathematics, which was more of an applied than theoretical nature. Maths was a tool for the creation of beautiful things, such as colorful pictures of fractals.

I have had the pleasure to work in close cooperation with several people, who have contributed tremendously to this thesis. The papers we have published together constitute the principal part of the work. In addition, I wish to recognize the valuable contributions of many others, whose names do not appear in the list of co-authors.

My supervisor, D. Sc. Krista Lagus suggested to me that morphology can be learned using unsupervised methods. I am grateful to Krista for teaching me many things about adaptive models. She set ambitious goals and consistently pushed me toward mathematically more sound and elegant model formulation. In her inclination toward perfectionism, Krista set a shining example how to write scientific publications.

My other supervisor, Doc. Mikko Kurimo contributed with a complementary, more application-oriented approach. In managing the Speech Group at the Laboratory of Computer and Information Science (CIS), Mikko is an encouraging leader. He has promoted the work in wider circles, especially through his leading role in arranging the so-called Morpho Challenge competition sponsored by the EU PASCAL network. Mikko's help and support also made my research visit to Berkeley possible.

Dr. Krister Lindén has been my mentor for many years, both during my time at Lingsoft, Inc. and afterwards when we were both working on our doctoral theses. It was Krister who got me involved in the field of speech recognition, when Lingsoft started to develop a Finnish speech recognizer in 1998. During our numerous and long discussions I have learned enormously.

I greatly appreciate Prof. Erkki Oja, head of the CIS laboratory, for his experienced leadership that guarantees that things run smoothly and efficiently. Doc. Timo Honkela has inspired me with his creative ideas and his openness to matters beyond the daily routine work. Timo also introduced me to the world of sushi. The cooperation and

exchange of ideas with the Speech Group has been essential for carrying out the present work. In particular, I would like to recognize the efforts of Teemu Hirsimäki and Vesa Siivola. The first-class work carried out by Sami Virpioja has been very valuable; among other things, he implemented the interactive web demo.

I am grateful to all the anonymous reviewers who have contributed to the quality of the publications through their insightful comments. My sincere thanks go to Assist. Prof. Richard Wicentowski and Doc. Jukka Heikkonen for pre-examining and commenting on this thesis. I particularly appreciate Richard's many suggestions for improvements, related both to content and language, as well as his stimulating company while giving a course in Helsinki in 2005.

My colleagues and the staff at the CIS laboratory have helped me on numerous occasions and I have enjoyed their company very much. My fellow students at the Finnish Graduate School of Language Technology (KIT) have been a great group, and we have had so much fun together. I am deeply indebted to the KIT Graduate School for funding this work, and I really appreciate the friendly and helpful professors and staff. I would also like to remember the people at the International Computer Science Institute (ICSI) in Berkeley for interesting discussions in the very final phase of this thesis work.

When it comes to a long-term project such as a doctoral thesis, distraction from work is a vital part of actually getting the work done. Fortunately, I have a wonderful family and great friends to whom I can turn for emotional support and the wider perspective that springs from meeting people with diverse professional backgrounds. I thank my parents, Harriet and Svante, my sister Carola and her husband Xavier, and my grand-parents Ilse, Nan and Carl-Johan for their love and support. Finally, I want to express my immense gratitude to my partner Anders for his constant encouragement and so much more.

Thank you everyone!

Berkeley, March 2006

Mathias Creutz

Contents

Abbreviations

AI	Artificial Intelligence
AIC	Akaike's Information Criterion *or* An Information Criterion
ASR	Automatic Speech Recognition
BIC	Bayesian Information Criterion
CMU	Carnegie Mellon University
EM	Expectation Maximization
FINTWOL	Finnish Two-Level Morphology Analyzer
FLM	Factored Language Model
FSA	Finite-State Automaton
FST	Finite-State Transducer
HMM	Hidden Markov Model
IA	Item and Arrangement
ICA	Independent Component Analysis
ILP	Inductive Logic Programming
IP	Item and Process
LSA	Latent Semantic Analysis
MAP	Maximum a Posteriori
MDL	Minimum Description Length
ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
MML	Minimum Message Length
NLP	Natural Language Processing
OOV	Out of Vocabulary
PCA	Principal Component Analysis
SLT	Statistical Learning Theory
SOM	Self-Organizing Map
TWOL	Two-Level Morphology (Analyzer)
WP	Word and Paradigm

Some Mathematical Notations

$\arg\max_X f(X)$	value of X for which the function $f(X)$ reaches its maximal value
μ_i	i^{th} morph
C_i	category of the i^{th} morph
$P(X)$	probability of an event X
$P(X \mid Y)$	(conditional) probability of an event X, when the event Y is known to have occurred
$P(X, Y, Z)$	joint probability of X, Y and Z, i.e., the probability that X, Y and Z co-occur
$L(X)$	code length of X (in the MDL formalism)
$P(C_j \mid C_i)$	transition probability from category C_i to category C_j, i.e., the probability of observing C_j directly following C_i in a sequence
$P(\mu_i \mid C_i)$	morph emission probability, i.e., the probability that the i^{th} morph is μ_i, given that the i^{th} category is C_i
NON	non-morpheme (noise morph)
PRE	prefix
STM	stem
SUF	suffix

Chapter 1

Introduction

Language learning in humans takes place through active interaction with the world around us. Meaning for us is grounded in real-world experiences, involving all of our senses. Compare this to a computer: One can expose a computer to large amounts of language data, text or speech; however, the computer has very limited means for active interaction with the external world. What, then, can a computer learn or "understand" about natural language?

In automatic Natural Language Processing (NLP), language data are processed and manipulated by a computer program that does not really understand language. Still computer programs are able, among other things, to retrieve relevant documents from large databases (e.g., the World Wide Web), correct spelling mistakes and bad grammar in written texts, and convert speech to text (and vice versa) rather successfully. Automatic translation from one language to another has not yet been perfected, but works well when dealing with a limited domain, such as weather forecasts or product documentation.

Thus, existing software demonstrates that rudimentary language "skills" may suffice for a machine to be a helpful tool. How these skills become available to the computer is an important issue. A standard approach consists in having experts design the required linguistic resources, e.g., vocabularies and grammatical rules, which are used as the basis for the computer skills. Unfortunately such resources must be tailored separately for each language, which demands a large amount of manual work. Moreover, specific task domains require specialized vocabularies which must keep up with rapidly evolving terminologies, e.g., within news reporting and fields of science: biology, medicine, physics, etc.

An alternative approach to hand-creating linguistic resources is to design systems that mimic intelligent behavior by "learning" themselves and adapting to the language and task-specific data they encounter. Language is not random: there are regularities that can be captured mechanically, by exploiting distributional patterns found in the language data. For instance, in a particular language, certain sound sequences may occur frequently whereas others are rare. Linguistic structuralists have claimed that the

boundaries and identities of words and sentences can be established by statistical means directly from unsegmented streams of speech, with no knowledge of the meanings of the words (Harris, 1954). Saffran et al. (1996) show that humans are capable of discovering word units rapidly in a stream of a nonsense language without any connection to meaning. This suggests that also humans use distributional cues, such as transition probabilities between sounds, in language learning. Brent and Cartwright (1996) share this view and support their argument with computational simulation.

Based on a comprehensive review of contemporary studies of how children start to acquire language, Kit (2003) concludes that children certainly make use of statistical cues. As a probable underlying mechanism, Kit proposes the least-effort principle, which can be given a straightforward mathematical formulation. Mathematical models can be implemented on computers, making it possible to design systems that learn relevant linguistic structure from ordinary language data in an unsupervised manner, without the help of a "teacher" providing correct answers. The strength of such methods is that they can be applied irrespective of language or specialized vocabularies. These methods can serve to complement or even replace manual linguistic work.

1.1 Morfessor

This thesis focuses on the learning of a specific kind of linguistic knowledge, namely morphology, which concerns the regularities within word forming. According to standard linguistic theory, morphemes are the smallest individually meaningful elements in the utterances of a language. Every word consists of one or several morphemes into which the word can be segmented; consider for instance the morpheme segmentations of the following English words: "hand, hand+s, left+hand+ed, finger+s, un+avail+able".

Some natural language processing applications operate with words as the smallest linguistic elements. These systems will inevitably miss the close semantic relationships between words (e.g., "hand, hands, left-handed"), as well as grammatical categories (e.g., the relationship of the plural "s" in "hands" to other plural forms: "heads, arms, fingers"). For instance, in an information retrieval task, the consequence of such an approach is that only documents containing the exact search keyword will be retrieved. That is, a search for "hand" will not find documents containing only the plural form of that word: "hands".

In highly-inflecting and compounding languages (e.g., Finnish, Turkish, and German) the number of possible word forms is very high. This poses special challenges to NLP systems dealing with these languages. For example, in automatic speech recognition it is customary to use pre-made lists of attested word forms as a "normative" vocabulary. The incoming acoustic signal is matched against the list, and only words contained in the vocabulary can be recognized. Such a word list can be created by collecting word forms from large text corpora or existing lexicons, and the aim is to obtain as good coverage as possible of the words of the language. When processing languages with extremely rich word forming, the resulting word lists are typically very large, which is demanding from a computational point of view. A more serious problem is that many perfectly valid word forms are likely to be missing from the list anyway, since they might never have

väitös	kirja	n	teko	vaihee	ssa
thesis	book	of	act	phase	in

Figure 1.1. Morpheme segmentation of the Finnish word "väitöskirjantekovaiheessa" ("in the phase of doing a Ph.D.").

occurred in the corpus used as a source. For instance, in a particular Finnish 32 million word corpus, there are 4400 different word forms containing the element "puu" ("tree, wood"), but the forms "puusi, puukaan, pyökkipuu" ("your tree, [not] even wood, beech wood") are missing, among numerous other possible forms.

The vocabulary problem can be alleviated considerably by using morphemes instead of words as basic vocabulary units, and this idea serves as a starting point for the current work.[1] A method, called *Morfessor*, has been developed for the discovery of morphemes in an unsupervised manner from raw text data. This thesis describes the development of Morfessor together with experimental results. The following list constitutes a subjective assessment of the major contributions of the work:

- Morfessor learns a simple morphology in an unsupervised manner from unannotated text. No predefined grammatical rules of any specific language are required, which makes Morfessor a language-independent method.
- In contrast to other unsupervised morphology learning algorithms, Morfessor copes with morphologies in which words can consist of lengthy sequences of morphemes. This is computationally more demanding than the common approach of assuming that words consist of only two parts, typically one stem followed by one ending. Such a restrictive assumption is justifiable for some, but not all languages. As is demonstrated by an example in Figure 1.1, Finnish words can consist of multiple morphemes, where stems and endings occur in alternation.
- In contrast to existing unsupervised word segmentation methods, the later versions of Morfessor take sequential dependencies into account. (Word segmentation is necessary in many Asian languages, where there is no overt marking of word boundaries in text.)
- The mathematical formulation of Morfessor relies on Bayesian statistics and (in some publications) on the Minimum Description Length principle (Rissanen, 1989). Since these theoretical frameworks are robust and well understood, they provide elegant descriptions rather than *ad hoc* solutions with poor explanatory power.
- The role of "meaning" as well as "form" in the Morfessor model is touched upon. The notion of these central linguistic concepts is rudimentary in the current implementation, but extensions are possible. Learning from tokens versus types is also discussed, an issue that intrigues psycholinguists, who study language learning in

[1] A simpler alternative, applicable in some situations, is *stemming*. The purpose of stemming is to map related word forms onto the same word stem. For instance, the final "s" of "hands" would be dropped, and thus "hands" would be mapped onto "hand". Stemming is commonly used in information retrieval, where it is sufficient to determine whether some word form ("hands") is a variant of a relevant search term ("hand"). However, stemming is not sufficient, when complete word forms (rather than stems) need to be recognized or generated, e.g., in speech recognition or machine translation.

humans. Learning from tokens means that the frequency of elements (e.g., word forms) affect their processing and representation in the human brain. Learning from types suggests that the identification of distinct elements (e.g., a word form unlike previously encountered word forms) dominates over frequency effects.

- The morpheme segmentation produced by Morfessor has served as the language model utilized in a large-vocabulary Finnish speech recognizer. In comparison to the other studied models, the Morfessor-based model performs best.
- The morpheme segmentation produced by Morfessor has also been compared directly to a grammatical, linguistic morpheme segmentation. In the evaluation that has been carried out, Morfessor outperforms a widely known benchmark algorithm (Goldsmith, 2001, 2005). Due to the fact that the linguistic resources required for the evaluation did not yet exist, a segmentation reference, or *gold standard*, was constructed based on existing resources. Gold-standard morphological segmentations for a large collection of Finnish and English words were produced. The resulting software package, called *Hutmegs*, has been made publicly available for research purposes.

Outside the scope of the current thesis, Morfessor has been used successfully in the recognition of Turkish as well as Estonian speech (Hacioglu et al., 2003; Kurimo et al., 2006b). In the so-called Morpho Challenge competition, organized within the EU PASCAL Network of Excellence, Morfessor outperformed all participants in the Finnish and Turkish morpheme segmentation task (Kurimo et al., 2006a; Creutz, 2006). Hagen and Pellom (2005) apply Morfessor in English speech recognition intended for oral reading tracking within an interactive reading tutor program for children. Morfessor has also been used in Finnish information retrieval, both in the retrieval of text (Engström, 2005) and spoken documents (Kurimo and Turunen, 2005). Furthermore, in a number of works on language modeling, the segments discovered by Morfessor constitute the basic vocabulary (Siivola and Pellom, 2005; Broman and Kurimo, 2005; Virpioja, 2005). Klami (2005) has analyzed the word splits obtained when running Morfessor on stories told by Finnish children.

There is an online demonstration of Morfessor on the Internet: `http://www.cis.hut.fi/projects/morpho/`. Currently, the demo supports three languages: Finnish, English, and Swedish. Those interested in larger-scale experiments can download the Morfessor program and train models using their own data sets. Within a period of one year (May 2005 – April 2006) a monthly average of 17 downloads of the program has been registered.

1.2 Publications and Author's Contribution

This thesis consists of an introductory part as well as eight separate publications:

Publication 1. Mathias Creutz and Krista Lagus. Unsupervised Discovery of Morphemes. In *Proceedings of the 6th Meeting of the ACL Special Interest Group in Computational Phonology in cooperation with the ACL Special Interest Group*

in Natural Language Learning: Workshop on Morphological and Phonological Learning, held in conjunction with the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02), pages 21–30, Philadelphia, Pennsylvania, USA, July 2002.

Publication 2. Mathias Creutz. Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 280–287, Sapporo, Japan, July 2003.

Publication 3. Mathias Creutz and Krista 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: Workshop on Current Themes in Computational Phonology and Morphology, held in conjunction with the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 43–51, Barcelona, Spain, July 2004.

Publication 4. Mathias Creutz and Krista Lagus. Inducing the Morphological Lexicon of a Natural Language from Unannotated Text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR05)*, pages 106–113, Espoo, Finland, June 2005.

Publication 5. Mathias Creutz and Krista Lagus. Unsupervised Models for Morpheme Segmentation and Morphology Learning. Manuscript accepted for publication in the *ACM Transactions on Speech and Language Processing*, 2006.

Publication 6. Vesa Siivola, Teemu Hirsimäki, Mathias Creutz, and Mikko Kurimo. Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003)*, pages 2293–2296, Geneva, Switzerland, September 2003.

Publication 7. Teemu Hirsimäki, Mathias Creutz, Vesa Siivola, Mikko Kurimo, Sami Virpioja, and Janne Pylkkönen. Unlimited Vocabulary Speech Recognition with Morph Language Models Applied to Finnish. *Computer Speech and Language*, 2006 (in press).

Publication 8. Mathias Creutz and Krister Lindén. Morpheme Segmentation Gold Standards for Finnish and English. Report A77, Publications in Computer and Information Science, Helsinki University of Technology, October 2004.

Publications 1 – 4 present four consecutive development steps of the Morfessor model. Retroactively these model versions have been named *Morfessor Baseline*, *Morfessor Baseline-Freq-Length*, *Morfessor Categories-ML*, and *Morfessor Categories-MAP*, respectively. The model variants can be seen as instances of a general model, which is described in Publication 5. In Publications 1, 3, 4, and 5, the ideas that led to the described methods were developed jointly with D. Sc. Krista Lagus. The detailed development of the methods, their implementation and the experiments were carried out by the author. The mathematical formulation of the models, the planning and analysis

of the experiments, as well as the writing of the articles were joint work with primary contributions by the author.

Publications 6 and 7 concern the application of Morfessor in Finnish large-vocabulary speech recognition. The two publications describe two different experimental setups, in which different word fragments (segments of words) are used as basic vocabulary elements. The author described and applied the Morfessor model and participated actively in the interpretation of the experimental results. Especially in Publication 7 the author contributed to the design of the language models used, since these were based on previous work by the author (the so called statistical and grammatical morphs obtained from Morfessor and Hutmegs, respectively). The solution to the out-of-vocabulary problem is the result of joint work.

Publication 8 describes the design of a linguistic reference segmentation of a large collection of Finnish and English word forms. This reference or gold standard, called Hutmegs, has been utilized for evaluation purposes in Publications 3, 4, 5, and 7. Publication 8 is furthermore intended as a user's guide for researchers interested in adopting Hutmegs. Prior to the design of Hutmegs, the author produced evaluation material for Finnish and English. Building on the experiences acquired, the better linguistically motivated Hutmegs gold standard was created jointly with Krister Lindén, based on discussions on central issues: models of morphology, specific details concerning Finnish and English, evaluation strategies, and the notation used. Krister Lindén produced the English gold standard, while the author produced the Finnish gold standard and the evaluation programs, as well as did all the writing work.

1.3 Structure of the Thesis

The introductory part of the thesis is meant to be a coherent presentation that can be read without much consultation to the eight attached publications. However, there are some issues that are discussed only in the publications (e.g., details on search algorithms and performed experiments).

The introductory part comprises five chapters. The current chapter (Introduction) is followed by Chapter 2, which presents some central linguistic theory as well as the basics of machine learning and automatic speech recognition. Chapter 3 then describes the development of the Morfessor model and the mathematical formulation of the model. Chapter 3 additionally gives a general account of search algorithms and related work. Chapter 4 discusses the "direct" and "indirect" evaluation of morphology-learning algorithms. A direct evaluation of the placement of morpheme boundaries is carried out using Hutmegs as a gold standard. An indirect evaluation is performed by utilizing Morfessor in an unlimited-vocabulary Finnish speech recognizer. Chapter 5 concludes the introductory part of the thesis.

Chapter 2

Building Blocks

This chapter is an introduction to the general foundations of the current work. The work rests upon several "building blocks", which represent central findings within the fields of linguistics, computational modeling, and machine learning. The chapter also covers some basics of automatic speech recognition, which exemplifies a natural language application that can benefit from the type of model developed in this work.

2.1 Linguistic Theory

Linguistic theory is a vast field with many neighboring sciences. The current presentation will address *some* fundamental themes in general linguistics: meaning and form, lexicon and grammar, and morphology. In addition, some mathematically tractable models are discussed.

2.1.1 Meaning and Form

The work of Ferdinand de Saussure (circa 1900) is considered as the foundations of linguistic *structuralism*. Saussure argued that the object of linguistics must be language (in French: "langue") as opposed to speech ("parole"). Language, for Saussure, is the symbolic system through which we communicate, whereas speech refers to actual utterances. Since we can communicate an infinite number of utterances, it is the system behind them that is important. This system is made up of *signs*, where each sign is defined precisely by how it differs from the other signs in the system (e.g., "tree" vs. "bush", "branch", or "hierarchy"). Languages are able to carry meaning because they are organized at every level by two sets of rules, *syntagmatic* and *paradigmatic*. Syntagmatic rules govern how signs can be combined into sequences, whereas paradigmatic rules state which subset of all signs can be put in a specific "slot" in the syntagm (see Fig. 2.1).

Signs are composed of two parts: a signifier ("signifiant") and a signified ("signifié").

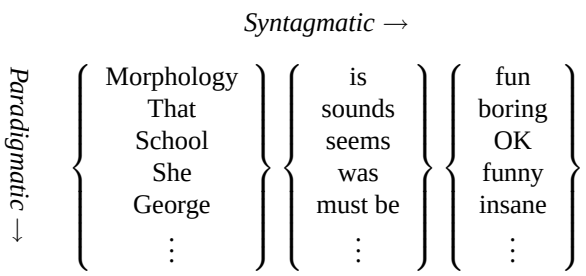


Figure 2.1: Illustration of syntagmatic and paradigmatic relations. The syntagm concerns the combination of several signifiers (here words) into a sequence, where the position is significant for the meaning of the phrase. The paradigm concerns the selection of a signifier to occupy a position in the syntagm.

The signifier is an *acoustic image* (e.g., the sound sequence [tri:]), which is used to refer to a *concept* (e.g., the idea of a tree). Saussure emphasizes that words (signifiers) are not labels for real things that exist in the world. Words rather refer to ideas we have about the world. For instance, the word "tree" refers to a concept we have in our heads, and linking this concept to a particular tree in the real world involves particular kinds of language work. (de Saussure, 1916; Peterson, 2001)

Arbitrariness of Form

Saussure claimed that the bond between the signifier (form) and signified (meaning) is arbitrary. This *arbitrariness of form* can be illustrated by comparing the words for "tree" in a few languages: "arbre", "Baum", "derevo", "puu", "träd". The choice of signifier is based on social convention rather than on any natural or essential link. The above words are equally well suited for referring to roughly the same concept because there is nothing about any of these sounds that is more treelike than any other. (de Saussure, 1916; Peterson, 2001)

Structuralist Approach to Meaning

Saussure was followed by several prominent structuralists, among others Leonard Bloomfield, Edward Sapir, and Zellig S. Harris. A central question was the role of *meaning* within linguistic science. The structuralists argued that meanings in language are closely related to the forms in language. Real-world experiences, such as pointing to the referent of a word, or understanding the meaning of a word from the situation in which it is said, are adequate and essential, but not sufficient, conditions. The less obvious meanings of many words, such as "time", "consider", "the", "of", can be learned only by much experience with the *neighboring words and sentences* with which they occur (Harris, 1991). In Harris' words: "we find that the meanings are not additional

properties unrelated to the syntactic forms, but are close concomitant of the constraints on word-choice in the operator-argument relation and on the participation of words in various reductions and constructions." That is, to understand a particular word, one has to know in which possible combinations with other words it occurs.

Interestingly, Harris also claims that the boundaries and identities of words and sentences can be established by *statistical means*, with *no knowledge of the meanings* (Harris, 1954, 1991). This opens interesting perspectives, which are explored in the current work.

Reflections

According to Saussure's claim of arbitrariness of form, the meaning of a word does not depend on the individual sounds it consists of. However, there is evidence that some sounds and sound clusters can evoke certain types of meanings: "[S]ome direct 'sound symbolism' has been seen between certain sound types and visual or tactile shapes. Most people agree that the made-up word 'oomboolu' would better designate a round, bulbous object than a spiky one. In addition, the appropriateness of the vowel sound represented by *ee* in English 'wee' and *i* in French *petit* 'small' and Italian *piccolo* 'small' for expressing things of small size has been traced in several languages." ("Language", 2005)

The structuralist approach to meaning is related to the *meaning-is-use* theory by Wittgenstein (1953). However, linguistic structuralists only study meaning *within* language and leave other types of meaning outside their scope of investigation. This view has been criticized by *cognitive linguists*, e.g., George Lakoff, which see linguistic abilities as but one part of general cognition. Cognition is embodied and involves multiple modalities: auditory (hearing), visual (sight), tactile (touching), kinesthetic (movement), etc. Cognitivists claim that the difference between language and other mental processes is possibly one of degree but not one of kind. Thus, it makes sense to look for general principles that are shared across a range of mental domains. Meaning is reflected in the *mental categories* (i.e., concepts or ideas) which people have formed from growing up and acting in the world. (Saeed, 1997)

A third, and classical, meaning theory is that of *denotational semantics*. This theory attempts to determine the truth conditions of uttered sentences. A sentence refers to a situation, and a listener who understands the sentence knows what conditions in the world would make the sentence true. An objection to this theory is that we have no access to a reality independent of human categorization, and therefore linguistic symbols cannot refer directly to states of affair in the world. (Saeed, 1997)

2.1.2 Lexicon and Grammar

Language is an open system where entirely new utterances can be produced and understood. This system relies on *productive* mechanisms by which a virtually infinite number of meanings can be conveyed by arranging a limited set of elements according to a limited set of rules. Commonly, the set of elements is called the *lexicon* and the set

of rules is called the *grammar*.[1]

In accordance with the principle known as Occam's razor, a scientific theory should be as simple as possible while still providing an adequate description of the phenomenon it tries to explain; see, e.g., Gibbs (1996). Such a strive for simplicity, or minimalism, has guided the structuralists as well as representatives of later linguistic movements. Both the structuralist Leonard Bloomfield and the generativist Noam Chomsky prefer models where all regularities found in language are captured by the grammar, whereas the role of the lexicon is to list the remaining facts, or "idiosyncrasies", that cannot be covered by any rules. (Bloomfield, 1933; Chomsky, 1965; de Beaugrande, 1991)

Morphemes

The lexicon, according to Bloomfield, consists of the set of minimal Saussurean signs, called *morphemes*. Further definitions for morphemes include the following:

- the smallest individually meaningful elements in the utterances of a language (Hockett, 1958),
- minimal meaningful form-units (de Beaugrande, 2004),
- the primitive units of syntax, the smallest units that can bear meaning (Matthews, 1991),
- linguistic forms which bear no partial phonetic-semantic resemblance to any other forms (Bloomfield, 1933).

Morphemes are thus portions of utterances that recur in other utterances with approximately the same meaning. They are minimal in the sense that they cannot be broken into independently recurring smaller pieces in such a way that the meaning of the whole form is related to the meanings of the smaller pieces (Hockett, 1958). For instance, if this definition is applied on the sentence "Sun springs us into joyfulness." the following morpheme sequence is obtained: "SUN SPRING -S US IN- TO JOY -FUL -NESS".

In some cases the morphemic status of a sound sequence is borderline. For example, the sequence "sl-" appears in the English words "slick, slip, slither, slide, slimy, slink, sling, slog, slosh, slouch, slow" etc., words which seem to have in common some notion of a smooth trajectory or substance, possibly with some obstacle involved. However, "sl-" occurs in combination with neighboring sound sequences (e.g., "-ick, -ip, -og") that do not themselves combine in a sufficiently regular way, as morphemes; consider, e.g., "sl-og" vs. "d-og, f-og, fr-og, j-og, l-og": what would be the common meaning of "-og" in these words? (Harris, 1991)

[1] Many linguists are likely to consider both the lexicon and rules part of the grammar, but in this work the more informal use of terms will be practiced.

Phonemes

The minimal meaning*ful* units, morphemes, consist of *phonemes*, which are the minimal meaning-*distinguishing* units in language. Phonemes are sound units and their organization is studied within the field of *phonology*. Each language has a particular set of phonemes. For instance, in English, the sounds [p] and [b] are phonemes, because this sound opposition is sufficient for distinguishing between morphemes, as in "pin" vs. "bin". By contrast, the aspirated "p" [p^h] occurring in "pin" [p^hɪn] is not an instance of another phoneme than the unaspirated "p" occurring in "spin" [spɪn]. If one pronounces these words with the wrong quality of the "p" sound, they will still be intelligible and mean the same thing (even though the speaker may reveal that he is not a native speaker of the English language). (Bloomfield, 1933)

Morphology and Syntax

Grammar is traditionally split into two subsystems: *morphology* and *syntax*. Morphology studies how words are formed from morphemes, whereas syntax studies how utterances are formed from words. This split is based on the view that words are important units and that the processes going on within words are different from those going on between words. In any case, it is difficult to draw a clear boundary between morphology and syntax. There have been attempts to construct a grammar based directly on morphemes without a division into morphology and syntax. However, current linguistic theories do generally maintain the division. Derivation and compounding are considered purely morphological phenomena, because their effects are limited to within one word. By contrast, inflection can be considered part of both morphology and syntax, since inflection pertains to the role of the word in the sentence. (Anderson, 1992) (Inflection, derivation, and compounding are described in Section 2.1.3.)

Reflections

The dualism between lexicon and grammar has been questioned, e.g., by modern connectionists, who draw inspiration from findings in psycholinguistic research. The connectionists argue that learning, representation, and processing of grammatical rules as well as lexical items takes place over a large number of interconnected simple processing units in the human brain. There are no mental rules and no distinct system to process rules. Rather both the lexical and grammatical knowledge is processed in the same areas of associative memory, and thus there is no clear division between lexicon and grammar (see, e.g., Elman et al., 1996).

However, the distinction between lexicon and grammar also gets some support, likewise on psycholinguistic grounds. Ullman (2001) claims that the memorization of arbitrary form-meaning pairings depends upon an associative or "declarative" memory (lexicon), whereas a "procedural" system is specialized for computing sequences (grammar). According to Ullman, these systems are indeed distinct. The entire mental lexicon as well as nonlinguistic knowledge about facts and events are stored in the associative memory. The procedural circuits are implicated in the learning and expression of motor and

cognitive “skills” and “habits”, from simple motor acts to skilled game playing.

The strive for optimality and minimalism in the description of language does not lead to one well-defined optimal model. This is due to the fact that the cost of different linguistic subsystems is difficult to assess and may be different in different situations. Suppose that associative memory is cheap, whereas the application of a multitude of grammatical rules is slow, i.e., computationally expensive. Moreover, some regularities that could be captured by rules have low learnability in practice, that is, most speakers of the language never grasp the particular underlying mechanism. In this scenario, it would be advantageous to store a large number of entire phrases in an enormous lexicon and reduce the number of rules to a minimum, as opposed to the Bloomfieldian view of the lexicon as a storage only for the arbitrary knowledge that cannot be captured by rules.

Naturally, a linguistic theory can be elegant in its own right without having to rely on how language is processed in the human brain. Nevertheless, the way morphemes are combined is very often restricted by specific constraints that are hard to capture by general rules. For instance, it is customary to say “*on* purpose” and “*by* accident” rather than “*by* purpose” or “*on* accident”. It is not impossible that such conventions may be explained by general underlying regularities, but probably they are best modeled as arbitrary lexical facts.

Representatives of cognitive linguistics assume that there is a continuum between syntax (grammatical rules) and lexicon. Cognitive linguists operate with *constructions* rather than morphemes. Constructions are defined as symbolic units consisting of a form and a meaning. This definition resembles that of the morpheme, but the minimal status of the construction is not stressed. Constructions can be fixed idiom-like expressions that always appear exactly in the same form, e.g., “*no can do*”. However, typically they allow for modifications, such as inflection of some of their elements, e.g., “*draw* a *conclusion*” vs. “*draw*ing *conclusion*s” (see, e.g., Goldberg, 2003; Croft and Cruse, 2004).

In addition to the specific constraints and conventions that apply to the combination of morphemes, frequently co-occurring morphemes tend to acquire nuances of meaning as a whole that are not deducible from the meanings of the individual morphemes. For instance, what is the morphemic status of the English word “joystick”? On the one hand, “joystick” can be defined as a morpheme, because its meaning is not a transparent composition of the meanings of the morphemes “joy” and “stick”. On the other hand, these constituents do contribute to the meaning of the whole; at least the “stick” aspect is part of the “joystick” concept. de Marcken (1996) proposes a model for unsupervised language acquisition, which involves two central concepts: *composition* and *perturbation*. Composition means that an entry in the lexicon is composed of other entries. Perturbation means that changes are introduced that give the whole a unique identity. This framework is similar to the class hierarchy of many programming languages, where classes can modify default behaviors that are inherited from superclasses. The more of its properties a lexical parameter inherits from its components, the fewer need to be specified via perturbations.

It is worth mentioning that not all scholars accept the morpheme as a unit in the first place. A morphology based on whole words and word stems is advocated by Anderson (1992). He argues that if the morpheme is a basic meaning-carrying unit, there

should predominantly exist *one-to-one* relations between categories of meaning and aspects of form. However, Anderson claims that this relation is very often *many-to-many* rather than one-to-one. For example, the Icelandic word "hafðir" (you had) is constituted by the following links between meaning and form: the morpheme HAVE → "haf", indicative mood → "a", past tense (preterite) → "a-ði", singular number → "ir", second person → "r". Anderson prefers word formation rules that take stems as input, apply operations on them, and produce word forms as output. For example, the regular plural formation of English nouns can be expressed as /X/ → /Xs/ ("dog" → "dogs"). Within the general typology of morphology, this is an Item and Process model of word structure (see Section 2.1.3, page 16).

2.1.3 On Morphology

Due to the scope of this thesis, morphology is treated more thoroughly than other subfields of general linguistics. In the following, some basic morphology-related concepts will be introduced. The classical division of morphology models into three types, as suggested by Hockett (1954), will be covered, followed by a discussion of the applicability of these models to different types of languages.

Morphs and Allomorphs

The sentence "Sun springs us into joyfulness." was above segmented into the morpheme sequence "SUN SPRING -S US IN- TO JOY -FUL -NESS". According to the classical convention of term usage, "morpheme" is reserved for an abstract concept, whereas actual segments of utterances are called *morphs*. This distinction can be illustrated by representing the corresponding morph sequence as "sun+spring+s+us+in+to+joy+ful+ness". Morphemes are abstract classes, realized as sets of *allomorphs*. Allomorphs are morphs that mean the same thing and occur in *complementary distribution*. If the word "sun" in the example is replaced by "sunny weather", the following morph segmentation is obtained: "sunn+y+weather+spring+s+us+in+to+joy+ful+ness". This demonstrates that the morpheme SUN can be realized as the two morphs "sun" and "sunn", which are allomorphs. The longer variant "sunn" is used before the morpheme "-Y", whereas the shorter variant "sun" is used in other contexts.

An alternative term usage convention is to refrain from using the term "morph" and talk about morphemes both in the abstract and concrete sense. When necessary to emphasize the "morph" aspect, other expressions, such as *morphemic segments* or *word-parts*, are used. (Harris, 1951; de Beaugrande, 2004)

It should be noted that morphologies can be formulated both for speech (pronounced form) and text (orthographic form). For languages, such as English, where the connection between orthography and pronunciation is sometimes intricate, the two morphologies may exhibit allomorphy at different points. For instance, in our previous orthographic example two allomorphs were found for the morpheme SUN, namely "sun" and "sunn". However, the two spelling variants are pronounced identically and there is thus only one allomorph from the point of view of pronunciation. The opposite is true for

the past tense ending "-ed", which can be pronounced in three ways: as [d] in "sailed", as [ɪd] in "waited", and as [t] in "kicked".

Stems and Affixes

Mainly three processes are involved in the creation of compositional word forms, i.e., words consisting of multiple morphemes: *inflection*, *derivation*, and *compounding*. Different *inflectional* forms of a word express different grammatical relations of the word with other words in the sentence. For instance, English nouns occur in both singular and plural number, e.g., "a *child*" vs. "all the *children*". In *derivation*, the semantics or part-of-speech of the original word changes; e.g., "*childish*", "*childly*", and "*childhood*" are derived from "child".

Inflection and derivation typically take place through *affixation*. *Affixes* are morphemes that never occur as free forms, but need a *stem* to attach to. Depending on the position of the affix in relation to the stem, the affix is classified as a *prefix*, *suffix*, *circumfix*, or *infix*. A prefix precedes the stem and a suffix follows it (e.g., English "un-" and "-ness" in "unhappiness"). A circumfix consists of a pre- and suffix that co-occur systematically, e.g., German "ge- -en" in "gesprochen" (Eng. "spoken"). Infixes are rare in European languages, but one example is the colloquial "bloody infix" in British English, e.g., "abso-bloody-lutely, Coca-bloody-Cola, fan-bloody-tastic".

A stem can be compositional and consist of layered substems and affixes. For instance, in "childishness", the suffix "-ness" is attached to the stem "childish", and the suffix "-ish" is attached to the stem "child". A minimal, indivisible, stem is called a *root* (e.g., "child" in "childishness"). Words formed by *compounding* contain multiple roots (e.g., "childcare, childbirth, childbedfever, flowerchild"). (Matthews, 1991; Karlsson, 1998)

Hockett's Models of Morphology

Hockett (1954) has identified and named three general approaches to the modeling of morphology: *Word and Paradigm* (WP), *Item and Arrangement* (IA), and *Item and Process* (IP).

Word and Paradigm is the classical school-book approach of grouping words with the same inflectional pattern into inflectional classes, or paradigms. Each slot in the paradigm corresponds to some grammatical features assigned to the word (e.g., present tense, 3rd person). For instance, the table below contains the inflections of five English verbs that exhibit some differences in their forms and that have been grouped into five separate paradigms:

	Paradigms				
Grammatical form	I	II	III	IV	V
Infinitive	wait	invite	split	sell	take
Present tense, 3rd person	waits	invites	splits	sells	takes
Present participle	waiting	inviting	splitting	selling	taking
Past tense	waited	invited	split	sold	took
Past participle	waited	invited	split	sold	taken

When the inflectional class of a new word is known, all its inflections can be determined by analogy from other words belonging to the same class. For example, if the following class memberships are assumed: "shout(I), like(II), cut(III), tell(IV), shake(V)", the following inflected forms can be deduced, among others: "shouts, liked, cutting, told, shaken".

In the Word and Paradigm model, morphemes are unnecessary concepts, but they are central in the **Item and Arrangement** model. In IA, word forms are composed of morphemes (items), which occur in certain arrangements. The morpheme representations of word forms can be sorted into paradigms just as in the WP model, e.g.:

I	II	III	IV	V
WAIT	INVITE	SPLIT	SELL	TAKE
WAIT + -S	INVITE + -S	SPLIT + -S	SELL + -S	TAKE + -S
WAIT + -ING	INVITE + -ING	SPLIT + -ING	SELL + -ING	TAKE + -ING
WAIT + -ED	INVITE + -ED	SPLIT + -ED	SELL + -ED	TAKE + -ED
WAIT + -EN	INVITE + -EN	SPLIT + -EN	SELL + -EN	TAKE + -EN

The inflection patterns now look identical for each paradigm; the infinitive consists only of the verb stem whereas the other forms are obtained by adding the endings -S, -ING, -ED, and -EN, respectively. How the morpheme sequences are realized as word forms is governed by *allomorphy*:

In the Item and Arrangement model, word segments which are allomorphs of the same morpheme are identified and it is necessary to determine which allomorph to use in which context. The following allomorphs are obtained for the morphemes in our example: WAIT = {wait}, INVITE = {invite, invit}, SPLIT = {split, splitt}, SELL = {sell, sol}, TAKE = {take, tak, took}, -S = {s}, -ING = {ing}, -ED = {ed, d, ∅}, and -EN = {ed, d, ∅, en}.[2] Now, the production of word forms is a matter of selecting the correct allomorph according to context. For instance, INVITE + -ING → invit + ing = inviting, SPLIT + -EN → split + ∅ = split, SELL + -EN → sol + d = sold, and TAKE + -ED → took + ∅ = took.

The existence of the *zero morph* (∅) is somewhat controversial. In the inflection of the verb "split", the past tense and the past participle are identical to the verb stem, which is "split". Therefore, it is customary to think that the ending morphemes (-ED and -EN) are not realized at all in these forms, i.e., they are realized as zero morphs. A more problematic analysis is provided for the past tense form of the verb "take". The morpheme -ED is used as a past tense marker, even though it is realized as a zero

[2] This list of allomorphs is not exhaustive with respect to the complete morphology of the English language.

morph, whereas the fact that the stem undergoes a vowel change ("take" → "took") is merely seen as an instance of stem allomorphy. A more appealing alternative solution for "took" is to have a discontinuous allomorph "t-k" of TAKE, and an infixed allomorph "-oo-" of -ED.

The **Item and Process** model provides a more elegant framework for treating morphological processes other than concatenation. The items of IP consist of morphemes and the processes correspond to morpho-phonological rules. (The morphemes are alternatively called roots in order to contrast IP with IA.) Word forms result from applying rules to one or several morphemes, which may alter the realization of the morphemes. For instance, the present participle of English verbs is produced by appending "-ing" to the verb stem, dropping the final "e" of the stem, and sometimes doubling the final consonant, if present: "wait+ing = waiting, invite+ing = inviting, split+ing = splitting". The past tense forms are usually produced by appending the ending "-ed" ("wait" → "waited"), but there are also cases of vowel change ("take" → "took").

Note that in IP not every phoneme (or letter) of a word necessarily belongs to a morpheme. For instance, in the word form "waited" the suffix "-ed" is a *marker* of the past tense formation process. In the form "took" the past-tense marker consists of the replacement of "-a-e" with "-oo-". In principle, one could use the terminology morpheme/allomorph for the processes and their markers insofar as these carry meaning. However, by tradition, morphemes are considered to be items (i.e., segments of words), never processes.

Morphology Typologies and Applicability of Hockett's Models

In classical morphological typology, dating back to work by Edward Sapir (1921), the world's languages are characterized by their position on two continua: *isolating* vs. *synthetic*, and *agglutinative* vs. *fusional*.[3]

The opposition between isolating and synthetic languages lies in the number of morphemes words typically consist of. The higher degree of synthesis, the higher the morpheme-per-word ratio is and the larger is the set of possible distinct word forms. In a strictly isolating language (e.g., Chinese) each word consists of one single morpheme. English is among the most isolating of the languages of the Indo-European family. Examples of moderately synthetic languages are German and Japanese. Finnish and Turkish are very synthetic and Inuit is polysynthetic (i.e., very highly synthetic).

The other continuum, agglutination vs. fusion, is concerned with how morphemes are put together in order to form words. Agglutination means that words consist of morphemes "glued" (or concatenated) together with a minimum of mix-up. In the resulting word form, the morphs are easy to distinguish and each morph typically corresponds to one unit of meaning, as for instance in the English word "open+heart+ed+ness" and the Finnish word "syö+tä+v+i+ä" ("[some] edible [thing]s"; literally: "eat + PASSIVE + PARTICIPLE + PLURAL + PARTITIVE"). Fusion, in contrast, implies that a combination of several morphemes is manifested in one single morph for which it is difficult to find a segmentation into smaller parts. Examples of fusion are the English verb "could"

[3] Isolating languages are alternatively called *analytic* languages.

(can + PAST TENSE) and the French article “au” (from “à + le” meaning “to the”),

Keeping these morphology typologies in mind, the three morphology models identified by Hockett can be ranked in order of increasing generality, as follows: Word and Paradigm, Item and Arrangement and Item and Process.

The Word and Paradigm model is only applicable to languages with a fairly high degree of isolation, because the memorization of paradigms containing hundreds or thousands of different forms, as would be the case for highly synthetic languages, is hardly fruitful. However, if fusion is common in a language, the WP model may be a reasonable choice. For instance, in many European languages, there is a very limited set of different endings, and each ending is typically “reused” in several inflectional forms. One single ending can code for case, number as well as gender, as for example the German suffix “-er”, which can stand for the nominative singular masculine (“ein nett*er* Mann”; “a nice man”), the genitive plural feminine (“nett*er* Frauen”; “of nice women”) and so on. In this case it makes no sense to segment out distinguishing markers for plural, genitive, feminine etc., as opposed to analyzing agglutinative word forming, as in Finnish “mukav+i+en nais+t+en” (“nice + PLURAL + GENITIVE woman + PLURAL + GENITIVE”). The German example illustrates how it sometimes comes more naturally to think of inflection as a mechanism whereby a word takes different shapes (e.g., “nett, netter, nette, nettes, nettem, netten”) than to think of a base form onto which additional meanings can be attached (e.g., Finnish “mukava → mukava+n → mukav+i+en”).

The IA and IP models can nevertheless be applied to the cases where WP is used. For instance, German adjectives can be thought to consist of a stem and an ending (“nett+er”), where the ending is considered as a morph manifesting three morphemes (e.g., “GENITIVE + PLURAL + FEMININE”). Alternatively, one can maintain that case, number, and gender systematically co-occur in German endings, whereby “-er” would manifest one such “heavy” morpheme. Note also that IA or IP are not only an option, but a requirement, if one wants to describe adequately the whole morphology of the German language. The productive compounding cannot be satisfactorily described by a WP model; consider word forms such as “Fuss·ball·spieler·daten·bank” (“database of football players”) and “Kunst·stoff·spritz·maschinen” (“plastics spraying machines”). When it further comes to nonconcatenative phenomena (e.g., Umlaut in “Fuss” vs. “Füsse”; “foot” vs. “feet”), the IP modeling framework is to be preferred over IA.

2.1.4 Mathematically Tractable Linguistic Models

The analytic tools provided by the structuralists have been used by linguists for studying and describing the phonological and morphological systems of numerous languages all over the world. Zellig Harris took a further step by proposing a fully *automatic* procedure for discovering morphemes from words and sentences. Noam Chomsky suggested a model of syntax that explicitly identifies all “well-formed”, or grammatical, sentences of a language together with their phrase structure. The two-level morphology formalism by Kimmo Koskenniemi provides an efficient means for the automated morphological analysis and generation of word forms in languages with complex morphology. In the following, each of these approaches will be discussed in some detail.

Harris' Morpheme Segmentation Method

Harris (1955, 1967, 1991) suggests the following method for automatic morpheme segmentation: "given the first m phonemes of a given n-phoneme sentence, for every m, $1 \leq m \leq n$, we count how many different phonemes follow these first m phonemes in all sentences which begin with these m phonemes. The same procedure can be used to count the predecessors of the last m phonemes of the sentence, for each m. The points in the given sentence at which the number of successors (or predecessors) forms a peak are, to a first approximation, the boundaries between the morphemic segments of the given sentence".

For instance, the following segmentations are obtained for some English words: "dis+turb+ance, dis+em+body, di+sulf+ide, de+form+ity, apple". (In this example, spelled words are used instead of phonemic sequences.) The segmentation of "deformity" is based on the following statistics: In the English corpus used, in all words beginning with "d" there were 15 different second letters. In all words beginning with "de" there were 26 different third letters. In all words beginning with "def" there were 9 different fourth letters. That is, there is a peak in the successor count between "e" and "f", and thus a morpheme boundary is suggested at that location. The subsequent successor counts form a series of decreasing numbers: 5, 4, 4, 3, 1 (no more peaks). When calculating the predecessor counts from the end of the word the following sequence is obtained: 25, 17, 19, 9, 2, 2, 4, 1. That is, there are two peaks (19 and 4): before the third to last letter ("-ity") and after the second letter ("de-").

Harris' method has been widely used in later research. Hafer and Weiss (1974) were able to perform more extensive computer experiments than Harris (1967). They also explored a larger number of segmentation policies; in addition to successor and predecessor *count*, they calculated successor and predecessor *entropy* to measure the predictability of the continuation of the word. Many modern word and morpheme segmentation algorithms rely entirely on Harris' method or use it for bootstrapping.

Generative Grammar

Harris' student Noam Chomsky criticized traditional grammars for being incomplete and "relying on the intelligence of the understanding reader". All languages provide means for expressing indefinitely many thoughts, and according to Chomsky (1965), the creative aspect of language use and related universal deep-seated regularities can be captured by a *universal grammar*. Chomsky wanted to supplement the grammars of particular languages with the universal grammar, since "a fully adequate grammar must assign to each of an infinite range of sentences a structural description indicating how this sentence is understood by the ideal speaker-hearer" (a person having full competence of the language). A mechanism that provides such an explicit analysis is called a *generative grammar*.

The generative grammar comprises transformation rules that specify how any well-formed string of a language can be generated from a *deep-level* "universal" representation. For instance, the two English *surface-level* sentences "The dog chases the cat." and "The cat is chased by the dog." would have a common deep-level analysis. Chom-

sky's work has been tremendously influential in general linguistics, despite the fact that many scholars do not share his view of an innate universal grammar; for a discussion, see, e.g., Elman et al. (1996).

Two-Level Morphology

In the 1960's and 70's, Chomskyan generative grammars were devised for describing phonology and morphology. The implementation involved ordered sequences of context-sensitive rewrite rules that converted abstract representations into surface forms through a series of intermediate representations. For instance, the Finnish words "kädessä, käsissä" ("in [the] hand, in [the] hands") could be generated from the deep-level morpheme representations "käte+ssa, käte+i+ssa" through the following steps: (1) The stem-final "e" is dropped before the plural marker "i": "käte+ssa, kät+i+ssa". (2) A "t" followed by "i" is changed to an "s": "käte+ssa, käs+i+ssa". (3) A "t" in a closed syllable ("tes") is softened to a "d": "käde+ssa, käs+i+ssa". (4) Due to vowel harmony, the back vowel in "-ssa" is changed to a front vowel: "-ssä": "käde+ssä, käs+i+ssä".

A notable problem with ordered rewrite rules is that they are indeed sensitive to the ordering. If rules (2) and (3) were to change place in the above example, one of the resulting word forms would be different: "kädessä, kä**d**issä". Another problem consists in the ambiguity of the underlying deep-level form. For example, if one observes the surface-level form "käsissä", how does one know that the deep-level form is "käte+i+ssa" rather than, e.g., "käsi+ssa"?

Koskenniemi (1983) solved the ordering and ambiguity problems by introducing a non-generative model which, however, retained the two separate levels: deep, or *lexical*, level vs. surface level. Hence, the model was called *two-level morphology* (TWOL). As opposed to the sequential rule processing in generative models, the rules work in *parallel* in the two-level formalism. The rules have simultaneous access to both the lexical and surface-level representation, and the character sets may be different on the different levels. For instance, the lexical representations of the words "kädessä, käsissä" may be "käTE+ssA, käTE+I+ssA". The rule "`T:s <=> _ E:i`" denotes that a "T" on the lexical level should be realized as an "s", when immediately followed by a lexical "E" realized on the surface as "i".

Each rule of a two-level morphology model can be implemented as a separate and fairly simple *finite state transducer* (FST). The same applies to the lexicon, which contains the deep-level representations of words and morphemes. The main lexicon can be split into logical sublexicons, each implemented as a finite state automaton: One sublexicon can contain the noun stems, another the verb stems, a third one the inflectional endings of nouns, and so on. All these components of the full morphological model can be merged into one huge FST by means of finite-state algebra, for which efficient tools exist; see, e.g., Karttunen and Beesley (2005); Mohri and Riley (2002).

The morphologies of numerous languages have been described in the two-level morphology formalism.[4] Koskenniemi has formulated an explicit and principled method,

[4] An early list of languages comprises Finnish, English, Japanese, Romanian, French, Swedish, Old Church Slavonic, Greek, Lappish, Arabic, and Icelandic (Koskenniemi, 1984). A search on the Internet additionally

addressed to linguists, for the discovery of appropriate two-level rules (Koskenniemi, 1991). Theron and Cloete (1997) has proposed an automatic approach, suitable for machine learning. Their system takes word pairs as input (inflected form vs. base form) and attempts to find an optimal transformation of the base form into the inflected form, by adding a prefix and a suffix and by applying sound changes, if necessary.

2.2 Machine Learning in Natural Language Processing

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics. It studies how computers can process and manipulate natural language and how computers eventually could "understand" human language, or at least, interact with humans in a seemingly intelligent manner using language. NLP applications include, but are not limited to *speech synthesis* (text-to-speech conversion), *speech recognition* (speech-to-text conversion), *machine translation*, *information retrieval*, *question answering*, and *text proofing* (checking and correction of misspellings and incorrect grammar).

NLP applications typically rely on large databases of linguistic knowledge, e.g., the vocabulary of a language (including inflections), thesauri, translation equivalents in pairs of languages, and pronunciation dictionaries. The manual design of such resources is labor-intensive and requires considerable effort by linguistic experts. Once the databases have been created, they need to be maintained on a continuous basis, since language evolves and new terminology emerges in a number of specialized domains. To reduce the amount of tedious manual work, *machine learning* can be utilized as a replacing or complementary technique.

Machine learning is the capacity of a computer to learn from experience (i.e., data) and to extract knowledge from examples. A successful learner should be able to make general conclusions about the data it is trained on. This allows it to act appropriately in new situations. For instance, an NLP application may learn that there are different categories of words: nouns, verbs, adjectives etc., and subcategories of these. When confronted with an unknown word form in context: "Did you see where I put my red *gambunk*?", the program should ideally be able to classify this word ("gambunk") as a noun designating some kind of an object. This entails, among other things, that this word can probably be inflected into plural number ("gambunks").

The following presentation of machine learning focuses on *statistical learning* in an *unsupervised setting*. Other learning methodologies are also touched upon. Unless explicit reference to specific scientific publications is made, a number of prominent text books have been used as a source (Gelman et al., 1995; Haykin, 1999; Manning and Schütze, 1999; MacKay, 2003; Nilsson, 1996; Russell and Norvig, 1995) together with Wikipedia[5].

finds references to works on Amharic, Croatian, Danish, German, Norwegian, Russian, Turkish, etc.

[5] `http://en.wikipedia.org/wiki/Main_Page`

2.2.1 Types of Learning

There exist three major types of machine learning: *supervised*, *unsupervised*, and *reinforcement learning*.

In supervised learning, there is a "teacher" that provides the learner with a set of input-output pairs. For instance, Erjavec and Džeroski (2004) present a system that is trained on pairs of Slovene inflected word forms and the morphosyntactic analyses of these words. The system is then capable of deducing the base forms of new words. Some entries for the verb "gledati" (to look) are the following: "`gleda` → `gledati Vmip3s-n`, `gledaš` → `gledati Vmip2s-n`, `gledajo` → `gledati Vmip3p-n`". When encountering the unknown word "igrajo" (they play), the system can correctly suggest the base form "igrati" (and the tag "`Vmip3p-n`").

In unsupervised learning, there is no teacher providing desired answers, but since the data are not entirely random, there are statistical regularities that can be captured and that can be applied autonomously in new cases. One example is the discovery of clusters in a data set, e.g., the grouping of words that occur in similar sentential contexts. The categories that emerge, fully without supervision, can be conceptually appealing: nouns are separated from verbs, and subgroups, such as animate and inanimate nouns, can be distinguished; see e.g., Honkela et al. (1995).

Reinforcement learning corresponds to something between the supervised and unsupervised approaches. It differs from supervised learning in the sense that explicit input-output pairs are not available. In reinforcement learning, an agent explores an environment (real or simulated) and is able to take actions. Depending on the outcome of the series of actions taken, the agent is rewarded or penalized. The strive for maximal reward makes the agent learn over time to improve its behavior. In order to apply reinforcement learning in NLP, true situations of interaction with humans would be needed. There are experiments with robots in which a common language emerges in the robot community (e.g., Steels, 1997), but naturally this robot language is not an existing human language.

2.2.2 Learning Methodologies

Different traditions have led to the emergence of different machine learning methodologies, the most central being *rule-based artificial intelligence*, *artificial neural networks*, *evolutionary modeling*, and *statistical learning*.

Traditional artificial intelligence (AI) is based on logic and the learning aims at discovering *logical rules* describing regularities in the data. Induction of *decision trees* and *Inductive Logic Programming* (ILP) are examples of the rule-based approach. Decision trees represent Boolean functions and can be used for performing classification. For instance, Vasilakopoulos (2003) utilizes an induced decision tree for labeling unknown words with part-of-speech tags. Inductive Logic Programming systems (Muggleton and Raedt, 1994) try to derive a logic program from a database of facts and expected results, which are divided into positive and negative examples. Kazakov and Manandhar (2001) generate word segmentation rules from a raw list of words using ILP in combination

with genetic algorithms (see below).

Artificial neural networks draw inspiration from the functioning of living brains. *Multilayer perceptrons* (MLP) are a type of neural networks, which serve as universal approximators of any continuous mappings from M-dimensional inputs to N-dimensional outputs. A classical example of the use of an MLP in natural language processing is NETtalk (Sejnowski and Rosenberg, 1987), which was able to convert English text to speech. In the training phase, the network was supplied with the correct pronunciation of 1000 frequent words. Another highly influential work (Rumelhart and McClelland, 1986) concerns the learning of the past tense form of English verbs; a pattern associator network ended up mimicking the behavior of children acquiring both regular and irregular forms (including incorrect regularized forms, such as "goed" instead of "went"). *Self-organizing maps* (SOM) exemplify a network architecture that learns without supervision (Kohonen, 1982, 2001). A SOM transforms an incoming signal pattern of arbitrary dimensionality into a usually two-dimensional discrete map in a topologically ordered fashion. As a consequence, similar inputs will be located close to each other on the map. Self-organizing maps have been used in a very broad range of NLP applications, including the following: exploratory information retrieval (Honkela, 1997; Kohonen et al., 2000), analysis of conceptual similarities of words (Honkela, 1997; Lagus et al., 2002), word sense disambiguation (Lindén, 2003), and acquisition of morphological categories (Pirrelli et al., 2004).

In nature, not only do individual animals learn to perform better, but species evolve to better fit in their niches. **Evolutionary modeling**, or *genetic algorithms*, imitate natural evolution in artificial environments and fit well into the reinforcement learning paradigm. In a typical setting, one or more individuals (agents or computer programs) are put to a task and those who perform the best are allowed to reproduce. The reproduction is often combined which some random "mutations" of the genes of the offspring. Evolutionary modeling is the natural choice for simulations of emergence and evolution of language (e.g., Steels, 1997; Kirby, 2002). Kazakov and Manandhar (2001) apply a genetic algorithm in order to discover a "naïve" morphology, where words are always split into two parts.

A problem that intrigues statisticians is how best to use samples drawn from unknown probability distributions to help decide from which distribution some new sample is drawn. A related problem is how to estimate the value of an unknown function at a new point, when values of this function are known for some set of points. Making such inferences can be called **statistical learning**. Statistical learning is the methodology used in the current work and the entire following section is dedicated to this topic.

2.2.3 Statistical Learning

The current presentation of statistical learning focuses on *Bayesian* probability theory (pages 23–26). Some central issues will be illustrated through an example: the inductive inference problem, maximum likelihood vs. maximum a posteriori optimization, and the different views on probabilities advocated by Bayesians and frequentists.

There is an alternative interpretation of statistical learning, based on *information the-*

ory. In information theory, the information content of a message is quantified into *bits* according to statistical properties of the data. Information-theoretic models comprise the Minimum Description Length (MDL) and Minimum Message Length (MML) formalisms (pages 27–29). Some further models are mentioned briefly.

Probabilistic Modeling and Maximum Likelihood Optimization

Traditional rule-based systems can make inferences successfully, e.g., TWOL might tell us that the Finnish word "kirjassa" is the inessive singular of the noun "kirja" ("book"). However, sometimes the outcome is ambiguous, e.g., the form "kirjasta" may be the elative singular of "kirja" or the partitive singular of "kirjanen" ("little book, booklet"). The latter interpretation is less likely than the former if no additional information is available, since "kirja" is a much more frequent word than "kirjanen". The rule-based TWOL, however, has no means of expressing this asymmetry, but gives the two alternatives on an equal footing.

The problem can be remedied by introducing probabilities in the model, that is, by turning to *probabilistic modeling*. Probabilistics provides a means for ranking different interpretations of the data, e.g., how likely it is that "kirjasta" means "book" vs. "booklet". *Additionally*, probabilities can be utilized for selecting a suitable model among a broad selection of possible models. A model can here be seen as a *lexicon of parameter values*. The more parameters are used, the more expressive the model is, i.e., the more degrees of freedom it has.

For instance, suppose that there is a family of simplistic models, each of which consists of a lexicon of morphs. Lexicons emerge from a stochastic process, where letters are chosen by random. The alphabet consists of the 26 lower-case letters in the English alphabet and of a morph separator (space). For simplicity, all letters (including space) have equal probability, i.e., $\frac{1}{27}$. The lexicon is generated by iteratively producing letters by random until two spaces are obtained in a row. Each letter position in the lexicon can be considered as a parameter and the actual letter that occupies that position can be considered as the value of that parameter.

The lexicon is a morph collection in the sense that each space-delimited string is a morph. The probability of the lexicon depends on its size (the number of free parameters in it). Some possible lexicons and their probabilities are:

Lexicon 1 "a␣c␣e␣g␣i␣j␣l␣m␣n␣o␣p␣r␣t␣u␣␣", $P(\textit{Lexicon 1}) = (\frac{1}{27})^{29}$

Lexicon 2 "apple␣juice␣lemon␣orange␣tree␣␣", $P(\textit{Lexicon 2}) = (\frac{1}{27})^{31}$

Lexicon 3 "apple␣applejuice␣appletree␣juice␣lemon␣lemontree␣orange␣orangejuice␣␣", $P(\textit{Lexicon 3}) = (\frac{1}{27})^{69}$

Lexicon 1 consists of 29 letters, *Lexicon 2* of 31, and *Lexicon 3* of 69 letters. The larger the lexicon is, the higher the number of possible configurations is, and the smaller the probability is, that the lexicon actually looks exactly as it happens to do. Therefore, *Lexicon 1*, which is smallest, is the most probable model, and *Lexicon 3* which is largest, is the least probable model, *a priori*.

Given these three models (lexicons), it is possible to compute probabilities for a small data set, namely the word list: "apple, orange, lemon, juice, applejuice, orangejuice, appletree, lemontree". We continue to keep the task simple, and assume that all morphs in a lexicon are equally likely to occur. In addition, we assume that there is word boundary morph (#), which is as likely as the other morphs in the lexicon.

Lexicon 1 contains 14 morphs (which are actually individual letters). Adding the word boundary morph makes them 15 and the probability of observing any of the morphs becomes $\frac{1}{15}$. The representation of the data using *Lexicon 1* looks as follows, where spaces indicate morph boundaries and number signs word boundaries: "`a p p l e # o r a n g e # l e m o n # j u i c e # a p p l e j u i c e # o r a n g e j u i c e # a p p l e t r e e # l e m o n t r e e # #`". The sequence consists of 69 morphs and its probability conditioned on *Lexicon 1* is thus: $P(\textit{Data} \,|\, \textit{Lexicon 1}) = (\frac{1}{15})^{69} \approx 7.1 \cdot 10^{-82}$.

Lexicon 2 contains only five morphs. With inclusion of the word boundary morph, the probability of observing one of the morphs is thus $\frac{1}{6}$. *Lexicon 2* produces the following representation for the data, consisting of 21 morphs: "`apple # orange # lemon # juice # apple juice # orange juice # apple tree # lemon tree # #`". The probability of the data when *Lexicon 2* is given as a model is: $P(\textit{Data} \,|\, \textit{Lexicon 2}) = (\frac{1}{6})^{21} \approx 4.6 \cdot 10^{-17}$.

Lexicon 3 contains eight morphs, which are in fact identical to the words in the data. Including the word boundary morph, the uniform morph occurrence probability is thus $\frac{1}{9}$. Now there is no longer an unambiguous morph segmentation of the data, because both "applejuice" and "orangejuice" can be represented in two ways: "`applejuice #`" vs. "`apple juice #`", and "`orangejuice #`" vs. "`orange juice #`", respectively. Since all individual morphs have the same probability, the alternatives with a smaller number of morphs are more likely, $(\frac{1}{9})^2 > (\frac{1}{9})^3$. The most likely segmentation of the entire data thus consists of 17 morphs: "`apple # orange # lemon # juice # applejuice # orangejuice # appletree # lemontree # #`". Consequently, the probability of the data conditioned on *Lexicon 3* is: $P(\textit{Data} \,|\, \textit{Lexicon 3}) = (\frac{1}{9})^{17} \approx 6.0 \cdot 10^{-17}$.

When the lexicons were compared according to their *prior probabilities* the following ranking was obtained: $P(\textit{Lexicon 1}) > P(\textit{Lexicon 2}) > P(\textit{Lexicon 3})$. However, as has also been demonstrated, if the lexicons are compared according to their likelihood with respect to the data, another (the opposite) ranking ensues: $P(\textit{Data} \,|\, \textit{Lexicon 3}) > P(\textit{Data} \,|\, \textit{Lexicon 2}) > P(\textit{Data} \,|\, \textit{Lexicon 1})$. That is, *Lexicon 3* provides the best fit for the data, whereas *Lexicon 1* provides the worst fit. Selecting the model that assigns the highest probability to the data is called *maximum likelihood* (ML) optimization.

Inductive Inference Problem

Inductive inference is the process of reaching a general conclusion from specific examples. This means that one can propose a model based on available observations and hope that this model will serve well for *prediction* in new situations. The challenge, of course, is how to discover a good model in general. This problem will be illustrated using the three lexicon models introduced above.

Lexicon 3 provides the best fit for the data available, which can be measured as the probability it assigns to the data. However, as such a lexicon is a very large model, it is flexible in the sense that it can learn the data very accurately, and in fact capture too many restrictions that by chance happen to be present. The model *overfits* the data by learning it by rote. This means that the prediction power of the model is poor. For instance, the probability assigned to the possible word "orangetree" is zero, $P(\text{“}\texttt{orangetree}\,\#\text{”}\,|\,\textit{Lexicon 3}) = 0$.

Lexicon 1 does not provide a very good fit for the observed data. The restrictions that apply are hardly modeled at all, which means that predictions made by the model are not reliable. In contrast to *Lexicon 3*, this model generalizes to new word forms to a larger extent. For example, the possible word "orangetree" has the probability: $P(\text{“}\texttt{o r a n g e t r e e}\,\#\text{”}\,|\,\textit{Lexicon 1}) = (\frac{1}{15})^{11}$. This is not much lower than the probability of the observed word "lemontree": $P(\text{“}\texttt{l e m o n t r e e}\,\#\text{”}\,|\,\textit{Lexicon 1}) = (\frac{1}{15})^{10}$. However, the nonsense string "akvppkrb" is more probable than both of the previous words, because in this model the probability of any word is simply a function of its length in letters: $P(\text{“}\texttt{a k v p p k r b}\,\#\text{”}\,|\,\textit{Lexicon 1}) = (\frac{1}{15})^{9}$. *Lexicon 1 underfits* the training data and therefore *over*generalizes to new data.

In this particular scenario, *Lexicon 2* appears to be a good compromise. It assigns the same probabilities to the possible words "orangetree" and "lemonjuice" as to, for instance, the observed words "appletree" and "orangejuice": $P(\text{“}\texttt{orange tree}\,\#\text{”}\,|\,\textit{Lexicon 2}) = (\frac{1}{6})^{3} = P(\text{“}\texttt{lemon juice}\,\#\text{”}\,|\,\textit{Lexicon 2}) = P(\text{“}\texttt{apple tree}\,\#\text{”}\,|\,\textit{Lexicon 2}) = P(\text{“}\texttt{orange juice}\,\#\text{”}\,|\,\textit{Lexicon 2})$. By contrast, the nonsense string "akvppkrb" has zero probability in this model.

Bayesian Inference

In the fruit words example, *Lexicon 2* is by far the most likely model, if both the prior probability of the model, $P(\textit{Lexicon X})$, and the probability of the data conditioned on the model, $P(\textit{Data}\,|\,\textit{Lexicon X})$, are taken into account: $P(\textit{Lexicon 2}) \cdot P(\textit{Data}\,|\,\textit{Lexicon 2}) \approx 1.9 \cdot 10^{-61} > P(\textit{Lexicon 3}) \cdot P(\textit{Data}\,|\,\textit{Lexicon 3}) \approx 1.0 \cdot 10^{-115} > P(\textit{Lexicon 1}) \cdot P(\textit{Data}\,|\,\textit{Lexicon 1}) \approx 2.2 \cdot 10^{-123}$. In this comparison the complexity of the model has been balanced against the fit of the training data, which favors a good compromise, that is, a model that does not overlearn and that adequately generalizes to unseen data. This can be seen as a mathematical application of the minimalist principle called Occam's razor or the principle of parsimony: "Pluralitas non est ponenda sine necessitate", in other words: "Keep things as simple as possible, but no simpler". The latter is claimed to be a paraphrase by Albert Einstein; see e.g., Gibbs (1996).

The proposed model selection procedure is based on maximizing the *posterior probability* of the model, $P(\textit{Lexicon X}\,|\,\textit{Data})$. The posterior can be rewritten using Bayes' rule:

$$P(\textit{Lexicon X}\,|\,\textit{Data}) = \frac{P(\textit{Lexicon X}) \cdot P(\textit{Data}\,|\,\textit{Lexicon X})}{P(\textit{Data})}. \tag{2.1}$$

It can be cumbersome to estimate $P(\textit{Data})$, the prior probability of the data set itself. This is the probability that, in general, we would come across data that actually look exactly like the data set we happen to have. If, however, the intention is to compare

different models on the very same data set, the probability of the data is a constant that does not affect the result of the comparison. Therefore, the probability of the data can be ignored, and we obtain the product used above:

$$P(\textit{Lexicon X} \mid \textit{Data}) \propto P(\textit{Lexicon X}) \cdot P(\textit{Data} \mid \textit{Lexicon X}). \tag{2.2}$$

In Bayesian inference, in principle, one does not choose the one most likely model when predictions are made about new observations. Rather, all possible models participate in the prediction, but, the more likely a specific model is, the more "trust" is put in it. This optimal result is obtained by weighting the prediction made by each individual model by the posterior probability of that model. The estimated probability that a particular new piece of data is observed (*New Obs*), when a specific data set is given, is thus:

$$P(\textit{New Obs} \mid \textit{Data}) = \sum_{X} P(\textit{New Obs} \mid \textit{Lexicon X}) \cdot P(\textit{Lexicon X} \mid \textit{Data}). \tag{2.3}$$

If the number of possible models is very high, as is usually the case, optimal Bayesian prediction is infeasible. A common approximation is to use the *maximum a posteriori* (MAP) estimate. This implies that after all, only the one model with the highest posterior probability is chosen.

In the fruit words example, the posterior probability of *Lexicon 2* is much higher than those of *Lexicon 1* and 3. Therefore, *Lexicon 2* would dominate completely in the weighted sum (2.3), and the predicted probability of a new word, e.g., "orangetree", $P(\text{“}\texttt{orangetree\#}\text{”} \mid \textit{Data})$, would be very close to the value based on the MAP estimate, $P(\text{“}\texttt{orangetree\#}\text{”} \mid \textit{Lexicon 2})$. This is, however, not always the case. Typically, there may be multiple competing models that have nearly equal probability.

Bayesian and Frequentist View on Probabilities

There are two major formulations of probability theory, that of the Bayesians and that of the frequentists. The frequentist view is that probabilities can only be *relative frequencies of occurrence*. For instance, if a database of English texts contains 50 million letters, and the letter "y" occurs one million times, the probability of "y" can be estimated to be $\frac{1}{50}$.

In contrast to the frequentists, Bayesians use probabilities for expressing *degrees of belief*. For example, prior to observing any corpus of English text, one could suggest that the probability of the letter "y" is $\frac{1}{26}$. That is, if one only knows that there are 26 different letters in the English alphabet, one might want to assign the same probability to each letter, a priori.

There are situations, in which probabilities make sense only from a Bayesian point of view, e.g., the probability that there is life on Mars. For a frequentist, there is one planet Mars, and there either is life there or not.

Minimum Description Length

As an alternative to the Bayesian approach, Jorma Rissanen introduced the Minimum Description Length (MDL) principle in 1978 (Rissanen, 1978). MDL is based on information theory. Since its introduction, MDL has undergone changes, and when talking about MDL one should be careful to specify which version of MDL one has in mind. Regardless of version, the fundamental idea of MDL is to view data compression as a basis for inductive inference. Any regularity in data can be used for compressing the data. Therefore, the more compact description one can obtain for a data set, the more regularity one has discovered and the more one has learned about the data.

Another consistent theme in the MDL methodology is the rejection of Bayesian prior probabilities. The Bayesian approach leaves room for subjectivity and in principle it is possible to "tailor priors for each occasion" by always choosing prior distributions that give a high posterior probability to the specific data set at hand. Such bad modeling practice is naturally not endorsed by the Bayesians, but the aversion against subjective criteria is more outspoken in the MDL philosophy.

In *Ideal MDL*, the goal is to discover the shortest possible computer program (Turing machine) that prints the desired data sequence and then halts. It turns out that the more regular a sequence is, the shorter the program code required for producing it is. The length of the code is called the *Kolmogorov complexity*, or *algorithmic complexity*, of the data sequence and it can be measured, e.g., in number of *bits* (binary digits); see, e.g., Grünwald et al. (2005). Theoretically, since for every imaginable data set there is an associated shortest program code, it is possible to define a probability distribution over all possible data sets. The probability of a specific data set is the probability of producing the corresponding program code, i.e., a sequence of zeros and ones. The probability that such a bit sequence emerges is $(\frac{1}{2})^d = 2^{-d}$, where d is the Kolmogorov complexity, i.e., the code length in bits. Thus, even though MDL does not adhere to the Bayesian notion of probability, it also abandons the frequentist view, because probabilities are related to code lengths rather than to relative frequencies of occurrence in some empirical test setting.

The Kolmogorov complexity is an objective optimization criterion, but it is of little practical use, since it can be shown that it is impossible to design an algorithm that computes the Kolmogorov complexity of an arbitrary data set. In practice, one has to choose coding schemes that are less expensive than general-purpose programming languages. Thus, one restricts the model search to a limited model family that one judges appropriate for the task and gives up the idealized goal of discovering every possible regularity that is present in the data.

The most wide-spread application of practical MDL in natural language processing is based on a *two-part coding scheme* (Rissanen, 1989). This approach has been rebaptized by Grünwald et al. (2005) as *Crude MDL* in contrast to a more recent version, called *Refined MDL*. Crude MDL picks the model that minimizes the sum $L(\mathit{Model}) + L(\mathit{Data}\,|\,\mathit{Model})$, where $L(\mathit{Model})$ is the description length for coding the values of the model parameters and $L(\mathit{Data}\,|\,\mathit{Model})$ is the description length for coding the data sequence conditioned on the particular model. When the data sequence is long enough, its optimal code length approaches the negative logarithm of the *probability* of

the data conditioned on the model: $L(\textit{Data} \mid \textit{Model}) \approx -\log_2 P(\textit{Data} \mid \textit{Model})$. Similarly, $L(\textit{Model})$ can be associated with a probability distribution such that $L(\textit{Model}) \approx -\log_2 P(\textit{Model})$. This implies that the minimization of the two-part MDL sum is equivalent to maximizing the corresponding posterior probability, proportional to the product $P(\textit{Model}) \cdot P(\textit{Data} \mid \textit{Model})$. Therefore, Crude MDL can be considered equivalent to maximum a posteriori (MAP) optimization (see pages 25–26).

From the point of view of "objectivity", the problem with Crude MDL is how to come up with the *Model* part of the two-part code. One can use an "intuitively reasonable" coding scheme, but this is not satisfactory, since the description length $L(\textit{Model})$ can be large under one code and small under another. In Crude MDL there is no principled mechanism for banning "subjective" codes and imposing "objective" ones.

Refined MDL differs considerably from Crude MDL. This modern version was introduced in the late nineties (Rissanen, 1996; Barron et al., 1998) and a rather accessible presentation can be found in the tutorial by Grünwald et al. (2005). Refined MDL employs a *one-part* rather than two-part code, due to the fact that the code is associated with an entire model class instead of one specific model (i.e., set of optimal parameter values). For instance, in the task of fitting a curve to a specific sequence of data points, Refined MDL might choose between the class of third-grade polynomials and that of fourth-grade polynomials, but it does not primarily state whether some $y = x^3 - 2x^2 + x - 5$ is to be preferred over $y = \frac{1}{9}x^4 + x^3 - 3x^2 + 2x - 4$. The description length that is calculated for a particular data sequence depends on two balancing factors: (1) How good a fit to the data can be obtained using the suggested model class, i.e., how good is the best-fitting model within the class? A good fit is reflected in a short code length. (2) Does the class additionally contain models that fit well to many other data sequences of the same length as the data set at hand? If this is the case, the *parametric complexity* of the model class is high, which results in longer code length. The parametric complexity is related to the degree-of-freedom of the model class, but it also takes the geometrical structure of the class into account. The parametric complexity equals the logarithm of the number of essentially different, distinguishable models within the class.

For large samples, the one-part code of Refined MDL can be given a two-part code interpretation after all: Models in a class are encoded by first discretizing the model space into a set of "maximally distinguishable models", and then assigning *equal code length to each of these*. After this, the data are encoded using the selected model. In any case, Refined MDL reduces the room for "subjectivity" to deciding upon which model classes to compare (e.g., polynomials of different degree). Once this is done, the probability distributions used for different parameter values emerge implicitly, in an "objective" manner.

From the point of view of data compression, Refined MDL theoretically produces shorter codes than any two-part encoding schemes. Two-part codes are inherently redundant due to the fact that they fail to take into account that one would never encode a data set conditioned on just any model, but only on the *optimal* model within the model family under consideration. That is, when one knows the first part of the code, i.e., the values of the model parameters, one can also infer that the second part of the code can only represent data sets for which this particular parameter configuration is optimal.

Minimum Message Length and Other Methods

The Minimum Message Length (MML) method (Wallace and Boulton, 1968; Wallace and Freeman, 1987) is closely related to MDL. The two methods are not seldom mistaken one for another. MML is based on a two-part code almost identical to that of Crude MDL. In contrast to Rissanen, Wallace and Freeman (1987) adhere to the philosophical view that prior information always exists and that it should be formulated mathematically as well as possible: "The health of Bayesian statistics can only be undermined by any return to the notions of ignorance current in the 1960's and 70's." Thus, MML relies on the Bayesian notion of probability and can be seen as a direct coding-scheme variant of MAP optimization.

The advocates of MML disagree with the later objectives of the MDL research community. Even though Refined MDL may provide a slightly more compact description of the data, it proposes a model class rather than a fully-specified model. Wallace and Dowe (1999) claim that the discovery of fully-specified models is more useful in many situations. In some problems, two models within the same class may have markedly different conceptual structure, which makes it questionable how to define meaningful classes in the first place.

The described methods for inductive inference (ML, full Bayesian, MAP, MDL, MML) rest on solid theoretical ground and their scope of application is very general. Fundamental issues in statistical learning theory (SLT) are also addressed by Vapnik (1998), where theoretical results are presented concerning the amount of data necessary for obtaining a desired approximation at a desired level of confidence. (For a more compact introduction to SLT, consult, e.g., Evgeniou et al., 2000) Under some circumstances less general and computationally less intensive methods for model selection might be preferred, e.g., Akaike's information criterion (AIC) (Akaike, 1974) and the Bayesian information criterion (BIC) (Schwarz, 1978). The latter is a quick approximation of a Bayesian model comparison procedure.

2.2.4 Vector Space Models

The treatment of machine learning in NLP is concluded with a discussion on vector representations for linguistic units (e.g., words and morphemes). This topic may not have direct relevance to the models implemented as a part of the current thesis, but it is important for the understanding of some related work.

Morphemes that have similar semantic and syntactic properties typically occur in similar distribution. That is, the context of other morphemes in which these morphemes occur are similar. In a machine learning setting, the context of a morpheme can be represented as a *vector*, which is a mathematically tractable object. A simple way to construct a context vector for a target morpheme is to accumulate counts for each morpheme that occurs within a fixed-sized window around the occurrences of the target morpheme in the corpus. The value on each dimension of the vector represents the number of occurrences of a specific morpheme. Relying on such a representation, the similarity of morphemes can be calculated using vector algebra. One common measure is the *cosine* of the angles between two vectors. (Consult a standard text book for concrete examples,

e.g., Manning and Schütze, 1999.)

Since the number of distinct morphemes is generally very large, the dimensionality (i.e., size) of the vectors is also very large. Moreover, most values in most vectors are zero, because most morphemes only occur in the context of a few other morphemes. A more compact representation can be obtained, e.g., using *random projection*, which rather accurately preserves the mathematical properties of full-sized vectors (Kaski, 1998). Further experiments and references can be found in Bingham and Mannila (2001).

In addition to calculating similarities between different morphemes, one may be interested in discovering categories, possibly with hierarchical structure. *Clustering* techniques can be utilized for this purpose, whereby similar morphemes are grouped together; see, e.g., Manning and Schütze (1999); Schalkoff (1992). Also the Self-Organizing Map (Section 2.2.2) can be utilized for clustering.

A further step of analysis consists in hypothesizing that the observed variables (i.e., the dimensions in the vectors) can be explained using a smaller number of underlying variables and that the effect of these underlying variables is manifested jointly in the set of observed variables. In *Principal Component Analysis* (PCA) a new coordinate system is fitted to the data points, such that as much variance in the data as possible is captured by the first dimensions. The last dimensions have but minor significance and can be left out, which results in a reduction of the original dimensionality; see e.g., Haykin (1999); Hyvärinen et al. (2001). In natural language processing, an extension of PCA called *Latent Semantic Analysis* (LSA) is frequently used, e.g., in information retrieval (Deerwester et al., 1990).

Independent Component Analysis (ICA) resembles PCA, but the aim is to discover components that are statistically independent in addition to being decorrelated as in PCA. Within natural language processing, promising applications of ICA include work by Bingham (2003), Honkela and Hyvärinen (2004), Väyrynen and Honkela (2005), and Lagus et al. (2005).

Vector space models can be applied within both the statistical and neural learning methodologies mentioned in Section 2.2.2. Not surprisingly there are connections between the two. For instance, it has been shown that neural networks can learn the principal components of the statistical PCA method (Oja, 1982).

2.3 Automatic Speech Recognition

In automatic speech recognition (ASR), speech is transcribed into text by a machine. The recognition system is comprised of a number of components, including *acoustic models*, a *lexicon*, and a *language model*. The acoustic models represent sounds of the language. Each phoneme is typically modeled separately, but the immediate context of the phoneme is taken into account; for instance, the /h/ sound in the Finnish words “vaha” (wax) and “vihko” (notebook) are pronounced differently, due to different phonemic neighborhoods. Acoustic models are more or less specific to a particular speaker under specific conditions of background noise. Adaptation techniques may be necessary in order to use the system in new situations (e.g., Viikki, 1999).

The lexicon of the ASR system consists of the recognizable vocabulary, typically a list of words. Any other, out-of-vocabulary (OOV) words cannot be recognized correctly. In a language, some word sequences are very likely to occur, while others are utterly improbable. The role of the language model is to determine the probability of word sequences. Taken together, the lexicon and language model allow the recognizer to make "good guesses" of what is uttered, given the speech signal. Humans are known to understand speech in a similar fashion: Our knowledge and expectations determine what we hear, since the plain audio signal is often noisy.

In the following, only issues related to language modeling are covered. However, a complete speech recognition system involves many other components. Central questions are the *extraction of relevant features* from the audio signal and the *decoding* of the signal, i.e., the search for the most probable recognition hypothesis. Rabiner and Juang (1993) present the mathematical foundations of speech recognition based on *Hidden Markov Models* (HMM's), which is the standard approach in modern ASR systems. The reader can alternatively consult the manual of the HTK speech recognition tool kit (Young et al., 1999) or the accessible introduction by Jurafsky and Martin (2000).

2.3.1 Basics of Language Modeling

Language models can be seen as probabilistic models of syntax, which determine the probability of word sequences, $w_1 \dots w_i$. The probability $P(w_1 \dots w_i)$ is often decomposed into a chain of products of probabilities, where the language model predicts the following word, when the history of observed words is known:

$$P(w_1 \dots w_i) = P(w_1) \cdot P(w_2 \,|\, w_1) \cdot \ldots \cdot P(w_i \,|\, w_1 \dots w_{i-1}). \tag{2.4}$$

For instance, the history could consist of the word sequence $w_1 \dots w_7$: "I like icecream, but I don't like . . .", and the language model might suggest the words "chocolate, fudge", and "milkshake" as likely continuations (w_8), whereas the words "they, says", and "afterwards" would be unlikely.

n-Grams

Since it is usually infeasible to compute $P(w_i \,|\, w_1 \dots w_{i-1})$ for large values of i, the history is generally truncated to a fixed number of previous words. This is the so-called Markov assumption:

$$P(w_i \,|\, w_1 \dots w_{i-1}) \approx P(w_i \,|\, w_{i-(n-1)} \dots w_{i-1}). \tag{2.5}$$

In an n-gram of order n the prediction of the next word is based on the $n-1$ preceding words. If one were to apply a trigram ($n = 3$) in the "icecream example", the eighth word would be predicted from the word sequence "don't like". Naturally, this short context is much less informative than the longer history. The trigram model might assign high probabilities to continuations, such as "it" or "about" (cf. "What I *don't like about* this food . . ."), although these alternatives do not seem plausible in the current larger context.

Nevertheless, n-grams (especially trigrams) have traditionally been the most used language modeling technique, both in research and applications, due to their simplicity and surprising effectiveness.

Smoothing

A language model estimated directly from a training corpus is biased toward that corpus. Many perfectly valid word sequences do not occur in the corpus and therefore have zero probability in the estimated model. By contrast, many observed word sequences, which are very specific to the corpus, obtain probability estimates which are too high. (For instance, the word sequence "the Morfessor model" is expected to be over-represented in this thesis compared to a larger collection of English texts.)

The purpose of *smoothing* techniques is to transfer some probability mass from over-represented events to under-represented ones. Typically this takes place through *discounting*, which implies that the maximum likelihood estimates obtained for the observed events are decreased by a discounting term. The "removed" probability mass is then reserved for new events, i.e., events that were not observed in the data.

Smoothing is often combined with backing-off or interpolation. In backoff models, the probability distribution of the predicted word is based on the longest observed history sequence (up to some value $n - 1$). All words following the different occurrences of this sequence are likely continuations. Thus, if the continuation of the phrase "I like icecream, but I don't like ..." is predicted using backoff 4-grams ($n = 4$), one first looks up occurrences of the three-word sequence "I don't like" in the training data. If this sequence does not occur, one backs off to "don't like", and further to "like" until occurrences of the desired word sequence is found. The last resort is to assign a probability to the predicted word independent of context, according to the relative frequency of this word in the training data.

In interpolation methods, the probability estimates of several sources (e.g., n-grams of different order) are combined into one final estimate. For instance, when predicting the successor word of our example phrase, the observed continuations of the following word sequences are probable candidates, with more weight on the longest observed contexts: "like", "don't like", "I don't like", etc.

A more thorough discussion about smoothing can be found in, e.g., the books by Manning and Schütze (1999) and Jurafsky and Martin (2000). Two smoothing techniques, in particular, are commonly used in language modeling for ASR: Until recently *Katz backoff* (Katz, 1987) was the state-of-the-art method, but *Kneser-Ney smoothing* (Kneser and Ney, 1995) is now taking its place. The so-called *Modified* Kneser-Ney smoothing method, which utilizes interpolation, has been shown to outperform other smoothing techniques (Chen and Goodman, 1999; Goodman, 2001a).

Beyond n-Grams

Basic n-gram models are unsophisticated from a linguistic point of view, as they essentially do nothing but memorize a large number of word sequences observed in the

training data. n-Grams miss dependencies extending beyond n consecutive words and they are poor at generalizing, e.g., seeing the parallels between the phrases "I'll eat fish for dinner." and "You drank coffee for breakfast."

A number of language modeling approaches have been suggested, aimed at increasing the generalizability of the models and at alleviating the data sparsity problem. Words can be grouped into *classes* (or *clusters*) of resembling words, either based on linguistic or data-driven statistical criteria. Words belonging to the same class behave similarly and can thus occur in similar contexts (e.g., "dinner" and "breakfast" could belong to the same class). The classes can be hard or soft, the former implying that each word neatly falls into one single class, and the latter implying that a word can belong to several classes with different strength. The following methods, among others, have been applied in class-based language modeling (in a broad sense of the term): *Hard clustering based on simple statistical techniques* (Brown et al., 1992), *Context-free grammars* (Lari and Young, 1990; Banerjee and Rosenfeld, 1992; Stolcke, 1995), *Latent Semantic Analysis (LSA)* (Bellegarda, 2000), and *neural networks* (Bengio et al., 2000). Additionally, in the language modeling of Finnish, the *Self-Organizing Map* (SOM) has been used by Siivola et al. (2001), and initial experiments using *Independent Component Analysis* (ICA) have been performed by Virpioja (2005).

In *factored language models*, words are viewed as bundles of features, or factors. Factors can be anything, including morphological classes, stems and data-driven word classes (Bilmes and Kirchhoff, 2003). Yet other models, abandoning the strict left-to-right anatomy of standard n-grams, comprise *skip-grams*, *structured language models* (Chelba and Jelinek, 2000), *decision tree models*, *Maximum Entropy models* (Rosenfeld, 1996, 1997), and *topic models* including *caching models* (Iyer and Ostendorf, 1999). Topic models are typically implemented as *mixture models*, where one interpolates probabilities estimated from separate data subsets, each representing a different topic or style. A topic model for Finnish (and English) is proposed by Kurimo and Lagus (2002); Lagus and Kurimo (2002).

Despite the multitude of more sophisticated model types, it remains rather difficult to obtain significant improvements over standard, high-order, appropriately smoothed, n-grams trained on very large corpora.

2.3.2 Limiting the Size of the Lexicon

It is important to have a lexicon and a language model with a good coverage of the likely utterances of the language. Problems arise with languages with rich, productive word forming, such as Finnish. Even if all unique word forms are collected from a large corpus, many perfectly valid words will not be contained in this set. Additionally, the large mass of rare words will lead to a poor language model, since the estimated probabilities are based on very few observations of each unique word.

A possible solution is to split words into shorter, more frequently occurring, segments. Instead of entire words, these segments are collected into the lexicon, and the language model is estimated from sequences of such word segments. Linguistically, morphemes are a well motivated choice. New word forms can be created productively by a combina-

tion of morphemes. In addition, a morpheme is a unit with the same syntactic behavior and approximately the same meaning throughout the contexts it occurs. Thus, morphemes could be the basis of a language model that incorporates some knowledge of the grammar of the language.

The use of morphemes alleviates the problem of vocabulary growth, but it does not solve it completely. A large mass of the vocabulary in any field is composed of names of people, places or items. Many of these names consist of one single morpheme. Consequently, gains in vocabulary size can be made only be splitting the morphemes further into sub-morphemic fragments, e.g., syllables. On the one hand, the advantage of such an approach is that rare names can be recognized by a model that captures typical sound sequences of the language. On the other hand, the morpheme as a whole is not recognized as an entity, which is a disadvantage in the modeling of syntax and semantics.

2.3.3 Evaluation of Language Models

The performance of a language model is ultimately determined by its contribution to the overall performance of the speech recognition application where it is used. However, language models also need to be evaluated in their own right, since speech recognition experiments are computationally expensive as well as time-consuming and there are many factors other than the language model that affect the performance of the ASR system (such as the acoustic modeling).

The quality of a language model is customarily assessed by computing the probability assigned by the model to an *independent* test set. This test set consists of data that were not used in the training of the language model. Since the test set probability depends strongly on the length of the text, derivative measures, normalized over the number of words in the data, are typically used, the most common being *perplexity* and *cross-entropy* (Chen and Goodman, 1999).

The perplexity is calculated as:

$$\textit{Perplexity}(w_1 \dots w_{N_T} \,|\, \theta) = P(w_1 \dots w_{N_T} \,|\, \theta)^{-\frac{1}{N_T}}, \tag{2.6}$$

where the test set consists of the word sequence $w_1 \dots w_{N_T}$ and $P(w_1 \dots w_{N_T} \,|\, \theta)$ is the probability assigned to this sequence by the language model θ. Note that the probability can be computed as a chain of products (Eq. 2.4) or as n-grams (Eq. 2.5), etc. It is not required that the probability be based exclusively on probabilities of *word* sequences; sequences of other fragments (e.g., morphs or individual letters) are equally suitable. However, if language models operating on different fragment inventories are compared, it is important that they all be normalized on the same number of units, such as the total number of words in the test set, N_T (in the exponent of Eq. 2.6).

The perplexity represents the average number of equiprobable choices at each word boundary. A perplexity of p corresponds to a situation, where on average, the following word needs to be drawn from a set of p words. Thus, the lower the perplexity of the model is, the better the model is at predicting the following word.

Rosenfeld (2000) addresses the problem of predicting speech recognition error rates from the perplexity of a language model: “Error rates are typically nonlinear and poorly

understood functions of the language model. Lower perplexity usually results in lower error rates, but there are plenty of counterexamples in the literature. As a rough rule of thumb, reduction of 5 % in perplexity is usually not practically significant; a 10 %–20 % reduction is noteworthy, and usually (but not always) translates into some improvement in application performance; a perplexity improvement of 30 % or more over a good baseline is quite significant (and rare)."

An alternative, but equivalent measure to perplexity is *cross-entropy*. The cross-entropy is the two-base logarithm ($\log_2$) of perplexity and it corresponds to the average number of bits per word that would be required to encode the test material using an optimal coder of the model.

As the result of a theoretical analysis, Goodman (2001b) hypothesizes a *linear relationship between cross-entropy and word error rate*. He also refers to previous work that seems to support his claim. However, many researchers report results where entropy and word error rate do not correlate. Nevertheless, Goodman's hypothesis implies that very large reductions in perplexity are necessary in order to get meaningful word error rate reductions. For instance, a 10 % perplexity reduction from 100 to 90 corresponds to only a 2 % entropy reduction, and conceivably a 2 % word error reduction. This example seems to be in line with Rosenfeld's rule of thumb.

Chapter 3

Morfessor

Morfessor is an unsupervised morphology learner and morpheme segmenter, developed as part of the current thesis work. The following presentation of Morfessor is divided into four parts: First, general characteristics of the Morfessor model are described, including differences between Morfessor and other models. Next, the chronological development of Morfessor is outlined, followed by an exact mathematical formulation. The fourth part addresses search algorithms on a fairly general level. The chapter is concluded by a survey of related work.

3.1 General Characteristics of the Model

In this section, some characteristics of the Morfessor model are highlighted. The intention is to position Morfessor theoretically, motivating some of the solutions made in this work.

3.1.1 Full Morph Segmentation

Morfessor learns a *morpheme segmentation* of the word forms in the input data. This is a necessary step towards an Item and Arrangement (IA) or Item and Process (IP) model of morphology (see Section 2.1.3). Current versions of Morfessor do not identify which morphs are allomorphs of the same morpheme and in that sense the result does not correspond to a model of the entire morphology of the language. The newer versions of Morfessor do, however, propose grammatical categories for the morphs discovered. Each morph is assigned a probability of its tendency to function as a stem, prefix, or suffix. Such information seems useful when one needs to distinguish between semantically rich morphs (stems) and grammatical functions (affixes). The stems are more important, e.g., in information retrieval.

Morfessor produces a *full segmentation* of the word forms in the data. This is an important objective when dealing with languages in which new word forms are cre-

ated productively not only through inflection, but also through derivation and compounding. Many morphology induction methods focus on the inflections only (e.g., Goldsmith, 2001), and they typically separate only the last suffix from the rest of the word form. The English word "dessertspoonfuls" would thus be segmented as 'dessertspoonful+s" by such methods, whereas Morfessor would go for the full segmentation "dessert+spoon+ful+s". To produce a full segmentation is computationally much harder, because the number of segments per word cannot be limited to only two.

3.1.2 Morph-Based Syntax

Morfessor does not induce inflectional paradigms (see Section 2.1.3, page 14). This means that it does not make precise statements about which affixes can be attached to a particular stem. For example, if the word forms "spoon, spoon+s, spoon+ful" are attested in the training corpus, also other forms will have some probability in the model. Such forms are "spoon+ness, spoon+ing+ly, spoon+est", which do not not belong to the standard inflectional paradigm of "spoon". Ideally the probabilities reflect how likely these imaginative words are to occur in real language use. The approach implies that there is no clear-cut border between morphology and syntax; words as well as longer phrases are formed by concatenating morphemes and the number of possible words and phrases is immense. (Yet most of the possible words and phrases have very low probability.) A well estimated model mimics the actual language use as it manifests itself in the data. That is, it is a model of language *performance* in Chomskyan terms (Chomsky, 1965).

If one does prefer to describe word forming using inflectional paradigms, the distinction between morphology and syntax is clearer. First one determines an exhaustive list of the existing inflectional forms (morphology) and then one determines how the word forms can be combined into phrases (syntax). There is a risk, however, that the resulting paradigms are not only logical and complete, but in fact *too* complete. For instance, if one assumes that Finnish nominals are inflected in two numbers and fifteen cases (which is a customary view), one might want to fill each slot in the paradigm with a word form. Following this procedure, one can form words such as "hiin" and "iin", which logically are the instructive plural forms of "hiki" (sweat) and "ikä" (age). However, it is questionable whether these forms would ever be used and understood in a natural situation of language use. Thus, we could say that we have a model of language *competence* in Chomskyan terms, i.e., a theoretical construct. By supposing this structure one may actually have a model that reserves modeling capacity for extremely improbable words, whereas it misses rather probable words that do not fit elegantly into the paradigm. An example of the latter is the frequently used double partitive "montaa" ([not] many), which is the partitive of "monta", which is already the partitive of "moni". (The word "montaa" is not recognized by the Finnish TWOL-analyzer, in contrast to "hiin" and "iin".)

3.1.3 Bayesian and MDL Framework for Model Formulation

Morfessor learns a lexicon of morphs from the corpus it is trained on. The lexicon consists of a set of morphs, such that every word form in the corpus can be produced by concatenating some morphs present in the lexicon. The earliest version of the Morfessor model was formulated using the MDL formalism (Publications 1, 6, and 7). Later versions incorporate general linguistic constraints encoded as Bayesian priors, and thus a Bayesian framework has been more appropriate for these model variants (Publications 2, 4, and 5). The crucial aspect of all of these models is that they attempt to find a balance between modeling accuracy and model complexity (cf. Section 2.2.3). That is, the resulting lexicon should cover the words in the training corpus well, but additionally generalize to new word forms not observed in the training data.

3.1.4 Meaning and Form of Morphs

The morph lexicon stores information about every morph it contains. Each morph has a *form*, which is here assumed to be a string of letters, e.g., "s p o o n". If we were dealing with speech instead of text, the form would consist of a string of phonemes ([s p u: n]) rather than a string of letters.

Additionally, every morph has its own role in the language, i.e., a *meaning* or syntactic function. In order for a system (biological or artificial) to produce or understand language successfully, it conceivably needs to store this morph meaning (or whatever one prefers to call it) in the lexicon alongside the form of the morph.

Morfessor is trapped in a text world. From a corpus of text alone, the program has to infer how language works. Therefore, any notion of meaning that is accessible to Morfessor is based solely on structures manifested in text. This is compatible with the structuralist view that the meaning of words (and morphemes) is reflected directly in how they are used. The idea is captured nicely in a classical phrase coined by J. R. Firth in the 1950's: "You shall know a word by the company it keeps" (Firth, 1957). Thus, the meaning of a morph "is" in a sense the distribution of possible contexts of other morphs the morph can occur in.

By parameterizing the contextual distribution, or typical *usage*, of a morph and storing the parameters in the lexicon, groups of morphs that have similar behavior can be detected. In this work, the parameterized usage of each morph is utilized for determining the likely grammatical category of the morph, namely the tendency of the morph to function as a stem, prefix, or suffix, which affects the probability that the morph is observed in different positions within words. For instance, the last morph of a word cannot be a prefix nor the first morph a suffix.

In Publication 4, the usage-based features stored in the lexicon for each morph are indeed called "meaning". However, due to the crude nature of the features utilized in the current model, this term has later been replaced by the more neutral term "usage", which is is employed in Publication 5 and the introductory part of the thesis.

3.1.5 Frequent Words and Hierarchical Representation of Morphs

From a theoretical point of view, all words can be split into their constituent morphs. For instance, the English word "suddenly" is logically composed of "sudden+ly". By recognizing this inner word structure, other words can be formed by analogy, e.g., "abrupt+ly, sudden+ness". This generalization capacity is crucial when dealing with *sparse* data, e.g., corpora in which most words are observed in only a few of all their possible forms. Typically, language data are *always* sparse.

However, "suddenly" is a very frequent word form. From a modeling perspective it could be economical to store it as its own entry in the lexicon. This would provide faster access to the word, since it would not have to be composed of parts. Additionally, this would allow "suddenly" to have a meaning that is not entirely the composition of the meanings of "sudden" and "-ly". Moreover, word frequency seems to play an important role in human language processing. Baayen and Schreuder (2000) refer to numerous psycholinguistic studies that report that high-frequency words are responded to more quickly and accurately than low-frequency words in various experimental tasks. This effect is obtained regardless whether the words have compositional structure or not.[1]

The latest version of Morfessor proposes a solution to this dilemma (Publication 4). The morph lexicon is *hierarchical*, such that each morph in the lexicon consists either of a string of letters or of two submorphs, which are themselves present in the lexicon. The submorphs can in turn recursively consist of shorter submorphs. Figure 3.1 shows the hierarchical representations obtained for the Finnish word "oppositiokansanedustaja" ("member of parliament of the opposition") and the English word "straightforwardness". The model utilizes information about word frequency: The English word has been frequent enough in the corpus to be included in the lexicon as an entry of its own. The Finnish word has been less frequent and is split into "oppositio" ("opposition") and 'kansanedustaja' ("member of parliament"), which are two separate entries in the lexicon induced from the Finnish corpus. Frequent words and word segments can thus be accessed directly, which is economical and fast. At the same time, the inner structure of the words is retained in the lexicon, because the morphs are represented as the concatenation of other (sub)morphs.

Additionally, every morph is tagged with a category, which is the most likely category for that morph in that context: prefix (PRE), stem (STM), or suffix (SUF). Not all morphs in the lexicon need to be "morpheme-like" in the sense that they carry meaning. Some morphs correspond more closely to syllables and other short fragments of words. The existence of these *non-morphemes* (NON) makes it possible to represent some longer morphs more economically, e.g., the Finnish "oppositio" consists of "op" and "positio" (position), where "op" has been tagged as a non-morpheme and "positio" as a stem (see also Section 3.2.4).

One might draw a parallel from the non-morphemes in the Categories-MAP model to findings within psycholinguistic research. McKinnon et al. (2003) suggest that morphological decomposition and representation extend to non-productive morphemes, such as

[1] Note, however, that these findings may not apply to all linguistic tasks. When test persons were exposed to word forms that were ungrammatical in context, high-frequency regular word forms seemed to be processed as if they were compositional rather than unanalyzed wholes (Allen et al., 2003).

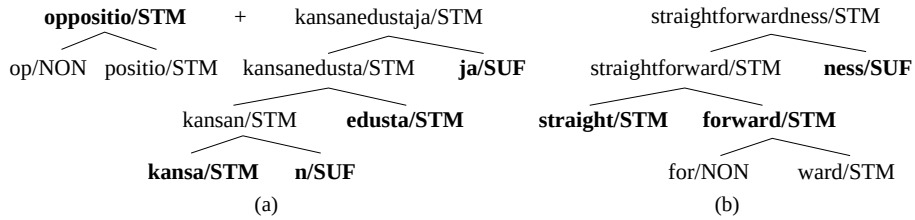


Figure 3.1: The hierarchical segmentations of (a) the Finnish word "oppositiokansanedustaja" ("MP of the opposition") and (b) the English word "straightforwardness" (obtained by the Categories-MAP model; see Section 3.1.5 for details). Additionally, every morph is tagged with a category, namely the most likely category for that morph in that context.

"-ceive, -mit", and "-cede" in English words, e.g., "conceive, permit, recede".

The idea of a hierarchical lexicon *per se* is not novel. Within the field of unsupervised natural language acquisition, de Marcken (1996) proposes a model in which linguistic units recursively consist of shorter linguistic units (see Section 2.1.2). Among other things, de Marcken applies his model in a task of unsupervised word segmentation of a text, where the blanks have been removed. As a result, hierarchical segmentations are obtained, e.g., for the phrase "for the purpose of": [[f[or]][[t[he]][[[p[ur]][[[po]s]e]][of]]]]. The problem here from a practical point of view is that there is no way of determining which level of segmentation corresponds best to a conventional word segmentation. On the coarsest level the phrase works as an independent "word" ("forthepurposeof"). On the most detailed level the phrase is shattered into individual letters.

Where de Marcken has no means of knowing which level of segmentation is the desired one, the Morfessor model provides a way out. In a task of morpheme segmentation, the hierarchical representation is expanded to the *finest resolution that does not contain non-morphemes*. In Figure 3.1 this level has been indicated using a boldface font. The Finnish word is expanded to "oppositio+kansa+n+edusta+ja" (literally "opposition + people + of + represent + -ative"). The English word is expanded into "straight+forward+ness". The morph "forward" is not expanded into "for+ward" (although this might have been appropriate), because "for" is tagged as a non-morpheme in the current context.

3.2 Development Steps of the Morfessor Model

Among the primary concerns in the development of the Morfessor model has been to figure out how to obtain a full morph segmentation while avoiding *over-* and *undersegmentation* as well as *morphotactic violations*. For instance, the correct full segmentation of the English word "swineherds" is "swine+herd+s". In oversegmentation, or excessive segmentation, words are split into too many parts (e.g., "s+win+e+her+d+s"). Undersegmentation, or incomplete segmentation, implies that some morpheme boundaries are missed (e.g., "swine+herds"). Morphotactic violations arise when a segment that can

function as a morph in some context is proposed in the wrong context. For example, "s" is a frequent English morph, but it is a suffix and should not be used in word-initial position. In "s+wine+herd+s", "swine" has been split into two morphs that do in fact exist, but the suggested combination is morphotactically incorrect.

The Morfessor model has undergone four development steps, which will be described briefly in the light of the three demonstrated difficulties. For a more thorough treatment the reader is referred to the attached publications. The Morfessor *Baseline* model was originally called the "Recursive MDL" algorithm (Publication 1)[2]. The following steps consist of Morfessor *Baseline-Freq-Length* (Publication 2), Morfessor *Categories-ML* (Publication 3), and Morfessor *Categories-MAP* (Publication 4). Mathematically, these four steps can be combined into a *unifying framework*, which is presented in Section 3.3 and Publication 5.

3.2.1 Morfessor Baseline

The Morfessor Baseline model (Publication 1) learns a lexicon of morphs, which is concise and which produces a compact representation for the words in the corpus used as data. The lexicon is flat, that is, morphs are simply strings of letters and do not have substructure. As a matter of fact, this model is very similar to the simple fruit words example in Section 2.2.3. But unlike the fruit words model, Morfessor Baseline does not assume a uniform probability distributions for the proposed morphs. Instead, the probability of a morph is its relative frequency among all the morphs in the suggested segmentation of the corpus. For instance, the segmentation obtained using *Lexicon 2* (Section 2.2.3) consists of 21 morphs: "`apple # orange # lemon # juice # apple juice # orange juice # apple tree # lemon tree # #`". The probability of the morph "apple", which occurs three times, would thus be $\frac{3}{21} = \frac{1}{7}$.

The Baseline model was inspired by the MDL criterion (Publication 1). In later work, the mathematical formulation has been refined. In Publication 7, an MDL formulation is given, which corresponds to two-part-code "crude" MDL according to the terminology of Grünwald et al. (2005) (page 27). Creutz and Lagus (2005) present an equivalent MAP formulation of the Baseline.

Due to its simplistic nature, Morfessor Baseline is prone to make errors of all the three kinds mentioned. A frequent string is most concisely coded in one piece, regardless of its linguistic structure. This sometimes leads to undersegmentation (e.g., "having, soldiers, seemed"). A rare string is, by contrast, best coded in short segments, which can cause oversegmentation (e.g., "vol+can+o"). The model does not assign any grammatical categories to the proposed morphs and it does not constrain the context in which a morph can occur. This lack of restrictions occasionally produces morphotactic violations (e.g., "s+wing, ed+win").

In spite of its inadequacies, the Baseline works surprisingly well. In all speech recognition experiments, the Baseline model has been used instead of the more sophisticated versions. In general, as shown in Creutz and Lagus (2005), Morfessor Baseline pro-

[2]Publication 1 also introduces a second model, called "Sequential ML". The Sequential ML model has not been developed further and is not described in the current presentation.

duces a better morph segmentation from a morphological point of view, if the data set used is a word list rather than a full corpus. In a word list every distinct word form occurs only once, which reduces the dominance of the frequent word forms that make up a considerable part of any corpus of running text. We will return to the performance of the different variants of Morfessor in Chapter 4.

3.2.2 Morfessor Baseline-Freq-Length

The Morfessor Baseline-Freq-Length model (Publication 2) extends the Baseline model by applying Bayesian prior probabilities to the *frequency* and *length distributions* of the morphs.

The prior distribution for morph frequency is derived from the well-known *Zipf's law* (Zipf, 1935). This law addresses a general tendency in the distribution of *token frequencies* for a set of *types*. A set of types is a set of distinct, unique items (e.g., morphs in the morph lexicon), whereas the same item can occur many times in a set of tokens (e.g., morphs in a corpus, where the words have been split at the morpheme boundaries). According to Zipf's law the frequency of an item is inversely proportional to its rank. This means that there are a very small number of extremely frequent items (e.g., the English morphs "and, of, -ing") and a large mass of very rare items (e.g., "cloot, gallet, ruckus, Waunugur"). The purpose of the frequency prior in the Baseline-Freq-Length model is to favor solutions where the frequency distribution of the proposed morphs is in accordance with Zipf's law.

A morph length distribution describes the proportion of morphs of a particular length, measured in letters. Figure 3.2 shows distributions corresponding to the lengths of the morphs of a *linguistic* morpheme segmentation. (The linguistic segmentation is that of the Hutmegs Gold Standard described in Section 4.1.2 and Publication 8.) Fig. 3.2 further shows that both for Finnish and English the desired linguistic length distribution can be approximated rather accurately by a *gamma distribution*. This state of affairs is exploited in the the Baseline-Freq-Length method, which utilizes a gamma distribution as a prior for morph length.[3]

Due to the priors for morph length and frequency, the Baseline-Freq-Length method does outperform the plain Baseline method. The number of over- and undersegmented words is reduced, but the model structure is still insufficient for preventing morphotactic violations. Furthermore, the difference between Morfessor Baseline and Morfessor Baseline-Freq-Length diminishes with larger amounts of data.

According to the experiments performed, the length prior is more effective than the frequency prior. This may be due to the fact that the Zipfian frequency distribution is such a "natural" phenomenon that it is not necessary to model it explicitly. In fact, randomly generated texts have been shown to exhibit Zipf's-law-like frequency distributions (Li, 1992). Due to the ineffectiveness of the frequency prior, it has been omitted in most ex-

[3]Note that we model the lengths of the morphs in the lexicon, i.e., a set of morph *types*. In order to model the length distribution of a set of *tokens*, one might prefer a *Poisson distribution*, as does Nagata (1997) when dealing with words of Japanese. A Poisson distribution is also applied in the so-called Sequential ML method of Publication 1, where morph tokens are concerned.

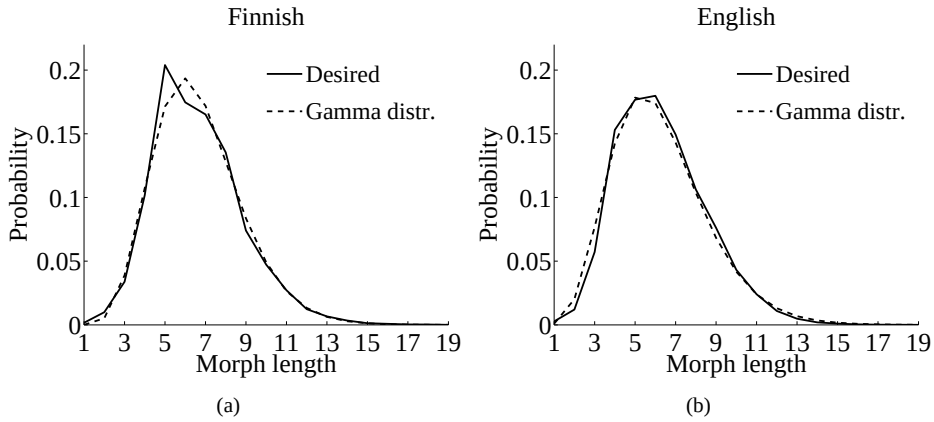


Figure 3.2: Length distributions for the morph types in Finnish and English lexicons (for data sets comprising 250 000 words). The solid line corresponds to the empirical distribution of morph lengths in the linguistic gold standard segmentation of the words, i.e., the desired result. The dashed line represents a gamma distribution fitted to the linguistic distribution.

periments with Morfessor. The corresponding model configuration is called Morfessor *Baseline-Length*. It is important to note that even if a property is not modeled through an *explicit* prior, one can always analyze how it is manifested *implicitly* in the model. A more thorough treatment of explicit and implicit length and frequency priors can be found in Section 3.3.5 and in Appendices A.1–A.3.

3.2.3 Morfessor Categories-ML

The Categories-ML model (Publication 3) introduces a simple morphotactics that reduces errors caused by the context-insensitivity of the Baseline models. In Categories-ML, a segmentation produced by one of the Baseline algorithms is reanalyzed using maximum likelihood (ML) optimization in combination with some heuristics.

Each morph in the segmented corpus is tagged with one of the following categories: prefix, stem, or suffix. The tendency of a morph to function as one of these categories is determined from a few usage-based features of the morph. (See Section 3.3.5, pages 50–54, for details.) In cases where none of the three "proper" categories is likely, morphs are tagged with an additional "noise" (or non-morpheme) category. Noise morphs are short segments, which are not morphemes at all (e.g., "vol" and "o" in "vol+can+o") or not morphemes in the current context (e.g., "can" in "vol+can+o"). Consequently, the presence of noise morphs typically indicates that a word has been oversegmented or that it contains morphological violations.

Oversegmentation is reduced by applying a heuristic that joins together noise morphs with their neighbors. This creates longer segments that are likely stems (e.g., "vol+can+o" becomes "volcano"). The opposite problem, undersegmentation, is alleviated by forcing splits of redundant morphs, with certain limitations. A morph is

redundant when it consists of other morphs that are contained in the lexicon (e.g., "seemed = seem+ed"). By removing the redundant morph the lexicon is made smaller. In Categories-ML, the size of the lexicon is controlled through these heuristics instead of an overall probability function. Once the lexicon has been modified, maximum likelihood reestimation is applied in order to resegment and retag the corpus.

A first-order Hidden Markov Model (HMM) is used for assigning probabilities to each possible segmentation and tagging of a word form. That is, there are *transition probabilities* between morph categories, e.g., the probability of observing a stem after a prefix. In addition, *emission probabilities* indicate how likely a morph is to occur when a category is given (e.g., the probability of suggesting the morph "ed" as a suffix).

The HMM is intended to model a morphotactics that is captured by the following regular expression:

$$\texttt{word = (prefix* stem suffix*)+} \tag{3.1}$$

This word structure is flexible enough to cope with extensive compounding and many consecutive affixes. A word can consist of any number of stems, each optionally preceded by prefixes and followed by suffixes.[4] No assumptions are made regarding whether the language is more likely to employ prefixation or suffixation. Nonetheless, there are important restrictions: A suffix may not start a word, or a prefix end it. Moreover, a prefix should not be followed directly by a suffix. Therefore, many of the morphotactic violations observed with the Baseline models are removed by Categories-ML. For instance, no suffixes are suggested word-initially in "swing" and "Edwin", in contrast to the Baseline segmentations "s+wing" and "ed+win".

3.2.4 Morfessor Categories-MAP

The Categories-MAP model (Publication 4) has a more sophisticated formulation than Categories-ML, in two respects: (1) Categories-MAP operates on data sets consisting of word tokens, whereas Categories-ML operates on word types. If one wants to draw parallels to language processing in humans, the former approach is more desirable, because knowing the frequency of a word is valuable information; see Section 3.1.5. (2) Contrary to Categories-ML, Categories-MAP is a complete maximum a posteriori model, which means that it does not need to rely on heuristics in order to determine the optimal size of the lexicon.

The improvements over the Categories-ML model have been made possible by introducing a *hierarchical lexicon structure*. The hierarchical lexicon has already been described in Section 3.1.5. The hierarchical structure provides different mechanisms for preventing over- and undersegmentation than the heuristics used in Categories-ML. In a morpheme segmentation task, undersegmentation can be avoided by expanding a lexical item into the submorphs it consists of. In order not to create the opposite problem, oversegmentation, the substructures are only expanded as long as they do not contain non-morphemes, i.e., noise morphs.

[4] Even in a fairly isolating language like English, words can consist of several morphemes (e.g., "un+interrupt+ed+ly"). Although affixes in English generally occur before the first stem of the word (as prefixes) or after the last stem (as suffixes), there are examples of affixes located *between* stems within a word, e.g., "coast+guard+*s*+man, clean+*ing*+lady, new+*ly*+wed+s".

The morphotactics is essentially the same in both Categories models. However, in Categories-MAP the non-morphemes are not removed from the model, even though they do not show on the "surface" in a morpheme segmentation. The existence of non-morphemes sometimes even helps against the oversegmentation of rare words. When for instance, a new name must be memorized, it can be constructed "economically" from two shorter familiar fragments without breaking it down into individual letters. In this way, the name can be coded in the lexicon at a lower "cost". For example, in one of the English experiments the name "Zubovski" occurred twice in the corpus and was added to the morph lexicon as consisting of the stem "zubov" followed by the non-morpheme "ski".

3.3 Unified Mathematical Formulation

This section presents a unified mathematical formulation of Morfessor (Publication 5), which incorporates every development step of the model. The unified framework rather closely corresponds to the latest and most sophisticated model variant, Morfessor Categories-MAP, of which the earlier variants can be seen as simplifications.

3.3.1 Maximum a posteriori estimate of the overall probability

The task is to induce a *model of language* in an unsupervised manner from a corpus of raw text. The model of language ($\mathcal{M}$) consists of a *morph lexicon*, which is an inventory of morphs, and a *grammar*, which determines how morphs can be combined into words. We aim to find the optimal model of language for producing a segmentation of the corpus. This is achieved through maximum a posteriori (MAP) optimization. The MAP estimate to be maximized, $P(\mathcal{M} \mid \mathit{corpus})$, consists of two parts: the probability of the model of language $P(\mathcal{M})$ and the likelihood of the corpus conditioned on the given model of language, written as $P(\mathit{corpus} \mid \mathcal{M})$:

$$\arg\max_{\mathcal{M}} P(\mathcal{M} \mid \mathit{corpus}) = \arg\max_{\mathcal{M}} P(\mathit{corpus} \mid \mathcal{M}) \cdot P(\mathcal{M}), \text{ where} \quad (3.2)$$

$$P(\mathcal{M}) = P(\mathit{lexicon}, \mathit{grammar}). \quad (3.3)$$

The probability of the model of language (Eq. 3.3) is the joint probability of the probability of the induced lexicon and grammar. Note that in the Categories-ML variant this part is replaced by a heuristic procedure (see Publication 3).

The representation of the lexicon, grammar and corpus will be described next, as well as the components of each.

3.3.2 Lexicon

The lexicon contains one entry for each distinct morph (morph type) in the segmented corpus. Suppose that the lexicon consists of M distinct morphs. The probability of

coming up with a particular set of M morphs $\mu_1 \dots \mu_M$ making up the lexicon can be written as:

$$P(\textit{lexicon}) = P(\textit{size}(\textit{lexicon}) = M) \cdot P(\textit{properties}(\mu_1), \dots, \textit{properties}(\mu_M)) \cdot M!. \tag{3.4}$$

The product contains three factors: (i) the prior probability that the lexicon contains exactly M distinct morphs, (ii) the joint probability that a set of M morphs, each with a particular set of properties, is created, and (iii) the factor $M!$, which is explained by the fact that there are $M!$ possible orderings of a set of M items and the lexicon is the same regardless of the order in which the M morphs emerged. (It is always possible to afterwards rearrange the morphs into an unambiguously defined order, such as alphabetical order.)

The effect of the first factor, $P(\textit{size}(\textit{lexicon}) = M)$, is negligible, since in the computation of a model involving thousands of morphs and their parameters, one single probability value is of no practical significance. Thus, we have omitted to define a prior distribution for $P(\textit{size}(\textit{lexicon}))$.[5]

The properties of a morph can be grouped into information regarding (1) the usage and (2) the form of the morph:

$$P(\textit{properties}(\mu_i)) = P(\textit{usage}(\mu_i), \textit{form}(\mu_i)). \tag{3.5}$$

The exact properties utilized in the Morfessor model are described in Section 3.3.5.

3.3.3 Grammar

The grammar determines how language units can be combined. In Morfessor, the grammar consists of a simple morphotactics, realized as a Hidden Markov Model (HMM). The states of the HMM correspond to grammatical categories. In Categories-MAP and Categories-ML, four categories are used: prefix (PRE), stem (STM), suffix (SUF), and non-morpheme (NON). In the Baseline model variants, there are no category distinctions, which corresponds to having an HMM with one single state.

The structure of the grammar is fixed and not estimated from the data. However, the values of some parameters (the transition probabilities in the HMM) are not fixed. It is convenient to use a uniform (non-informative) prior for these probability values. Since any set of transition probability values is equally likely, $P(\textit{grammar})$ is a constant (say k), and $P(\mathcal{M})$ in Equation 3.3 reduces to $P(\textit{lexicon}) \cdot k$.

It is possible for a morph to be assigned different categories in different contexts. The tendency of a morph μ_i to be assigned a particular category C_i, $P(C_i \,|\, \mu_i)$, (e.g., the probability that the English morph "ness" functions as a suffix) is derived from the usage-related parameters of the morph:

$$P(C_i \,|\, \mu_i) = P(C_i \,|\, \textit{usage}(\mu_i)). \tag{3.6}$$

[5] If one were to define a proper prior, one possible choice would be Rissanen's universal prior for positive integers (see Eq. 3.20).

The inverse probability, i.e., the probability of a particular morph when the category is known, is needed for expressing the probability of the segmentation of the corpus. This *emission probability* $P(\mu_i \,|\, C_i)$ is obtained using Bayes' formula:

$$P(\mu_i \,|\, C_i) = \frac{P(C_i \,|\, \mu_i) \cdot P(\mu_i)}{P(C_i)} = \frac{P(C_i \,|\, \mu_i) \cdot P(\mu_i)}{\sum_{\forall \mu_{i'}} P(C_i \,|\, \mu_{i'}) \cdot P(\mu_{i'})}. \tag{3.7}$$

The category-independent probabilities $P(\mu_i)$ are maximum likelihood estimates, i.e., they are computed as the frequency of the morph μ_i in the corpus divided by the total number of morph tokens. In the Baseline models, where one single category is used, the whole expression in Eq. 3.7 reduces to the category-independent probability $P(\mu_i)$.

3.3.4 Corpus

Every word form in the corpus can be represented as a sequence of some morphs that are present in the lexicon. Usually, there are many possible segmentations of a word. In MAP modeling, the one most probable segmentation is chosen. The probability of the corpus, when a particular model of language (lexicon and grammar) and morph segmentation are given, takes the form:

$$P(\textit{corpus} \,|\, \mathcal{M}) = \prod_{j=1}^{W} \Big[P(C_{j1} \,|\, C_{j0}) \prod_{k=1}^{n_j} \big[P(\mu_{jk} \,|\, C_{jk}) \cdot P(C_{j(k+1)} \,|\, C_{jk}) \big] \Big]. \tag{3.8}$$

As mentioned in the grammar section above, this is a Hidden Markov Model and it is visualized in Figure 3.3. The product is taken over the W words in the corpus (token count), which are each split into n_j morphs. The k^{th} morph in the j^{th} word, μ_{jk}, is assigned a category, C_{jk}. The probability that the morph is emitted by the category is written as $P(\mu_{jk} \,|\, C_{jk})$. There are transition probabilities $P(C_{j(k+1)} \,|\, C_{jk})$ between the categories, where C_{jk} denotes the category assigned to the k^{th} morph in the word, and $C_{j(k+1)}$ denotes the category assigned to the following, or $(k+1)^{\text{th}}$, morph. The transition probabilities comprise transitions from a special word boundary category (#) to the first morph in the word, as well as the transition from the last morph to a word boundary.

3.3.5 Features of Usage and Form of Morphs

Each morph in the lexicon consists of a bunch of features, which can be divided into properties of usage[6] and form. The usage-based properties of a morph include properties of the morph itself and properties of the context it typically appears in. By the form of a morph we understand the symbolic representation of the morph, i.e., the string of letters it consists of.

The usage and form are parameterized and stored in the lexicon. Which parameter values are likely is determined by probability density functions (pdf:s), which are priors,

[6] called "meaning" in Publication 4

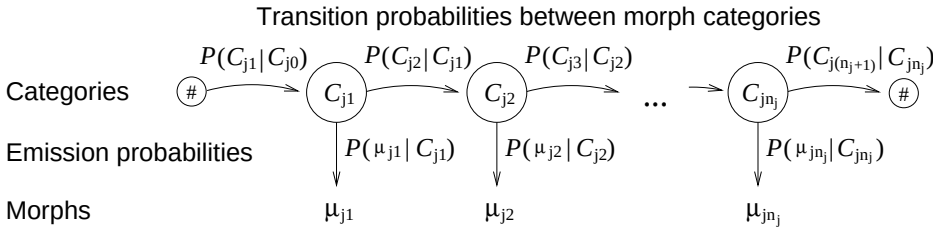


Figure 3.3: The HMM model of a word according to Equation 3.8. The word consists of a sequence of morphs which are emitted from latent categories. For instance, a possible category sequence for the English word "unavailable" would be "prefix + stem + suffix" and the corresponding morphs would be "un + avail + able".

either in the Bayesian sense or in the more coding-oriented MDL sense, that favor linguistically motivated solutions.

Given the distinction between usage and form, we make the assumption that they are statistically independent:

$$P(\textit{properties}(\mu_1), \ldots, \textit{properties}(\mu_M)) = \\ P(\textit{usage}(\mu_1), \ldots, \textit{usage}(\mu_M)) \cdot P(\textit{form}(\mu_1), \ldots, \textit{form}(\mu_M)). \quad (3.9)$$

Form of a Morph

We further make the simplifying assumption that the forms of the morphs in the lexicon are independent of each other, thus:

$$P(\textit{form}(\mu_1), \ldots, \textit{form}(\mu_M)) = \prod_{i=1}^{M} P(\textit{form}(\mu_i)). \quad (3.10)$$

Morphs in the lexicon have hierarchical structure. A morph can either be a flat string of letters or have substructure. The probability of the form of the morph μ_i depends on whether the morph is represented as a string of letters (Eq. 3.11a) or as the concatenation of two submorphs (Eq. 3.11b):

$$P(\textit{form}(\mu_i)) = \\ \begin{cases} (1 - P(\textit{sub})) \cdot \prod_{j=1}^{\textit{length}(\mu_i)} P(c_{ij}). & (3.11a) \\ P(\textit{sub}) \cdot P(C_{i1} \mid \textit{sub}) \cdot P(\mu_{i1} \mid C_{i1}) \cdot P(C_{i2} \mid C_{i1}) \cdot P(\mu_{i2} \mid C_{i2}). & (3.11b) \end{cases}$$

The probability that a morph has substructure, $P(\textit{sub})$, is estimated from the lexicon by dividing the number of morphs having substructure by the total number of morphs. $P(\textit{sub})$ is non-zero only in the Categories-MAP model version.

In (3.11a), $P(c_{ij})$ is the probability of the j^{th} letter in the i^{th} morph in the lexicon. The probability distribution to use for the letters in the alphabet is estimated from the corpus.

Equation 3.11b resembles Equation 3.8, where the probability of the corpus is given. $P(C_{i1} \mid \textit{sub})$ is the probability that the first morph in the substructure is assigned the

category C_{i1}. $P(C_{i2} \mid C_{i1})$ is the transition probability between the categories of the first and second submorphs. $P(\mu_{i1} \mid C_{i1})$ and $P(\mu_{i2} \mid C_{i2})$ are the probabilities of the submorphs μ_{i1} and μ_{i2} conditioned on the categories C_{i1} and C_{i2}. The transition and morph emission probabilities are the same as in the probability of the corpus (Eq. 3.8). Examples of concrete substructures were shown in Figure 3.1 on page 41.

Features Related to the Usage of a Morph

The set of features that could be used for describing usage is very large: The typical set of morphs that occur in the context of the target morph could be stored. Typical syntactic relations of the morph with other morphs could be included. The size of the context could vary from very limited to large and complex. A complex context might reveal different aspects of the usage of the morph, from fine-grained syntactic categories to broader semantic, pragmatic or topical distinctions.

However, currently only a very limited set of features is used, and only based on information contained in word lists. As properties of the morph itself, we count the *frequency* of the morph in the segmented corpus and the *length* in letters of the morph. As "distilled" properties of the context the morph occurs in, we consider the intra-word *right* and *left perplexity* of the morph. (Perplexity measures how predictable the context of a given morph is.)

Using the above features, the probability of the usages of the morphs in the lexicon becomes:

$$P(\textit{usage}(\mu_1), \ldots, \textit{usage}(\mu_M)) = \\ P(\textit{freq}(\mu_1), \ldots, \textit{freq}(\mu_M)) \cdot \prod_{i=1}^{M} \left[P(\textit{length}(\mu_i)) \cdot P(\textit{right-ppl}(\mu_i)) \cdot P(\textit{left-ppl}(\mu_i)) \right]. \tag{3.12}$$

Due to practical considerations in the current implementation, it is assumed that the length, right and left perplexity of a morph are independent of the corresponding values of other morphs. In contrast, the frequencies of the morphs are given as a joint probability, that is, there is one single probability for an entire morph frequency distribution. The probability distributions have been chosen due to their generality and simplicity. In a more sophisticated model formulation, one could attempt to model dependencies between morphs and their features, such as the general tendency of frequent morphs to be rather short.

Next, we describe the individual features and the prior probability distributions that are used for the range of possible values of these features. The treatment of morph usage is concluded by a description of how the usage of a morph translates into category membership probabilities in the current grammar.

Frequency. Frequent and infrequent morphs generally have different semantics. Frequent morphs can be function words and affixes as well as common concepts. The meaning of frequent morphs is often ambiguous as opposed to rare morphs, which are predominantly content words.

The knowledge of the frequency of a morph is required for calculating the value of $P(\mu_i)$ in Equation 3.7. The probability that a particular frequency distribution emerges can be expressed using a *non-informative*, implicit, prior:

$$P(\textit{freq}(\mu_1), \ldots, \textit{freq}(\mu_M)) = 1/\binom{N-1}{M-1} = \frac{(M-1)!(N-M)!}{(N-1)!}, \tag{3.13}$$

where N is the total number of morph tokens in the corpus, which equals the sum of the frequencies of the M morph types that make up the lexicon. The derivation of the formula can be found in Appendix A.1. This probability distribution corresponds to a non-informative prior in the sense that only the total number of morph tokens and types matter, not the individual morph frequencies.

In the Morfessor Baseline-Freq-Length model, an explicit, Bayesian frequency prior is used instead of the non-informative one. It is then assumed that the frequency of one morph is independent of the frequencies of the other morphs. Thus,

$$P(\textit{freq}(\mu_1), \ldots, \textit{freq}(\mu_M)) = \prod_{i=1}^{M} P(\textit{freq}(\mu_i)). \tag{3.14}$$

An expression for $P(\textit{freq}(\mu_i))$ is derived in Publication 2 and it is based on Mandelbrot's correction of Zipf's law. However, that derivation is unnecessarily complicated and incomplete. A better derivation is given in Appendix A.2 and the result is:

$$P(\textit{freq}(\mu_i)) = \textit{freq}(\mu_i)^{\log_2(1-h)} - (\textit{freq}(\mu_i)+1)^{\log_2(1-h)}. \tag{3.15}$$

The parameter h represents the user's prior belief of the proportion of *hapax legomena*, i.e., morph types that occur only once in the corpus. Typically, the proportion of hapax legomena is about half of all morph types.

In practice, the difference between the two alternative frequency priors is small. A mathematical comparison is carried out in Appendix A.3.

Length. In the Morfessor model, it is assumed that the length of a morph affects the probability of whether the morph is likely to be a stem or belong to another morph category. Stems often carry semantic (as opposed to syntactic) information. As the set of stems is very large in a language, stems are not likely to be very short morphs, because they need to be distinguishable from each other.

A simple way to assign lengths to the morphs in the lexicon is to use a special *end-of-morph* character, which is part of the alphabet, and which is appended to each morph string in the lexicon (in Eq. 3.11a). This approach was also applied in the fruit words example in Section 2.2.3. The probability that a morph of a particular length $\textit{length}(\mu_i)$ will emerge in this scheme is:

$$P(\textit{length}(\mu_i)) = [1 - P(\#)]^{\textit{length}(\mu_i)} \cdot P(\#), \tag{3.16}$$

where $P(\#)$ is the probability of the end-of-morph marker. The probability $P(\textit{length}(\mu_i))$ is the result of first choosing $\textit{length}(\mu_i)$ letters other than the end-of-morph marker and

finally the end-of-morph marker. This is an *exponential distribution*, that is, the probability of observing a morph of a particular length decreases exponentially with the length of the morph. Such a prior corresponds poorly with real morph length distributions, examples of which were shown in Figure 3.2 on page 44.

Instead of using an end-of-morph marker, one can first explicitly decide the length of the morph according to a more realistic probability distribution and then choose the selected number of letters to form the morph string. In the Baseline-Freq-Length model, a *gamma distribution* is used (see Fig. 3.2):

$$P(\mathit{length}(\mu_i)) = \frac{1}{\Gamma(\alpha)\beta^\alpha}\mathit{length}(\mu_i)^{\alpha-1}e^{-\mathit{length}(\mu_i)/\beta}, \tag{3.17}$$

where

$$\Gamma(\alpha) = \int_0^\infty z^{\alpha-1}e^{-z}\mathrm{d}z. \tag{3.18}$$

There are two constants, α and β that determine the exact shape of the gamma pdf. The maximum value of the density occurs at $\mathit{length}(\mu_i) = (\alpha-1)\beta$, which corresponds to the most common morph length in the lexicon. The value of β governs the spikiness of the curve, the higher β the flatter and less discriminative is the pdf.

Intra-Word Right and Left Perplexity. The left and right perplexity give a very condensed image of the immediate context a morph typically occurs in. Perplexity serves as a measure for the predictability of the preceding or following morph.

Grammatical affixes mainly carry syntactic information. They are likely to be common "general-purpose" morphs that can be used in connection with a large number of other morphs. We assume that a morph is likely to be a prefix if it is difficult to predict what the following morph is going to be. That is, there are many possible right contexts of the morph and the right perplexity is high. Correspondingly, a morph is likely to be a suffix if it is difficult to predict what the preceding morph can be and the left perplexity is high. The right perplexity of a target morph μ_i is calculated as:

$$\mathit{right\text{-}ppl}(\mu_i) = \Big[\prod_{\nu_j \in \text{ right-of}(\mu_i)} P(\nu_j \mid \mu_i)\Big]^{-\frac{1}{\mathit{freq}(\mu_i)}}. \tag{3.19}$$

There are $\mathit{freq}(\mu_i)$ occurrences of the target morph μ_i in the corpus. The morph tokens ν_j occur to the right of, immediately following, the occurrences of μ_i. The probability distribution $P(\nu_j \mid \mu_i)$ is calculated over all such ν_j. Left perplexity can be computed analogously.

It can be noted that the best results are obtained when only context morphs ν_j that are longer than three letters are included in the perplexity calculation. As becomes clear in the following section, this means that the right and left perplexity are estimates of the predictability of the *stems* that can occur in the context of a target morph. Including shorter morphs seems to make the estimates less reliable, because of the existence of non-morphemes (noise morphs).

As a reasonable probability distribution over the possible values of right and left perplexity, we use *Rissanen's universal prior* for positive integers (Rissanen, 1989):[7]

$$P(n) \approx 2^{-\log_2 c - \log_2 n - \log_2 \log_2 n - \log_2 \log_2 \log_2 n - \ldots}, \tag{3.20}$$

where the sum includes all positive iterates, and c is a constant, about 2.865. To obtain $P(\textit{right-ppl}(\mu_i))$ and $P(\textit{left-ppl}(\mu_i))$, the variable n is substituted by the appropriate value, $\textit{right-ppl}(\mu_i)$ or $\textit{left-ppl}(\mu_i)$. Note that the left and right perplexity play no role in the Baseline models, but are only used in the Categories versions.

Category Membership Probabilities

In the grammar of the Morfessor Categories models, the tendency of a morph to be assigned a particular category (PRE, STM, SUF, or NON) is determined by the usage of the morph (Equation 3.6). The exact relationship,

$$P(C_i \,|\, \textit{usage}(\mu_i)) = P(C_i \,|\, \textit{freq}(\mu_i), \textit{length}(\mu_i), \textit{right-ppl}(\mu_i), \textit{left-ppl}(\mu_i)), \tag{3.21}$$

could probably be learned purely from the data, but currently a fixed scheme is used, involving a few adjustable parameters.

A measure of *prefix-likeness* is obtained by applying a graded threshold realized as a sigmoid function to the right perplexity of the morph (see Figure 3.4a):

$$\textit{prefix-like}(\mu_i) = \big(1 + \exp[-a \cdot (\textit{right-ppl}(\mu_i) - b)]\big)^{-1}. \tag{3.22}$$

The parameter b is the perplexity threshold, which indicates the point where a morph μ_i is as likely to be a prefix as a non-prefix. The parameter a governs the steepness of the sigmoid. The equation for suffix-likeness is identical except that left perplexity is applied instead of right perplexity (Fig. 3.4b).

As for stems, the *stem-likeness* of a morph is assumed to correlate positively with the *length* in letters of the morph. A sigmoid function is employed as above, which yields:

$$\textit{stem-like}(\mu_i) = \big(1 + \exp[-c \cdot (\textit{length}(\mu_i) - d)]\big)^{-1}. \tag{3.23}$$

where d is the length threshold and c governs the steepness of the curve (Fig. 3.4c).

Prefix-, suffix- and stem-likeness assume values between zero and one, but they are not probabilities, since they do not usually sum up to one. A proper probability distribution is obtained by first introducing the *non-morpheme* category, which corresponds to cases where *none* of the proper morph classes is likely. Non-morphemes are typically short, like the affixes, but their right and left perplexity are low, which indicates that they do not occur in a sufficient number of different contexts in order to qualify as a pre- or suffix. The probability that a segment is a non-morpheme (NON) is:

$$P(\text{NON} \,|\, \mu_i) = [1 - \textit{prefix-like}(\mu_i)] \cdot [1 - \textit{suffix-like}(\mu_i)] \cdot [1 - \textit{stem-like}(\mu_i)]. \tag{3.24}$$

[7] Actually Rissanen defines his universal prior over all *non-negative* integers and he would write $P(n-1)$ on the left side of the equation. Since the lowest possible perplexity is one, zero is not included as a possible value in our formula.

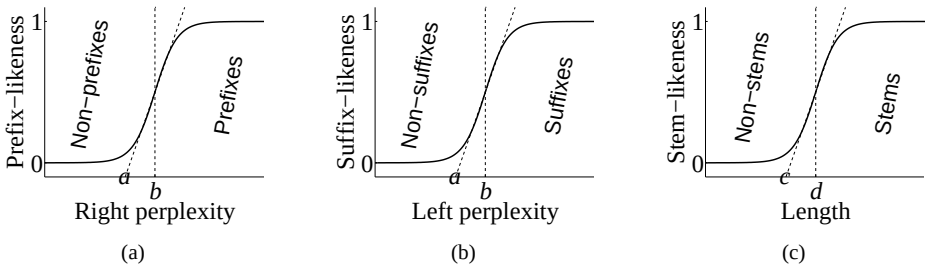


Figure 3.4: Sketch of sigmoids, which express our prior belief of how the right and left perplexity as well as the length of a morph affects its tendency to function as a prefix, suffix, or stem.

Then the remaining probability mass is distributed between prefix, stem and suffix, e.g.:

$$P(\text{PRE} \mid \mu_i) = \frac{\textit{prefix-like}(\mu_i)^q \cdot [1 - P(\text{NON} \mid \mu_i)]}{\textit{prefix-like}(\mu_i)^q + \textit{stem-like}(\mu_i)^q + \textit{suffix-like}(\mu_i)^q}. \tag{3.25}$$

The exponent q affects the normalization. High values of q produce spiky distributions ("winner-take-all effect"), whereas low values produce flatter distributions. The values $q = 1$ and $q = 2$ have been tested, and $q = 2$ turned out to produce better results.

Finally, if the morph consists of submorphs, its category membership probabilities are affected by the category tagging of the submorphs. This prevents conflicts between the syntactic role of a morph itself and its substructure. Details can be found in Appendix A.4.

As mentioned above (page 50), the frequency of a morph could possibly be used for distinguishing between "semantic" morphs (stems) and "grammatical" morphs (affixes). In the current scheme, the frequency *as such* is only used for computing the category-independent probabilities $P(\mu_i)$ (Eq. 3.7). Nonetheless, right and left perplexity are indirect measures of frequency, because a high frequency is a precondition for a high perplexity.

There is a similar idea of using the features frequency, mutual information and left and right entropy[8] in the induction of a Chinese dictionary from an untagged text corpus (Chang et al., 1995). There, the features are applied in classifying character sequences as either words or non-words, which resembles the morpheme categories and the non-morpheme category in the current work. In another work, Feng et al. (2004), a somewhat simpler feature called accessor variety is used in order to discover words in Chinese text. The use of such features within the field of word segmentation dates back to Harris (1955) and Hafer and Weiss (1974). However, in Morfessor, perplexity is not utilized to discover potential morph boundaries, but to assign potential grammatical categories to suggested morphs.

[8] Entropy equals the two-base logarithm ($\log_2$) of perplexity.

3.4 Search Algorithms

Morfessor is a generative model: A given observed corpus is assumed to have been generated from an underlying morph lexicon and grammar. The mathematical formulation provides a means for computing an overall probability for every possible lexicon and every possible segmentation of the corpus. The goal is to find the configuration that yields the highest probability. *Search algorithms*, or *search heuristics*, are used for testing different configurations. Since the *search space* (the set of different configurations) is extremely large, only a very small proportion of all possible configurations can be tested. A good search algorithm "navigates wisely" in the search space and finds promising "paths" that lead to a nearly optimal solution. Finding *the* optimal solution is virtually impossible, due to the size and complexity of the search space.

In Morfessor, the most difficult search problem is to come up with a good set of morphs that will constitute the lexicon. When the set of morphs is known, it is rather straightforward to refine the necessary probability distributions (transition and emission probabilities) and to obtain the most likely segmentation and tagging of the corpus. The probabilities and the segmentation are refined using the Expectation-Maximization algorithm, which is used in all versions of the Morfessor model. The versions do, however, differ with respect to the algorithms they use for optimizing the set of morphs in the lexicon. In the following, the search strategies employed will be touched upon briefly. Further details can be found in the original publications.

3.4.1 Optimizing the Lexicon

The morph lexicon is modified through *greedy* search. At each point in time, a particular morph segmentation of the words in the corpus is assumed. In the Categories model variants, each morph is additionally tagged with a category. During the search, modifications to the existing segmentation and tagging are suggested. The greediness of the search means that the modification that yields the highest overall posterior probability is always selected.

When modifying the segmentation and tagging of the corpus, new morphs can emerge and old morphs disappear. Additionally, the properties of a morph can change (i.e., its frequency, right and left perplexity). Thus, the explicit modification of the representation of the corpus implies an implicit and simultaneous modification of the lexicon.

In the Baseline versions of Morfessor, the search is initialized by representing the corpus as it is, as a set of unsplit words. This corresponds to having a lexicon that includes each unique word form occurring in the corpus. Next, all word forms are shuffled at random, and for each word, every split into two parts is tested. The most probable split location (or no split) is selected and in case of a split, the two parts are recursively split in two. All words are iteratively reprocessed until the probability of the model converges. Figure 3.5 illustrates the hypothetical splitting trees of the two English words "reopened" and "openminded". The morphs into which these words have been split are found in the leaf nodes of the trees ("re+open+ed, open+mind+ed"). Note that the lexicon of Morfessor Baseline does not incorporate hierarchical structures, in contrast

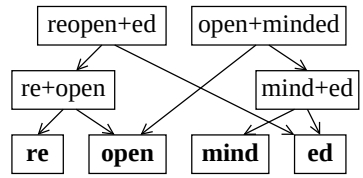


Figure 3.5. Hypothetical splitting trees for two English words, as proposed by the recursive splitting search algorithm used in the Morfessor Baseline model variants.

to Categories-MAP. Thus, only the leaf nodes correspond to morphs that are contained in the lexicon. The higher levels are used solely in the search.

The recursive splitting algorithm used in the Morfessor Baselines is more thoroughly described in Publications 1 and 7. This algorithm is not only used in the Baseline variants of Morfessor, but it additionally serves as an initialization (or bootstrapping) phase for the Morfessor Categories models.

In Categories-ML, the segments proposed by the Baseline splitting are first tagged with the most likely morph category. Then, heuristics for optimizing the lexicon are applied. Short morphs tagged as non-morphemes ("noise morphs") are removed by joining them together with neighboring morphs. Long morphs consisting of other existing morphs are split into their most probable constituents, with the limitation that splitting into noise morphs is prohibited. For details on these procedures, consult Publication 3.

In Categories-MAP, the Baseline segmentation is modified through alternating phases of resplitting and rejoining of morphs. When two morphs μ_1 and μ_2 are joined, the alternative of concatenating them into a longer morph without substructure is weighed against the alternative of adding hierarchical structure, i.e., a higher level morph consisting of μ_1 and μ_2. A more detailed description of the search algorithm is given in Publication 4.

In general, it is computationally more demanding to optimize a context-sensitive model (such as the Categories models) than a context-insensitive model (such as the Baseline models). For a context-insensitive model, the optimal segmentation of a particular string is the same for all occurrences of that string in the corpus, regardless of context. For instance, the substring "lily" in the adverb "friendlily" and the noun "Lily" will obtain the same segmentation in both cases ("li+ly" or "lily").[9] In a context-sensitive model, due to transition probabilities between segments, the optimal representation of a string is not universal. Thus, different occurrences of the same string may obtain different segmentations and category taggings. The string "lily" between two word boundaries (as in "Lily") may be tagged as a stem, whereas "lily" between a stem and a word boundary (as in "friendlily") may be split into two suffixes: "li+ly".

In a context-sensitive model, in principle one should test modifications to the representation of a particular string separately for every context type the string occurs in. However, this approach has some drawbacks in addition to being very slow. The more fine-grained the category distinctions, the higher the number of unique contexts becomes. This number does not need to be very high before the observations of a particular string is "shattered" onto many contexts, with a low number of observations of the string in

[9] Of course, this holds only as long as the segment "lily" has been identified in the word "friendlily". The situation is different if the word has been segmented differently in the first stage, e.g., "friendli+ly".

each particular context. A low number of observations means that the greedy search will have great difficulties in finding modifications that lead to an overall higher probability, because the "weight" of each tested modification is not high enough to support the addition or deletion of morphs from the lexicon. In order to reduce the problems of too fine-grained context distinctions, the contexts can be clustered, or grouped, into a smaller number of coarser context types, where each cluster contains contexts that resemble each other. Simple grouping schemes have been applied in the search algorithms of both Morfessor Categories models.

3.4.2 Optimizing the Segmentation of the Corpus

As described above, the optimization of the lexicon takes place through fairly *local* modifications of the segmentation of the corpus. In an attempt to avoid local maxima of the overall posterior probability, at times a *global* resegmentation of the words in the corpus and a reestimation of the probabilities take place. In the Categories models, the category tagging of the segments is additionally modified.

The global reestimation is performed using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). During the EM phase, the set of morphs in the lexicon remains fixed, but the properties of a morph can change (i.e., frequency, right and left perplexity). The set of morphs can change only if some morph is not used at all in the new segmentation of the corpus, which results in that morph being dropped from the lexicon. In the implementation of the EM algorithm, the Viterbi approximation has been applied (Viterbi, 1967). That is, the one most probable segmentation and tagging has been selected at each point.

3.4.3 Discussion of the Search Algorithms

Admittedly, the focus of the current work does not lie on the perfection of the design of search algorithms. Yet, the model formulation together with the search algorithm determines the quality of the outcome of the learning task. Some directions for possible sophistication are outlined here:

In the EM optimization of the segmentation of the corpus, the Viterbi algorithm (Viterbi, 1967; Forney, 1973) can be replaced by the Forward-Backward algorithm (Baum et al., 1970). In the Forward-Backward algorithm, all possible paths through the states of the HMM are calculated and weighted according to their probability, instead of choosing the single best path as in the Viterbi algorithm. (A path here corresponds to a particular morph segmentation and category tagging.) Promising results have already been obtained with respect to the Categories-ML algorithm (Varis, 2004).

As far as the lexicon optimization is concerned, a corresponding model averaging technique could conceivably be applied. Rather than choosing one optimal lexicon configuration, several alternative configurations could be weighted according to their prior probabilities. Such an approach would most likely be extremely computationally expensive, since each possible configuration would typically contain tens of thousands of morphs. A more promising avenue from a practical point of view could be the design of

less greedy search strategies. For instance, in *simulated annealing*, modifications that lower the overall posterior probability are sometimes accepted, which may reduce the risk of the search getting stuck in local optima. Descriptions of simulated annealing and other iterative improvement algorithms can be found in general machine learning literature; see, e.g., Russell and Norvig (1995); Haykin (1999).

3.5 Related Work

Unsupervised morphology induction is closely connected with the field of automatic word segmentation, i.e., the segmentation of text without blanks into words (or sometimes morphemes). Many languages, such as Chinese and Japanese, employ writing systems where word boundaries are not explicitly marked in text. The discovery of word boundaries, i.e., word segmentation, is thus the first necessary step for any natural language processing task dealing with written text. For some languages, e.g., Chinese, each word corresponds to one morpheme. In this case, word segmentation and morpheme segmentation amount to the same thing.

In the following, alternative approaches to segmentation and morphology learning will be discussed. The existing algorithms in these fields include examples from both the supervised and unsupervised machine learning paradigms. Here, the focus is on unsupervised and minimally supervised methods. For a broader overview, which includes work on supervised algorithms, the interested reader is referred to, e.g., Goldsmith (2001); Kit et al. (2002).

3.5.1 Learning a Segmentation

Recent word and morpheme segmentation methods are typically based on statistical modeling. Alternatively or in addition, they may employ some extension of Harris' segmentation method (Section 2.1.4). The induction of finite-state automata (FSA) constitutes a third approach, and additionally, some researchers have explored the use of artificial neural networks.

In maximum likelihood (ML) modeling, only the accuracy of the representation of the data is considered when choosing a model. In order not to overlearn, some restrictive model search heuristics or model smoothing is required. Segmentation algorithms based on ML, where the complexity of the model is controlled heuristically, include work by Ge et al. (1999), Peng and Schuurmans (2001), and Kneissler and Klakow (2001).

Probabilistic maximum a posteriori (MAP) models and equivalently models based on the Minimum Description Length (MDL) principle choose the best model by simultaneously considering model accuracy and model complexity. A number of segmentation algorithms have been formulated either using MDL or MAP, e.g., Redlich (1993); de Marcken (1996); Brent and Cartwright (1996); Deligne and Bimbot (1997); Brent (1999); Kit and Wilks (1999); Yu (2000); Goldsmith (2001); Snover and Brent (2001); Argamon et al. (2004). In these works, the goal is to find the most likely lexicon (model) as well as a likely segmentation of the data.

The strive for conciseness can also by achieved within finite-state models. There exist algorithms that try to learn finite state automata that compactly model the word forms observed in the training data (Johnson and Martin, 2003; Goldsmith and Hu, 2004). Also Altun and Johnson (2001) induce a stochastic finite-state automaton describing Turkish morphology, but their method works only in a supervised learning task, that is, they require a segmented, labeled corpus to begin with. Another supervised FSA learner, applied on Finnish morphology, is presented by Carlson (2005).

Parallels from the automaton approach can be drawn to Harris-like methods, where a word or morpheme boundary is suggested at locations where the predictability of the next letter in a letter sequence is low (e.g., Déjean, 1998; Ando and Lee, 2000; Adda-Decker, 2003; Feng et al., 2004). If the letter sequences (words or sentences) are sorted into a suffix tree, these "low-predictability locations" correspond to nodes with a high branching factor. The suffix tree could be compressed by merging nodes that have identical continuations, thereby producing a more compact data structure, which is an FSA.

Researchers interested in human language acquisition have applied recurrent neural networks to the segmentation of utterances. Like Harris, Elman (1990) assumes word boundaries at locations, where the network cannot easily predict the next letter in the sequence. Christiansen et al. (1998) utilize segmentation cues obtained from speech, e.g., stress.

Commonly, algorithms designed for word segmentation utilize very little prior knowledge or assumptions about the syntax of the language. Instead, prior knowledge about typical word length may be applied, and small seed lexicons are sometimes used for bootstrapping. The segmentation algorithms try to identify character sequences that are likely words without consideration of the context in which the words occur.

3.5.2 Learning Morphological Structure

The Morfessor model provides a good means for the *segmentation* of words into morphemes. Alternatively, the model can be applied to word form *generation*. The rather few restrictions incorporated in the current model makes it a very permissive model of morphology. Such a model predicts a large number of words outside of the observed training corpus. This is desirable behavior, since a successful learning algorithm should be able to generalize to unseen data. However, a permissive model also makes many mistakes. Many alternative approaches to morphology learning focus on the acquisition of more restrictive morphologies, where much fewer words outside of the training corpus are recognized.

Some works discover pairs of related words or pairs of multiword collocations. Jacquemin (1997) discovers morphological variants of multiword collocations, e.g., "longitudinal record*ing*" vs. "longitudinal*ly* record*ed*". The collocations essentially have the same semantics and can be identified through regular suffix patterns, e.g., {(∅, ing), (ly, ed)}. Baroni et al. (2002) and Neuvel and Fulop (2002) propose algorithms that learn similarities in the spelling of word pairs. The discovery of patterns is not restricted to concatenation, but also include, e.g., vowel change such as the German Umlaut: "An-

schlag" vs. "Anschläge". Generation takes place by predicting missing word pairs. For instance, the pair "receive" vs. "reception" yields the pair "deceive" vs. "deception" by analogy (where it is assumed that the word "deception" was not in the training set).

Other works aim at forming larger groups of related word forms. Gaussier (1999) learns derivational morphology from inflectional lexicons. Orthographically similar words are clustered into relational families. From the induced word families, derivational rules can be acquired, such as the following French verb-to-noun conversions: "produire" → "production", "produire" → "producteur". Schone and Jurafsky (2000, 2001) make use of a Harris-like algorithm to separate suffixes and prefixes from word stems. Whether two orthographically similar word forms are morphologically related is determined from their context of neighboring words. A semantic representation for a word is obtained from the context using Latent Semantic Analysis (LSA). The semantic properties of a word are assumed to emerge from a large context window, whereas syntactic properties can be determined from a narrow window of the immediate word context. In addition to orthographic, semantic, and syntactic similarity, transitive closure is utilized as a fourth component. That is, if "conductive" is related to "conduct" and "conductivity" is related to "conductive", then "conductivity" is likely to be related to "conduct". Bordag (2005) presents a model that functions in a similar way to that of Schone and Jurafsky, but uses simpler mathematics.

Yarowsky and Wicentowski (2000), Yarowsky et al. (2001) and Wicentowski (2002) discover shared root forms for a group of inflected words. Verbs in numerous languages are studied. In addition to orthographic and contextual similarity, frequency distributions are included as a clue to whether words are related. For instance, the English word "singed" can be discarded as a past tense candidate of "to sing" because "singed" is far too rare. Furthermore, parallel corpora in multiple languages are utilized, and one language can function as a "bridge" to another language. For example, the French verb "croire" can be discovered as the root of "croyaient", since these two forms are linked to the English verb "believe" in a parallel text. A missing link from the resembling verb forms "croissant" and "croître" tells us that these are not likely to be related to "croire". Once related words have been proposed, string transductions can be learned that map new words to their likely roots. Wicentowski (2004) learns a set of string transductions from inflection-root pairs and uses these to transform unseen inflections to their corresponding root forms. In this particular experiment, however, supervised training is used.

A further step consists in inducing complete inflectional paradigms, i.e., discovering sets of stems that can be combined with a particular set of suffixes. Goldsmith (2001) formulates his well-known algorithm Linguistica in an MDL framework, whereas Snover and Brent (2001) and Snover et al. (2002) present a similar, probabilistically formulated, model. These models do not predict any word forms outside of the corpus data. If the following English verb forms have been observed: "talk, talks, talking, walk, walked, walks", the verbs "talk" and "walk" will go into separate paradigms: "talk" with the suffix set {∅, s, ing} and "walk" with the suffix set {∅, ed, s}. More general paradigms can be obtained by "collapsing them" together, i.e. clustering them based on context similarity (Hu et al., 2005b). This model can, in principle, predict the missing verb forms "talked" and "walking".

Existing models make the learning of higher-level morphological structure computationally feasible by assuming that a word consists of maximally two, or three, morphemes. This is clearly insufficient for a highly-inflecting and compounding language like Finnish. In recent work, Goldsmith and Hu (2004) and Hu et al. (2005a) move towards morphologies with a larger number of morphemes per word. A heuristic is described that is capable of learning 3- and 4-state FSA:s that model word forming in Swahili, a language with rich prefixation.

Chapter 4

Evaluation

Different morphology learning algorithms focus on different aspects of the learning of morphological structure. Therefore, it is difficult to come up with one single evaluation method that adequately measures the performance of all morphology learning systems. The ultimate test could be to assess how well word forms are understood and produced in a "real setting", where the morphology learner is integrated in an application. However, when evaluating morphology modeling within, for instance, automatic speech recognition applications, there are many factors beyond morphology that considerably affect the result. It is thus valuable also to measure the success of the learning task directly without mediation through an application.

Morfessor has been evaluated in the two manners mentioned: directly, by comparing to a linguistic gold standard, and indirectly, through speech recognition experiments. This chapter reports on the experiments carried out and summarizes some alternative approaches.

4.1 Linguistic Evaluation

This section first presents some common approaches to the linguistic evaluation of morphology learning. Then, the design of the *Hutmegs* morphology evaluation gold standard is described, together with experimental results obtained using Morfessor on different-sized corpora of Finnish and English.

4.1.1 Different Approaches to Linguistic Evaluation

A morphology learning algorithm can be evaluated by its capacity to segment words into morphemes. The morpheme boundaries proposed by the algorithm can be compared to a gold standard, where the "correct" linguistic boundaries have been marked. This evaluation method is particularly popular for assessing the performance of *word segmentation* algorithms, i.e., methods that discover word boundaries in text with no

marked delimiters between words.

Word and morpheme segmenters are typically evaluated in terms of the accuracy and coverage of the proposed word or morpheme *boundaries*; see, e.g., Ando and Lee (2000); Brent (1999); Cohen et al. (2002); Feng et al. (2004); Ge et al. (1999); Johnson and Martin (2003); Kit and Wilks (1999); Nagata (1997); Peng and Schuurmans (2001); Yu (2000). Alternatively, one can apply a somewhat stricter criterion and calculate the proportion of correctly discovered words or morphemes. In order for a word or morpheme to be correct, its *two* delimiting boundaries must naturally be correct. Here the accuracy can be assessed either on the corpus (tokens) or on the lexicon (types). For instance, Feng et al. (2004) apply token-wise evaluation, whereas Brent (1999) and Hacioglu et al. (2003) apply type-wise evaluation.

The above approaches to evaluating segmentation performance are straightforward and have intuitively interpretable results. However, there are some difficulties related to the design of a good gold standard. The ideal level of segmentation is sometimes unclear. For instance, one may wonder whether the following English words borrowed from Latin consist of one or two morphemes: "assume, conceive, consume, presume, receive, resume, subsume". A possible way out is to make use of hierarchical segmentations, e.g., the word "conceivable" can be represented as [[[con][ceiv]][able]]. When comparing the gold standard segmentation to the output of the algorithm, one penalizes for *crossing brackets*, i.e., proposed brackets that overlap, but are not contained within a bracket in the gold standard; see e.g., Ando and Lee (2000); de Marcken (1996). For instance, the following segmentations of "conceivable" would be considered correct, among others: [con][ceiv][able], [conceivable]. By contrast, the segmentations [con][ceivable] and [conceiva][ble] contain crossing brackets and are incorrect. Despite its merits, the crossing-bracket rate is a rather crude evaluation measure; for example, by splitting any string into its individual characters, a crossing-bracket rate of zero is obtained. Consequently, complementary evaluation measures are necessary.

Another solution to the problem of ambiguous levels of segmentation consists in having not one, but several "gold standards", each produced by a native speaker of the target language. The decisions made by the algorithm can be compared to those made by the human experts, especially the tricky cases involving large disagreement among the experts (Baroni, 2003). However, the construction of such a gold standard appears to be a rather time-consuming procedure.

Not only the level of segmentation, but also the exact location of a transition from one morpheme to another can be unclear. For example, does the "e" belong to the stem or the ending in the English past tense form "typed"? The solution proposed in the Hutmegs gold standard is described in the following section. There is, however, another direction one can take, which does not emphasize the exact locations of morpheme boundaries or level of segmentation. If the purpose of the induced model of morphology is to provide a means for producing and "understanding" new word forms, one can evaluate how well the proposed segments (allomorphs) can be mapped onto underlying morphemes in a gold standard. Such a morpheme-allomorph alignment approach is introduced in Publications 1 and 2. Another alignment technique is described by Kontorovich et al. (2003).

In the evaluation method employed in Publications 1 and 2, the idea is to compute the

expected probability that the correct morpheme (deep-level unit of meaning) can be deduced from a suggested surface-level segment (allomorph). For instance, if the word forms "berry" and "berries" are segmented linguistically ("berry, berri+es") the underlying morpheme BERRY can be deduced unambiguously from both allomorphs ("berry, berri"), since neither occurs as an allomorph of any other morpheme. Another possible segmentation could be "berr+y, berr+ies", where "berr" would likewise provide strong evidence for an underlying BERRY. By contrast, leaving only "be" as a stem would be a bad segmentation, due to the ambiguity of this segment: It could signify the verb BE and possibly many other morphemes in addition to the intended BERRY. Unfortunately, many morphs are inherently ambiguous in language, e.g., the English "-s" used as a plural marker of nouns as well as a marker for the third person singular of present tense verbs. Also such "rightly" ambiguous morphs are penalized by this evaluation measure, which is an undesirable effect. The method was therefore abandoned in later publications in favor of the more standard evaluation technique based on Hutmegs. However, the idea behind the morpheme-allomorph mapping approach seems very justified despite the fact that the implementation leaves room for improvement.

Morphology learning algorithms that discover inflectional paradigms are often evaluated in terms of the correctness of the induced paradigms. The evaluation can take place as a manual rating of the goodness of a representative sample (Goldsmith, 2001; Erjavec and Džeroski, 2004). In automated evaluation, it is common to measure the accuracy of the proposed *conflation sets* (Jacquemin, 1997; Gaussier, 1999; Baroni et al., 2002; Schone and Jurafsky, 2000, 2001; Snover et al., 2002). That is, credit is given for words that are grouped together and truly are related word forms (e.g., "singeing" and "singed"). Accordingly, suggested relationships that are incorrect are penalized (e.g., "sing" and "singed"). Moving beyond the discovery of just any related word forms, Yarowsky and Wicentowski (2000); Yarowsky et al. (2001) require that the correct root form be detected for every word; for example, the mapping "singeing → singe" is correct, whereas "singeing → singed" is not. Neuvel and Fulop (2002) make use of a slightly different approach, where previously unseen word forms are generated by the model, and the proportion of the generated words that are indeed possible in the language is assessed.

4.1.2 Hutmegs

The major evaluation method applied in this work consists in comparing the proposed placement of morpheme boundaries to a linguistic gold standard segmentation. As mentioned above, such an evaluation is straightforward and intuitive – provided that an adequate gold standard exists. Segmentation gold standards are valuable resources, which require large amounts of work by linguistic experts. Once they exist they can be used for other purposes beyond evaluation of unsupervised learning algorithms. An example is given in Section 4.2, where one of the tested language models is based on a linguistic morpheme segmentation.

The need for publicly available resources is apparent. Within the Penn Chinese Treebank project (Xue et al., 2004) a 100 000 word corpus of Mandarin Chinese has been segmented into words, tagged with part-of-speech tags and provided with syntactic bracketing. In Western languages, there are spaces between the words, and word seg-

mentation of written text is trivial. However, large amounts of work has gone into the annotation of corpora, e.g., part-of-speech tagging, morphological analysis and syntactic bracketing. For American English, the Penn Treebank (Marcus et al., 1993) is an example of existing resources.

A more detailed annotation of the morphological structure of words can be found in the CELEX databases of English, Dutch and German (Baayen et al., 1995). Among other things, the databases provide information on the derivational and compositional structure as well as inflectional paradigms of tens of thousands of word forms. Corresponding morphological analyses of word forms, though less detailed, can be obtained using software based on the two-level morphology of Koskenniemi (1983). Such TWOL analyzers exist for, e.g., Finnish, the Scandinavian languages (Swedish, Danish, Norwegian), English and German.[1]

What the existing analyzers and databases lack, however, is an explicit morpheme *segmentation* of the *surface* forms of the words. The information provided by CELEX and TWOL can be interpreted as a morpheme segmentation of a word, but the morphemes are not indicated as they are realized on the surface, as word segments or allomorphs, but as deep-level morphemes (or base forms), e.g., the English word "bacteriologist" yields the segmentation "bacterium+ology+ist".

Publication 8 describes the production of segmentations of the surface forms of both Finnish and English words. These segmentations are proposed as a reference, or gold standard, which can be used freely for research purposes. The gold standard is called *Hutmegs*, which is an abbreviation for "Helsinki University of Technology Morphology Evaluation Gold Standard".

The Hutmegs gold standard relies on existing resources: the Finnish TWOL and the contents of the English CELEX database[2]. The additional work consists in producing an alignment between the surface, or allomorph, segmentation and the deep-level, or morpheme, segmentation, as in the following examples:[3]

```
tieteellisessä  tietee:tiede|N llise:DN-LLINEN ssä:INE
bacteriologist  bacteri:bacterium|N olog:ology|s ist:ist|s
```

The Finnish Gold Standard contains segmentations for 1.4 million distinct word forms (word types). The English Gold Standard contains segmentations for 120 000 word types. The locations of morpheme boundaries in the surface form is not always obvious and the interpretation chosen relies on Hakulinen (1979) for Finnish and Quirk et al. (1985) for English. If a word has many possible segmentations, all of these are supplied, e.g., "evening" (time of day) vs. "even+ing" (verb).

The segmentation has been performed semi-automatically with the help of rulesets and a number of scripts. For Finnish, some extra processing was required for derived word forms in order to obtain the baseform of the root rather than the baseform of the deriva-

[1] Licenses can be obtained from Lingsoft, Inc. `http://www.lingsoft.fi`.

[2] An alternative source for creating the English gold standard could have been Englex / PC-KIMMO version 2, `http://www.sil.org/pckimmo/v2/doc/guide.html`.

[3] The Finnish word 'tieteellisessä' means 'in [the] scientific'. The segments shown consist of an allomorph part separated from the morpheme part by a comma. The morpheme part may contain a part-of-speech tag, which is preceded by a vertical bar.

tion (e.g., “tiede” instead of “tieteellinen”; i.e., “science” instead of “scientific”). For English, possessive forms of nouns had to be added (e.g., “king’s, queen’s”). In addition, so-called “fuzzy morpheme boundaries” have been introduced as an optional feature. Fuzzy boundaries can be applied for cases where it is inconvenient to define one exact transition point between two morphemes.

Fuzzy Locations of Morpheme Boundaries

In some cases, the “linguistically correct” location of a morpheme boundary may not seem the only plausible solution. Historic development of the language may affect the way linguists describe the contemporary morphology. However, from the point of view of natural language applications, this may not be the optimal description.

In the Hutmegs gold standard, there is a notation for marking “fuzziness” of morpheme boundaries. The fuzziness consists in alternative locations for the same morpheme boundary, i.e., the boundary does not have an unambiguous location. Fuzziness is allowed as follows: If at the end of a morpheme, there is one phoneme (or sometimes more) that may be totally absent in some allomorphs of the morpheme, this phoneme is considered to lie on a fuzzy boundary between two morphemes. (The latter morpheme is always a suffix.) The phoneme is on the fuzzy boundary only if it alternates phonologically with a “zero phoneme”, not if it is replaced by another phoneme. This is a somewhat arbitrary definition, but our motivation is that the phoneme (or phonemes) behaves as a seam, or a joint, which is not always needed. If the “joint phoneme” is present only in combination with some following suffixes, it could be considered part of the suffix as easily as part of the preceding morpheme.

For instance, in English, the stem-final ‘e’ in verbs is dropped in some forms. The user of the gold standard can choose whether to consider only the traditional linguistic segmentation correct, as in:

`invite, invite+s, invit+ed` and `invit+ing`,

or whether also to allow for an alternative interpretation, where the ‘e’ is considered part of the suffix, as in:

`invit+e, invit+es, invit+ed` and `invit+ing`,

In the former case, there are two allomorphs of the stem (“invite” and “invit”), and one allomorph for the suffixes. In the latter case, there is only one allomorph of the stem (“invit”), whereas there are two allomorphs of the third person in the present tense (“-s” and “-es”) and an additional infinitive ending (“-e”). Since there are a much greater number of different stems than suffixes in the English language, the latter interpretation lends itself to more compact Item and Arrangement models of morphology.[4]

Corresponding fuzzy cases exist in the Finnish language, and in greater number than for English. For instance, the proper name “Windsor” has three allomorphs in Finnish:

[4]Note that the possible segmentation “invite+d” is *not* considered correct, due to the fact that there is no indication that the regular past tense ending “-ed” ever loses its “e”, whereas the preceding stem unquestionably does so, e.g., in “inviting”.

"Windsor" (nominative singular, genitive plural), "Windsori" (oblique cases in singular, nominative plural), and "Windsore" (oblique cases in plural). The following segmentations are linguistically conventional, e.g., "Windsor, Windsori+n, Windsori+lla, Windsori+t, Windsor+i+en, Windsore+i+lla". Since the final vowel of the stem is not always present, it belongs to a fuzzy boundary, and can therefore also be attached to the ending: "Windsor, Windsor+in, Windsor+illa, Windsor+it, Windsor+i+en, Windsor+ei+lla". Further examples and details can be found in Publication 8.

4.1.3 Experiments

The Hutmegs package contains some evaluation scripts in addition to the morphological gold standard segmentations of Finnish and English words. In the quantitative evaluation, three measures are used: *precision*, *recall*, and *F-measure* of the proposed placement of morpheme boundaries. These measures assume values between zero and 100 %, where high values reflect good performance.

Evaluation Measures

Precision is the proportion of correct boundaries among *all morph boundaries suggested by the algorithm*. Recall is the proportion of correct boundaries discovered by the algorithm in relation to *all morpheme boundaries in the gold standard*. For example, suppose that a segmentation algorithm proposes the following segmentations for two English words: "beautiful+ly, flu+s+ter+ed". The corresponding gold standard segmentations are: "beauti+ful+ly, fluster+ed". For this minimal sample the total number of boundaries suggested by the algorithm is four and the total number of boundaries in the gold standard is three. Two of the boundaries suggested by the algorithm are correct according to the gold standard. Thus, the resulting precision is $\frac{2}{4} = 50\,\%$ and the resulting recall equals $\frac{2}{3} = 67\,\%$.

Precision and recall can be combined into a third evaluation measure, (evenly-weighted) F-measure, which is the harmonic mean of the two:

$$\textit{F-Measure} = 1/[\frac{1}{2}(\frac{1}{\textit{Precision}} + \frac{1}{\textit{Recall}})]. \tag{4.1}$$

In our example, an F-measure value of $1/[\frac{1}{2}(\frac{4}{2} + \frac{3}{2})] = \frac{4}{7} = 57\,\%$ would be obtained.

The above evaluation is performed on word types, i.e., each word form is counted only once, which implies that frequent and rare words have equal weight in the evaluation. Alternatively, the evaluation can be based on word tokens, where each word is counted as many times as it occurs in the corpus. For instance, if "beautifully" occurs 100 times and "flustered" two times in our corpus, the total number of morpheme boundaries proposed by the algorithm is $100 \cdot 1 + 2 \cdot 3 = 106$ and the total number of boundaries in the gold standard amounts to $100 \cdot 2 + 2 \cdot 1 = 202$. The number of boundaries proposed that are correct equals $100 \cdot 1 + 2 \cdot 1 = 102$. Thus, the token-wise precision is $\frac{102}{106} = 96\,\%$, token-wise recall is $\frac{102}{202} = 50\,\%$, and token-wise F-measure is $1/[\frac{1}{2}(\frac{106}{102} + \frac{202}{102})] = 66\,\%$.

A general tendency across languages is that the most irregular forms occur with high

frequency, whereas rare words predominantly have regular inflection patterns. Consequently, token-wise evaluation stresses the performance of the algorithm on frequent, less regular cases. However, from the point of view of generalizability (prediction of unobserved forms), it is crucial to learn regular morphology as accurately as possible. This motivates the use of type-wise rather than token-wise measures. In the following, all reported figures are calculated on word types.

Algorithms and Data Sets

Experiments have been carried out on Finnish and English data. The data consist of plain text with no linguistic annotations. The Finnish data are composed of prose and news texts from the Finnish IT Center for Science (CSC)[5] and the Finnish National News Agency. The English data consist of the Brown corpus, a sample of the Gigaword corpus[6], as well as prose, news and scientific texts from the Gutenberg project[7].

Four different versions of Morfessor are tested: Baseline, Baseline-Length, Categories-ML, and Categories-MAP. In addition, Goldsmith's algorithm Linguistica is included for comparison (Goldsmith, 2001, 2005).

Evaluations are carried out on data sets containing 10 000, 50 000, 250 000 and 16 million words for Finnish. The same data set sizes are used for English, except for the largest data set, which contains 12 million words. Parameter values (Equations 3.22 and 3.23) have been set using held-out development sets, which are not part of the final test sets.

Results

The F-measures of the morpheme boundaries proposed by the algorithms are shown in Figure 4.1. In the comparison against the Hutmegs gold standard, "fuzzy" morpheme boundaries have been allowed. (The raw numerical values of the F-measures, together with the related precision and recall values, can be found in Appendix A.5.1.)

In the results for Finnish, the five assessed algorithms fall into three distinct groups (Fig. 4.1a): (i) The best performing group consists of the two Categories versions of Morfessor. They increase their lead with increasing data size. Categories-MAP rivals Categories-ML as the best-performing algorithm. For the data sizes 10 000 and 250 000 words the difference between the two is not even statistically significant (T-test level 0.05). (ii) The length prior applied in the Baseline-Length method seems slightly beneficial compared to the plain Baseline, but the difference is statistically significant only on the 50 000 word data set. (iii) Linguistica is the worst-performing algorithm on Finnish data.

Some different tendencies can be observed in the English results (Fig. 4.1b): (i) Generally, in terms of F-measure, the algorithms perform better than on the corresponding

[5] `http://www.csc.fi/kielipankki/`

[6] The Gigaword sample and the Brown corpus are available at the Linguistic Data Consortium: `http://www.ldc.upenn.edu/`.

[7] `http://www.gutenberg.org/browse/languages/en`

Finnish data sets. This may be a consequence of the less complex English morphology, which is easier to model. This may also explain the fact that there is a less evident advantage of the more sophisticated Categories models over the simpler Baseline models. Linguistica is more suited for this type of language and performs second best on the 50 000 and 250 000 word sets. Note, however, that among the four best-performing methods the difference in performance is statistically significant only between Categories-ML and the lowest-scoring algorithm at each data size (Linguistica at 10 000 words; Baseline-Length at 50 000 and 250 000 words). (ii) The performance degrades somewhat for most algorithms on the largest English data set. This may be caused by the multitude of foreign words contained in this set, which causes patterns to be discovered that do not belong to contemporary English morphology, e.g., "plex+us, styl+us". (iii) The length prior is beneficial; Baseline-Length clearly outperforms the plain Baseline method, especially on small data sizes. Note that the Baseline methods are trained on lists of unique word types rather than corpora of running text. In an English and Finnish corpus of the same size, the number of distinct word forms is much lower in the English corpus (see Figure 4.2). On smaller data sets, the impact of a prior is larger, which explains why the length prior is more effective for English than Finnish.

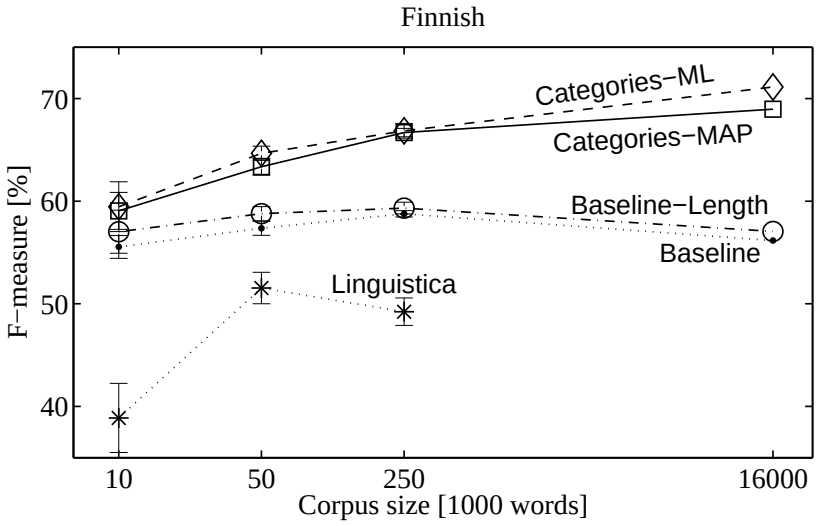
Out-Of-Vocabulary Words

Unfortunately, gold standard segmentations do not exist for all word forms in the test sets. The morphology-learning algorithms suggest a morph segmentation for all words in the data, but those words that are out-of-vocabulary, i.e., not contained in the gold standard, are disregarded in the evaluation. Out-of-vocabulary (OOV) words make up about 3.5 – 4.0 % of the word *tokens* in the Finnish and English test data, regardless of the size of the subset used.

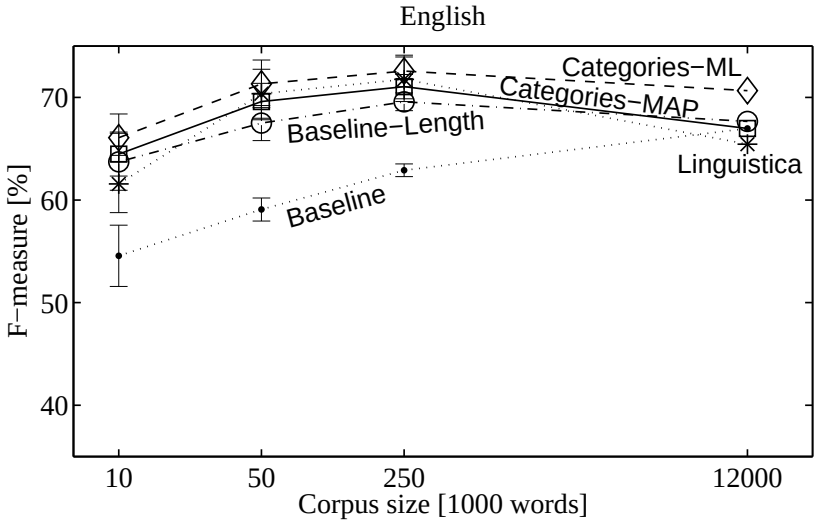
However, the F-measures reported are based on word *types*. The proportion of out-of-vocabulary word *types* increases steadily with the size of the set: In the Finnish 10 000 word test set, 4.3 % of the word types are OOV, and in the Finnish 16 million word set, 15.2 %. The corresponding figures for English are 7.5% (10 000 word set), 16.4 % (250 000 word set) and 56.0 % (12 million word set). The high rate of OOV word types for the largest English test set seems to be due to the fact that a large number of names, both English and foreign, are missing from the gold standard.

As mentioned in the "Results" section above, it is difficult for the algorithms to learn a correct *English* morphology from data containing a large mass of rare "atypical" foreign words. Undesired patterns are discovered, which somewhat affects the "typical" English words, and which is seen as decreased performance for the largest 12 million word set in Figure 4.1b. The negative effect seems to be even more drastic on the foreign words themselves, as suggested by the following additional evaluation:

On the largest English set with the very high OOV rate, a small sample of 125 OOV's were picked by random and correct segmentations for these words were produced manually. F-measures were then calculated for the five algorithms on this small sample. That is, the training data remained the same from the previous experiment, but the evaluation was performed on a new subset of words. Here Categories-MAP performed the best with an F-measure of only 47 %. Categories-ML came second (41 %), whereas the Baselines



(a)



(b)

Figure 4.1: Morpheme segmentation performance of four versions of Morfessor as well as Goldsmith's algorithm Linguistica, both on (a) Finnish and (b) English test data. Each data point is an average of 5 runs on separate test sets, with the exception of the 16 million words for Finnish and the 12 million words for English (1 test set). In these cases the lack of test data constrained the number of runs. The standard deviations of the averages are shown as intervals around the data points. There is no data point for Linguistica on the largest Finnish test set, because the program is unsuited for very large amounts of data due to its considerable memory consumption.

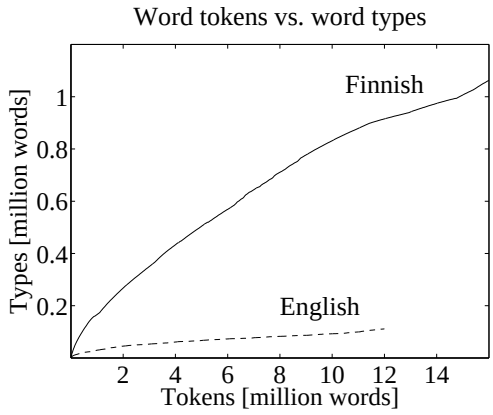


Figure 4.2. The number of unique word forms (types) encountered in growing portions of running text (tokens) of Finnish and English.

and Linguistica achieved around 30 %. If the sample is representative enough, the hardly surprising result suggests that a morphology model should be *trained* on typical words of the language, after which this model should only be *applied* on less frequent, atypical words.

The OOV rate for the largest Finnish set is considerably lower, and therefore no corresponding investigation has been performed on the Finnish data. The Baselines and Linguistica do, however, display decreasing performance on large amounts of data, which may be due to the same reason: atypical words (Fig. 4.1a).

Random Segmentation

When assessing the level of performance of an algorithm, it is important that one understands the difficulty of the task. A useful algorithm must obtain a considerably better level than what is achievable by very simple means. A commonly used "worst benchmark" consists in using a *random* technique, in our case *random segmentation*.

Suppose that boundaries are placed by random after letters in the words in the data. Here the last letter of each word is excluded, because it is known to be followed by a boundary. In the Finnish gold standard, each 5.3^{th} non-word-final letter is on average followed by a morpheme boundary. Regardless of which probability is used to produce random boundaries, every 5.3^{th} suggested boundary is expected to be correct, because the boundaries are expected to be evenly distributed. This corresponds to an expected precision of $\frac{1}{5.3} = 19\,\%$. The corresponding value for English is lower, because morpheme boundaries are less frequent: $\frac{1}{8.0} = 13\,\%$.

Expected recall is maximized when the probability of suggesting boundaries is maximized. This ensures that as few correct boundaries as possible are missed. By "guessing" a boundary after each letter with a 100 % probability, a recall of 100 % is obtained. That is, optimal "random" segmentation is achieved by splitting the words in the corpus into individual letters!

The expected maximal F-measures obtained using random segmentation are thus $1/[\frac{1}{2}(\frac{1}{19\,\%} + \frac{1}{100\,\%})] = 32\,\%$ (Finnish), and $1/[\frac{1}{2}(\frac{1}{13\,\%} + \frac{1}{100\,\%})] = 22\,\%$ (English).

All the compared algorithms perform clearly better than these very worst benchmarks.

4.1.4 Example Segmentations

The quantitative evaluation has only concerned the accuracy of the placement of morph boundaries. A few examples will now serve to illustrate phenomena *beyond* mere segmentation. Table 4.1 contains a number of Finnish and English words, segmented using the Baseline-Length and Categories-MAP algorithms. In addition, the corresponding gold standard segmentations are supplied.

The examples verify that the introduction of a simple morphotactics in the Categories models reduces the occurrences of under- and oversegmented words as well as misalignments due to the insensitivity of context, which are observed in the Baseline models. Examples of such cases comprise the Finnish words “epä+tasa+paino+inen” (“imbalanced”), ‘jani+lle’ (“for Jani”), as well as the English words “photo+graph+er+s” and “fluster+ed”.[8]

The algorithms produce different amounts of information: the Baseline and Baseline-Length methods only produce a segmentation of the words, whereas the other algorithms (Categories-ML, Categories-MAP and Linguistica) also indicate whether a segment functions as a prefix, stem, or suffix. Tagging of categories can be very useful. It can be applied in order to identify and separate semantic segments (mainly stems) from syntactic segments (mainly affixes). The stems contained in a word form could be considered as a canonical (or base) form of the word, whereas the affixes could be considered as inflections. Such a canon form for words could be an alternative to the base forms retrieved by hand-made morphological analyzers or stemming algorithms, which are used, e.g., in information retrieval.

The lexicon learned by Categories-MAP contains hierarchical representations, which can be interpreted as the attachment hierarchy of the morphemes. With the current model, the construction of the hierarchy is likely to take place in the order of most frequently co-occurring word segments. Sometimes this is also grammatically elegant, e.g., Finnish: “[epä [[tasa paino] inen]]” (“imbalanced”, literally bracketed as “[un [[even weight] ed]]”), and English: “[[[photo graph] er] s]”, “[[un [expect ed]] ly]”. But the probability of coming up with grammatically less elegant solutions is also high, e.g., English “[micro [organism s]]”.

The Morfessor algorithms can incorrectly “overgeneralize” and, for instance, suggest a suffix, where there is none, e.g., “maclare+n” (“MacLaren”). Such overgeneralization should conceivably be less common in Linguistica, which learns paradigms (called signatures). In general, to propose the segmentation “maclare+n”, other forms of the proposed stem would be expected to occur in the data, such as “maclare” or “maclare+ssa”. If none of these exist, the segmentation should be discarded. However,

[8] Note that Categories-MAP produces the correct segmentation for the word “epätasapainoinen”, whereas the corresponding gold standard segmentation is strange. The bad gold standard segmentation is due to the fact that FINTWOL does not mark in any way the boundary between the prefix “epä-” and a following stem. Also the English gold standard contains some debatable cases: the identification of prefixes in the words “*con*-figuration, *dis*-appoint, *in*-sure” may have a historical motivation, but hardly reflects the contemporary semantics of these words.

Table 4.1: Examples of Finnish and English morpheme segmentations learned by two versions of Morfessor from the largest test sets. The corresponding gold standard segmentations are also supplied. Proposed prefixes are underlined, stems are rendered in **bold-face**, and suffixes are *slanted*. Square brackets [ ] indicate higher-level stems and parentheses () higher-level suffixes in the hierarchical lexicon of Categories-MAP.

Baseline-Length	Categories-MAP	Hutmegs Gold Standard
aarre kammioissa	[**aarre kammio**] *issa*	**aarre kammio** *i ssa*
aarre kammioon	[**aarre kammio**] *on*	**aarre kammio** *on*
bahama laiset	**bahama** *laiset*	**bahama** *laise t*
bahama saari en	**bahama** [**saari** *en*]	**bahama saar** *i en*
epä esteettis iksi	epä [[**esteet** *ti*] *s*] *iksi*	**epäesteett** *is i ksi*
epätasapaino inen	[epä [[tasa **paino**] *inen*]]	epätasa **painoinen**
haapa koskeen	[**haapa** [**koskee** *n*]]	**haapa koske** *en*
haapa koskella	[**haapa** [**koske** *lla*]]	**haapa koske** *lla*
ja n ille	**jani** *lle*	**jani** *lle*
jäädyttä ä kseen	[**jäädy ttää**] *kseen*	**jäädy** *ttä ä kse en*
ma clare n	**maclare** *n*	–
nais autoilija a	[nais [**autoili** *ja*]] *a*	**nais autoili** *ja a*
pää aiheesta	**pää** [**aihe** *esta*]	**pää aihee** *sta*
pää aiheista	[pää [**aihe** *ista*]]	**pää aihe** *i sta*
päähän	[pää **hän**]	**pää** *hän*
sano ttiin ko	[**sano** *ttiin*] *ko*	**sano** *tt i in ko*
työ tapaaminen	**työ** [**tapaa** *minen*]	**työ tapaa** *minen*
töhri misistä	**töhri** (*mis istä*)	**töhri** *mis i stä*
voi mmeko	[[**voi** *mme*] *ko*]	**voi** *mme ko*
accomplish es	[**accomplish** *es*]	**accomplish** *es*
accomplish ment	[**accomplish ment**]	**accomplish** *ment*
beautiful ly	[**beautiful** *ly*]	**beauti** *ful ly*
configu ration	[**configur ation**]	con **figur** *ation*
dis appoint	**disappoint**	dis **appoint**
express ive ness	[**expressive** *ness*]	**express** *ive ness*
flu s ter ed	[**fluster** *ed*]	**fluster** *ed*
insur e	**insure**	in **sure**
insur ed	[**insur** *ed*]	in **sur** *ed*
insur es	[**insure** *s*]	in **sure** *s*
insur ing	[**insur** *ing*]	in **sur** *ing*
long fellow 's	[[**long fellow**] *'s*]	–
master piece s	[[**master piece**] *s*]	**master piece** *s*
micro organism s	[**micro** [**organism** *s*]]	micro **organ** *ism s*
photograph ers	[[[**photo graph**] *er*] *s*]	photo **graph** *er s*
re side d	**resided**	**resid** *ed*
re side s	[**reside** *s*]	**reside** *s*
re s id ing	[re **siding**]	**resid** *ing*
un expect ed ly	[[un [**expect** *ed*]] *ly*]	–

especially with large amounts of data Linguistica is oversensitive to common strings that occur at the end of words and proposes segmentations, such as “allu+de”, “alongsi+de”, “longitu+de”; “anyh+ow”, “highbr+ow”, “longfell+ow”.

4.1.5 Other Languages

After the completion of the work on the publications included in this thesis, linguistic evaluation data have become available for two additional languages: Turkish and the Egyptian dialect of Arabic. Morpheme segmentation results obtained using three versions of Morfessor on data of these two languages are shown alongside results for Finnish and English in Table 4.2.

The results reported for Finnish, English, and Turkish correspond to the performance of Morfessor in the so-called Morpho Challenge competition, organized within the EU PASCAL Network of Excellence (Kurimo et al., 2006a; Creutz, 2006). The Arabic results are based on yet unpublished work.

Morfessor did not officially take part in the Morpho Challenge, since this challenge was organized by the developers of Morfessor. However, Morfessor would have outperformed all participants in the Finnish and Turkish segmentation tasks and would have come second in the English task. It is especially encouraging to see that the Turkish results are so good, since Turkish data were never utilized during the development of the Morfessor model.

Hutmegs was used as the linguistic gold standard for Finnish and English. The reported F-measures are slightly lower than those obtained in the earlier experiments, which is due to the fact that the “fuzzy” morpheme boundary option was not in use, and thus fewer alternative segmentations were considered correct. The Turkish linguistic segmentations are based on a morphological parser developed at Bogazici University (Cetinoglu, 2000; Dutagaci, 2002).

The Arabic data set, consisting of transcripts of telephone conversations[9], is considerably smaller than the data sets of the other languages. The gold standard used for Arabic is based on a lexicon of Egyptian Colloquial Arabic[10]. All vowels are marked in the data, unlike the common practice when writing Arabic. Arabic words consist of a stem, possibly preceded by a number of prefixes, and followed by a number of suffixes. The stems of many Arabic words are formed through the insertion of a vowel pattern into a “consonantal skeleton”; for instance, the consonant sequence “k-t-b” means “writing-related”, and the following stems can be formed, among others: “kitaab” (book), “kutub” (books), “kaatib” (writer). However, in the gold standard utilized in these experiments, the stems have been treated as unanalyzed wholes; that is, “kitaab”, “kutub”, and “kaatib” are treated as separate morphemes.

By inspecting the results in Table 4.2, some general conclusions can be drawn: The performance of the Morfessor Baseline algorithm varies greatly over the data sets of different sizes and languages, whereas Categories-ML and Categories-MAP seem to perform

[9] Callhome Egyptian Arabic Transcripts, provided by the Linguistic Data Consortium (LDC): `http://www.ldc.upenn.edu`.

[10] also available through the LDC

Table 4.2: F-measures obtained when running three versions of Morfessor on data sets of four different languages. The sizes of the data sets used in the experiments are also shown (token and type count).

Corpus size / Method	Finnish	English	Turkish	Egyptian Arabic
Word tokens	32 000 000	24 000 000	17 000 000	150 000
Word types	1 600 000	170 000	580 000	17 000
Baseline	54.2	66.0	51.3	41.7
Categories-ML	67.0	69.0	69.2	67.9
Categories-MAP	66.4	66.2	70.7	68.1

on an equal level independent of language (with F-measures of about 70 %). The more elegantly formulated Categories-MAP outperforms the more heuristic Categories-ML method on Turkish and Egyptian Arabic data, whereas the opposite is true for Finnish and English.

4.2 Evaluation through Speech Recognition Experiments

As explained in Section 2.3.2, there are potential disadvantages associated with large lexicons in speech recognition. If the lexicon is composed of all distinct word forms encountered in a large corpus, the number of entries is likely to be very high: hundreds of thousands or even millions of words when dealing with a highly-inflecting and compounding language, such as Finnish. One way to limit the size of the lexicon is to split the words into smaller sub-word units.

The aim of the following investigation is to compare different types of vocabularies (i.e., lexicons) in Finnish speech recognition: vocabularies of words, vocabularies of morphs, and vocabularies of syllables. Two different experiments have been carried out, originally presented in Publications 6 and 7. In order to comply with the terminology of Publication 7, the units that the vocabulary consists of are called word *fragments*. A fragment can thus denote a single letter, a syllable, a morph, or an entire word.

In the experiments, described next, n-gram language models based on different word fragments have been evaluated. The models have been compared both in terms of their prediction capacity (cross-entropy on test corpora; cf. Section 2.3.3) and as integrated components of a large vocabulary speech recognition system.

4.2.1 Word Fragment Inventories

A text corpus consisting of 32 million Finnish words has been used as data. The corpus is composed of books, magazines, and newspapers from the Finnish IT Center for Science (CSC), and of short newswires from the Finnish News Agency. The words in this corpus have been split, in turn, into: (i) *syllables*, (ii) *statistical morphs*, and (iii) *grammatical morphs*. Additionally, the words have been left unsplit, as (iv) entire *words*.

The term "statistical morphs" denotes morphs produced by the Morfessor Baseline algorithm, whereas the "grammatical morphs" correspond to morphs obtained from the linguistic morpheme segmentation of the Finnish Hutmegs gold standard.

Once the desired splitting of the training corpus has been obtained, an n-gram model is estimated over the sequence of word fragments. Table 4.3 shows different segmentations of one particular phrase of the training corpus. The division into "Experiment I" and "Experiment II" reflects the two different experimental setups of Publication 6 and 7, respectively. Syllables were investigated only in Publication 6, whereas grammatical morphs were studied only in Publication 7. The size of the fragment inventory (lexicon) of each model is indicated within parentheses after the model name. For instance, there are 79 000 distinct fragments in the grammatical morph model (type count).

When a text is split into word fragments, the locations of word boundaries must be modeled explicitly. This has been solved by introducing an additional word boundary unit (#), which functions as any other fragment in the sequence. Naturally, in case of the word models, a separate word boundary fragment is superfluous, because every word can implicitly be assumed to end in a word boundary. Also these "implicit boundaries" have been marked explicitly in Table 4.3.

The grammatical morph model and word models cannot cover the whole vocabulary of the Finnish language. Some words lack a grammatical morphological analysis, since they are not present in the Hutmegs gold standard, i.e., they are not recognized by the FINTWOL morphological analyzer. Correspondingly, it is impossible to include all possible word forms in a word model. The treatment of out-of-vocabulary words will be described below, separately for Experiment I and II.

4.2.2 Speech Recognition Tasks

The speech recognition performance has been assessed using two different speech data sets, which are independent of the text corpus used in the training of the language models. For these two sets, acoustic models (i.e., models of speech sounds) are trained and evaluated separately. The speech recognition is *speaker-dependent*; that is, the acoustic models are tuned to one particular speaker.

The first data set is a Finnish audio book[11] containing twelve hours of read speech from one female speaker. The second data set consists of about five hours of news broadcasts read by another female speaker. The major parts of these data sets have been used as training sets for training acoustic models. Smaller independent sets have been used for parameter tuning (development set) and final evaluation (test set). In addition to training acoustic models, the reference transcriptions of the training portions of the book and news data are utilized for evaluating the cross-entropies of the language models.

[11] *Syntymättömien sukupolvien Eurooppa* by Eero Paloheimo

Table 4.3: A phrase of the training corpus: *"Tuoremehuasema aloitti maanantaina omenamehun puristamisen Pyynikillä."* transcribed using different fragment inventories. (An English translation reads: *"On Monday a juice factory started to press apple juice in Pyynikki."*) The fragments are separated by space. Word breaks are indicated by a number sign (#). In case of the word models, the word breaks are part of larger fragments; otherwise they are units of their own.

Experiment I

Model	Segmentation
Syllables (37k)	tuo re me hu a se ma # a loi tti # maa nan tai na # o me na me hun # pu ris ta mi sen # pyy ni ki llä #
Statist. morphs (64k)	tuore mehu asema # aloitti # maanantaina # omena mehu n # purista misen # pyynikillä #
Words-OOV (64k)	OOV# aloitti# maanantaina# OOV# OOV# pyynikillä#

Experiment II

Model	Segmentation
Statist. morphs (26k)	tuore mehu asema # al oitti # maanantai na # omena mehu n # purista misen # pyy nik illä #
Gramm. morphs (79k)	tuore mehu asema # aloitt i # maanantai na # omena mehu n # purista mise n # p yy n i k i ll ä #
Words (410k)	t u o r e m e h u a s e m a# aloitti# maanantaina# omenamehun# puristamisen# pyynikillä#

4.2.3 Experiment I

The first experiment was performed only on the audio book data, with three types of word fragments: syllables, statistical morphs, and words. For each fragment inventory, a *trigram* language model was estimated over the segmented corpus. The CMU-Cambridge language modeling toolkit (Clarkson and Rosenfeld, 1997) was used with Katz backoff (Katz, 1987) and Good-Turing smoothing (Good, 1953). The CMU toolkit is rather restrictive in terms of the maximal lexicon size it supports. The lexicon can contain no more than 64 000 entries (in our case word fragments). This is especially to the disadvantage of the word model.

Syllables

In Finnish, syllabification is based on pronunciation rather than morphological structure. Each syllable must contain a vowel or a vowel cluster as its nucleus. If the nucleus is followed by a consonant cluster, the last of the consonants is transferred to the next syllable, e.g., "a+se+ma" (station), "de+monst+roi+da" (to demonstrate). This simple scheme may, however, produce the wrong result for compound words, where the syllable boundary should, in fact, be placed according to morphological criteria, at the

inner boundary of the compound, e.g., “a·va·ruus+a·se·ma” (space station) rather than “a·va·ruu+sa·se·ma”.[12]

Although mistakes are made with some compounds, syllabification based on pronunciation has been applied in this work. The simplicity of this approach is emphasized by the straightforwardness of Finnish orthography; each letter essentially corresponds to one phoneme, except the “ng”-sound and double letters, which represent long sounds. Splitting long sounds in two parts was not considered desirable for speech recognition. Therefore, long phonemes have always been kept within the same fragment, which can be seen in the examples in Table 4.3. (In standard syllabification, the double letters would be separated from each other, e.g., “Pyy+ni+kil+lä” instead of “Pyy+ni+ki+llä”.)

Foreign words are problematic in the sense that their pronunciation is not easily available. In the current experiment, these words have only been transcribed very rudimentarily. Letters that are not used in genuine Finnish words have been replaced, e.g., “c” becomes “k”, “x” becomes “ks”, and “z” becomes “ts”. The inability to transcribe foreign words explains the surprisingly high number of distinct syllables: 37 000. Consider, for instance, the following syllabification of non-Finnish words, which adds some very dubious syllables to the lexicon: “mkknight” (mcknight), “vords+vorth” (wordsworth).

Statistical Morphs

The Morfessor Baseline algorithm is applied on the 32 million word text corpus. Morfessor is here trained on word tokens; that is, frequent word forms have more weight than rare ones. Consequently, frequent words mostly come out unsplit, whereas rare word forms are split into smaller parts. The total number of morphs discovered is 300 000, which exceeds the 64k lexicon size limit. Therefore, the morph inventory is pruned to contain only the 64 000 most frequent morphs, after which the corpus is resegmented using only these morphs. The resegmentation is carried out using the Viterbi algorithm (Viterbi, 1967; Forney, 1973). The out-of-vocabulary rate remains 0 % regardless of the pruning of the morph lexicon, because the lexicon contains one fragment for each individual phoneme. Thus, in the worst case any word form can be rewritten as a sequence of phonemes.

Words-OOV

The text corpus contains 1.6 million distinct word forms, but no more than the 64 000 most frequent words can be included in the recognition lexicon. The remaining words are replaced by a special out-of-vocabulary (OOV) fragment, as illustrated in Table 4.3. The proportion of OOV’s is high: 20 % (token count). The words replaced by the OOV symbol cannot be predicted by the language model, nor can they be recognized by the speech recognizer.

[12] Note also that one letter syllables are customarily avoided, because they may look awkward in written text. That is, one would not like to see the initial “a” of “asema” at the end of one line and “sema” in the beginning of the next line. The preferred syllabification thus becomes: “ase+ma” (Iisa et al., 2002). In the current speech recognition experiment, this rule has been disregarded.

Table 4.4: Experiment I (Book data): Prediction ability of the language models (cross-entropy) as well as speech recognition performance (word error rate). The differences between the models are statistically significant.

Model	Cross-entropy	Word-error rate
Statistical morphs	14.8 bits	31.7 %
Syllables	16.0 bits	43.9 %
Words-OOV	∞	56.4 %

Results

The three types of word fragment inventories have been compared, in terms of language modeling as well as speech recognition performance. Table 4.4 shows that the statistical morphs outperform both the syllable and the word model: the morph model achieves the lowest cross-entropy as well as the lowest word-error rate. Note that all reported figures refer to words, which allows for a fair comparison across models, regardless of the different fragment inventories.

The word-error rate is calculated as the sum of the number of substituted words, inserted words and deleted words divided by the total number of words in the correct transcription of the speech data. In case of the words-OOV model, a word error rate below the OOV rate (20 %) would be impossible to achieve. The presence of OOV's also makes an estimation of cross-entropy meaningless, because these words have zero probability in the language model.

All tested language models of Experiment I are trigram models. However, the typical length of a fragment differs from one model to another: syllables are the shortest, and words are the longest, on average. Therefore, the average trigram spans a different number of phonemes in each model: 5 phonemes / syllable trigram, 9 phonemes / morph trigram, and 19 phonemes / word trigram. Consequently, one may suspect that especially the syllable and morph models could benefit from higher-order n-grams. In order to evaluate the models more extensively, the optimal n-gram level for each fragment set should be determined first, and then the models should be compared with each other.

4.2.4 Experiment II

The second experiment (Publication 7) was performed on both audio data sets: book and news data. The CMU-Cambridge language modeling toolkit was replaced by the SRI toolkit (Stolcke, 2002), which supports larger lexicons and better smoothing (Modified Kneser-Ney). n-Gram models of orders 2–7 were estimated and their cross-entropy on the two test sets were evaluated. Speech recognition tests were carried out on n-grams of orders 3–5. Three types of fragment inventories have been studied: statistical morphs, grammatical morphs, and words. The text data has been preprocessed using software that produces satisfactory pronunciations for foreign names, and expands numbers and abbreviations to full written forms (Volk, 2004). The modeling of speech sound duration

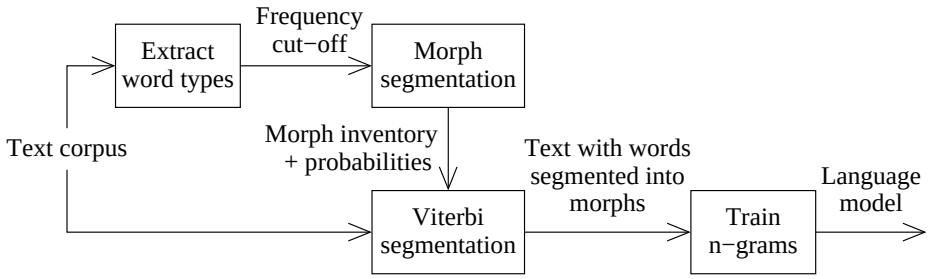


Figure 4.3: Experiment II: From a text corpus to an n-gram language model based on statistical morphs.

has also been improved compared to Experiment I (Pylkkönen and Kurimo, 2004; Juang et al., 1985).

Statistical Morphs

The so-called statistical morphs have been produced using the Morfessor Baseline algorithm. Morph inventories of different size have been generated and compared. The generation process is depicted in Figure 4.3. Morfessor is trained on word types (distinct word forms), which occur in the large text corpus. Morph inventories of different size can be obtained by setting different frequency thresholds (cut-offs) on the words that Morfessor is trained on. The resulting morphs (i.e., word fragments) are then utilized for the segmentation of the whole corpus, using the Viterbi algorithm. When word forms occurring less than three times in the corpus are excluded, Morfessor generates a morph inventory consisting of 26 000 morphs. Interestingly, this small fragment set is not outperformed by larger sets, when applied in n-gram language modeling.

Grammatical Morphs

The Hutmegs gold standard (see 4.1.2) provides a linguistic, or grammatical, morph segmentation for the words in the corpus. However, not all word forms are available in the gold standard. Such OOV words make up 4.2% of all the words in the training corpus, and 0.3% and 3.8% of the words in the two test sets (Book and News respectively).

In contrast to Experiment I, OOV's are not excluded in the current experiment (Experiment II). Instead, OOV words are split into individual phonemes, which makes it possible to construct any word form by a concatenation of phonemes. Table 4.3 provides an example: The place name "Pyynikki" has not been recognized by the morphological analyzer, on which the Hutmegs gold standard relies. Therefore, "Pyynikillä" (in Pyynikki) has been split into individual phonemes in the grammatical morph model.

The absence of out-of-vocabulary words makes it possible to compare language model entropies across all fragment inventories (unlike the Words-OOV model in Experiment I). Since *any* possible phoneme string obtains some probability in each model, one

could talk about *unlimited-vocabulary* speech recognition. The utility of such an OOV-splitting approach naturally requires that it does not perform *worse* than the standard method of substituting OOV's with a special OOV symbol. A speech recognition experiment carried out in Publication 7 suggests that this is not the case: Both approaches performed on an equal level.

Words

The SRI toolkit makes it possible to experiment with large recognition lexicons. A lexicon was constructed, containing each word form occurring four times or more in the text corpus. The number of such words amounts to 410 000. The remainder of the words were split into individual phonemes, and the phonemes were added to the lexicon as entries of their own. Note that even if such a large lexicon is used, 5.0 % of all word tokens in the training corpus are OOV and need to be split. The OOV rate of the test sets Book and News are 7.3 % and 5.0 %, respectively.[13]

Word breaks of split words have been modeled in such a manner that there are two variants of each phoneme, one for occurrences in the beginning or middle of a word and one for occurrences at the end of a word. Each unsplit word is assumed implicitly to end in a word break. This is illustrated in Table 4.3, where the rare word form "tuoremehuasema" (juice factory) has been split by the Word 410k model, whereas the other words are contained in the lexicon.

Results

Cross-entropies and word error rates for the three models on the two test sets are plotted in Figures 4.4 and 4.5. (The corresponding numerical values can be found in Appendix A.5.2.) On a whole, the statistical morph model performs the best, although the word model does achieve as good or slightly better levels of entropy. However, the word model arrives at its lowest entropy level with a considerably higher memory consumption than the statistical morph model. Moreover, in speech recognition the word model is outperformed by the statistical morph model.

Unfortunately, there are no word error measurements for n-grams of higher order than 5. Thus, the level that the word model would eventually attain is unknown. Nevertheless, if one allows oneself to extrapolate the entropies of the 7-gram word models onto word error rates, in accordance with Goodman's hypothesized linear relationship between entropy and word error rate (see Section 2.3.3), the statistical morph model would still outperform the word model by a good margin on both test sets.

The grammatical morphs perform as well as the statistical morphs on the book data, but they perform worse in the news task. The news text contains a large number of names that are unknown to the linguistic gold standard. These OOV words need to be constructed phoneme by phoneme by the speech recognizer, which is a very error-prone process. The grammatical morph model encounters more OOV words in the news data

[13] It is desirable that the OOV rate of a large vocabulary speech recognition system is below 1 % (personal communication with Andreas Stolcke).

than the book data. For the word model, the opposite is true. This explains the speech recognition performance of these two models on the two test sets.

The error rates of the morph models do not seem to decrease, when higher order n-grams are used. This may be explained by the relatively high overall error rate. If speech recognition errors are frequent, long n-gram histories are very likely to contain errors. Predictions based on such erroneous histories may be worse than predictions based on shorter, less erroneous, histories.

An example of actual output from the speech recognizer is shown in Table 4.5. For comparison, the recognized text is aligned with the correct transcription. Most recognition errors consist in single phonemes being recognized wrongly and compound words written apart.

4.2.5 Related Work

Different approaches have been proposed for dealing with the problem of vocabulary growth in large vocabulary speech recognition. Geutner et al. (1998) present a two-pass recognition approach, where the vocabulary is augmented adaptively. In the first pass, a word lattice is created using on a traditional word vocabulary. In the second pass, inflectional forms of the words in the lattice are added. In the recognition of Serbo-Croatian, word accuracy improvement from 64.0 % to 69.8 % are reported. McTait and Adda-Decker (2003) simply advocate the use of large word vocabularies. In a German task, the use of a lexicon of 300 000 instead of 60 000 words lowered the word error rate from 20.4 % to 18.5 %.

Factored language models (FLM) (Bilmes and Kirchhoff, 2003) have been proposed for the incorporation of morphological knowledge in the modeling of inflecting languages. Rather than conditioning the probability of a word on a few of the words preceding it, the probabilities are conditioned on sets of features of words. Minor word error rate reductions are reported in the recognition of Arabic speech, when morphological knowledge was utilized in an FLM (Kirchhoff et al., 2003; Vergyri et al., 2004).

Several researchers abandon the word as a basic unit of language modeling and speech recognition. Words are split into sub-word fragments, which typically correspond to grammatical affixes and stems. Solutions have been proposed for different languages, and perplexity reductions have been achieved, but few have reported clear word error reductions. Byrne et al. (2000) apply a morphological analyzer to Czech in order to split words into stems and endings. A language model based on a vocabulary of 9 600 morphemes produces better results than a model based on a vocabulary of 20 000 words. However, with larger vocabularies (61 000 words and 25 000 morphemes), the word based models perform better (Byrne et al., 2001). Kwon and Park (2003) also use a morphological analyzer to obtain morphemes in a Korean recognition task. Merging short morphemes improves the results in their experiments. Szarvas and Furui (2003) compare a plain morpheme n-gram model to a combination of morpheme n-grams and a stochastic model of morphotactics, which prevents ungrammatical stem + ending combinations. Relative morpheme error reductions between 1.7% and 7.2% were obtained in their experiments with Hungarian. Arisoy and Arslan (2005) do not achieve im-

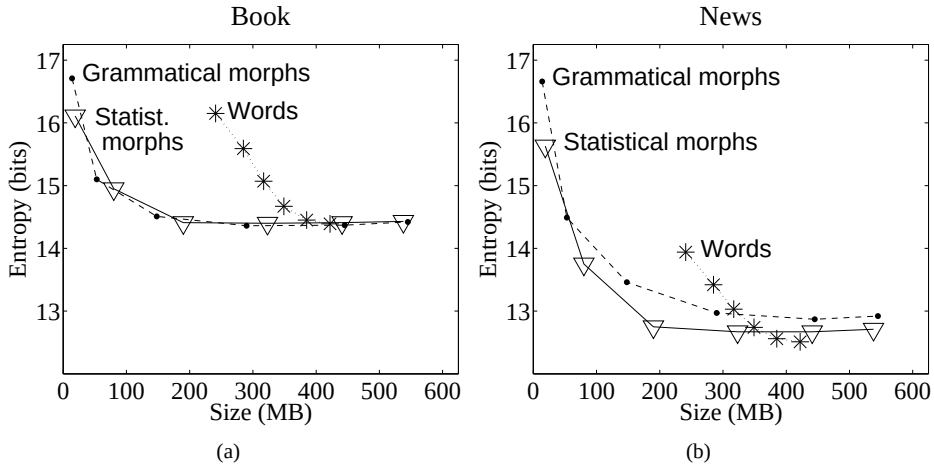


Figure 4.4: Experiment II: Cross-entropies, normalized over the number of words in the test set, plotted against language model size. Three types of language models are compared on two types of data: book and news text. The six points along each curve represent the orders 2–7 of the n-gram models (from left to right). Low entropy values correspond, on average, to a good capacity of predicting the following word in a sequence.

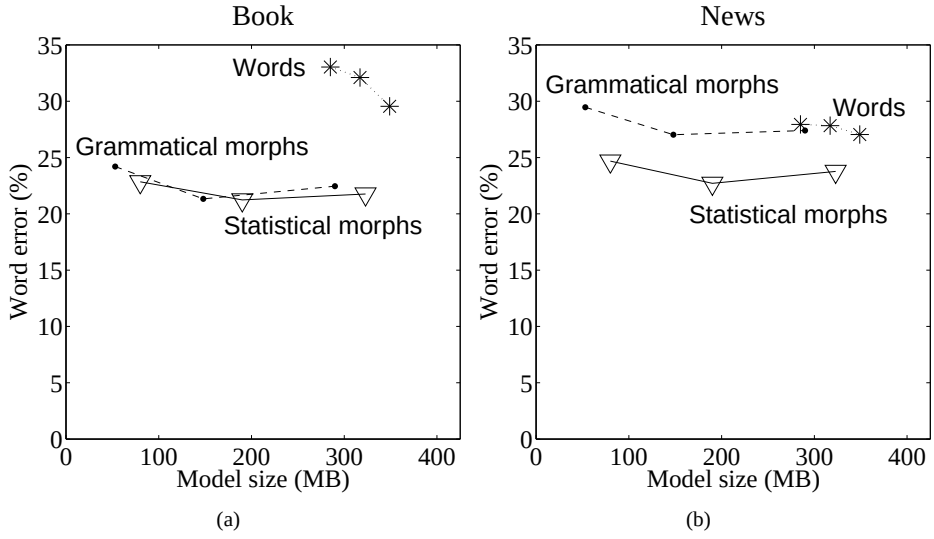


Figure 4.5: Experiment II: Word error rates in speech recognition plotted against language model size. Three types of language models are tested on two types of data. There are three points on each curve, corresponding to the n-gram orders 3–5 (from left to right). Higher order n-grams were not tested, due to their very high memory requirements.

Table 4.5: Example output from the speech recognizer: Read news text has been recognized using a 4-gram language model based on statistical morphs. Morph boundaries have been marked using small dots (·). Neither capitalization nor sentence boundaries are recognized by the current language model, but in order to facilitate the reading, each end of sentence has been marked with a slash (/). Recognition errors have been underlined. In two cases the errors are due to the speaker's stumbling over her words ("ydin*osta*. . ." on line 9 and "*kieli*" on the last line). An interesting mistake is shown on line 17: The phrase "ettei Irak ollut" (that Iraq had not) has been recognized as the grammatically correct "että Irak oli" (that Iraq had). Thus, the language model has ensured the required grammatical agreement: "että . . . oli" vs. "ettei . . . ollut".

	Recognized	Correct
1	uu·ossa ei usko i·rak iskuun /	usa ei usko irak-iskuun /
2	amerikkalais·viranomaiset pitävä·t	amerikkalaisviranomaiset pitävät
3	hyvin epätodennäkö·isen·ä että	hyvin epätodennäköisenä että
4	yhdys·valla·t hyökkä·si ase·in	yhdysvallat hyökkäisi asein
5	iraki·a vastaan lähi·aikoina /	irakia vastaan lähiaikoina /
6	viranomaiset arvioi·vat perjantai·na	viranomaiset arvioivat perjantaina
7	että sana·llinen viesti rii·ttää	että sanallinen viesti riittää
8	ja iraki·n johto paljas·taa pia·n	ja irakin johto paljastaa pian
9	lopu·t·kin ydin·osta·laitteisto·ssa /	loputkin ydinlaitteistonsa /
10	y·k on turvallisuus·neuvosto·n viisi	yk:n turvallisuusneuvoston viisi
11	pysy·vää jäsen·tä oli·vat pyytä·neet	pysyvää jäsentä olivat pyytäneet
12	itä ja paista·maan ydinase·iden	irakia paljastamaan ydinaseiden
13	valmista·miseen tähtää·vän	valmistamiseen tähtäävän
14	ohjelma·ssa kokonais·uudessaan	ohjelmansa kokonaisuudessaan
15	torstai·hin menne·ssä /	torstaihin mennessä /
16	valko·inen talo il·moitti kui·t·enkin	valkoinen talo ilmoitti kuitenkin
17	torstai·na että i·rak oli tä·yttänyt	torstaina ettei irak ollut täyttänyt
18	vaatteita /	vaateita /
19	amerikkalais·edustajien mukaan	amerikkalaisedustajien mukaan
20	sotilaa·lliselle hyökkäykse·lle ei ole	sotilaalliselle hyökkäykselle ei ole
21	tarvetta n·iin kau·a·n kun i·rak	tarvetta niin kauan kun irak
22	jat·kaa ydin·teknologia·ssa	jatkaa ydinteknologiansa
23	paljasta·mista /	paljastamista /
24	erä·s edustaja tot·esi ette·i	eräs edustaja totesi ettei
25	ase·llisella hyökkäykse·llä saatta·isi	aseellisella hyökkäyksellä saataisi
26	kui·te·nkaan tuho·ttua kaikki·a	kuitenkaan tuhottua kaikkea
27	kieli iraki·n ydin·materiaali·a /	irakin ydinmateriaalia /

provements when utilizing morphemes instead of words in a Turkish news broadcast recognition task, but a weakness in their experimental setup consists in the use of the same low n-gram order (bigram) for words and morphemes alike.

In addition to using existing morphological analyzers, purely data-driven algorithms for splitting words into smaller units have been explored in speech recognition. Whittaker and Woodland (2000) propose an algorithm for segmenting a text corpus into fragments that maximize the bigram likelihood of the segmented corpus. Small improvements in error rates (2.2 % relative) were obtained in an English recognition task when the sub-word model was interpolated with a traditional word-based trigram model. Ordelman et al. (2003) present a method for decomposing Dutch compound words automatically, and reports minor improvements in error rates. Hacioglu et al. (2003) use the Morfessor Baseline algorithm combined with some postprocessing in a Turkish recognition task and achieve a 20 % relative reduction of the word error rate in comparison to word-based recognition.

There is little previous work on Finnish speech recognition based on sub-word units. Kneissler and Klakow (2001) segmented a corpus into word fragments that maximize the unigram likelihood of the corpus. Four different segmentation strategies are compared in a Finnish dictation task. The strategies require various amounts of input from an expert of the Finnish language. However, no comparisons to traditional word models are performed.

Morpheme-like units have also been discovered when word segmentation algorithms have been applied to transcribed speech. Deligne and Bimbot (1997) derive a model structure that can be used both for word segmentation and for detecting variable-length acoustic units in speech data. Their data-driven units do not, however, produce as good results as conventional word models in recognizing the speech of French weather forecasts. Brent (1999) and Venkataraman (2001) are interested in the acquisition of a lexicon in an incremental fashion and apply probabilistic models to the segmentation of transcripts of child-directed speech.

Chapter 5

Conclusions and Future Directions

There are regularities in word forming which can be discovered using unsupervised machine learning techniques. This observation is not new, but there has been a lamentable lack of solutions suitable for highly-inflecting and compounding languages, where the average number of morphemes per word is high. The Morfessor model proposed in this work can handle words consisting of lengthy sequences of morphemes. The segmentations produced by Morfessor for Finnish and English words match a linguistic morphological segmentation well; in this task a widely used benchmark algorithm is outperformed by Morfessor. Furthermore, the use of Morfessor as a basis for language modeling in Finnish large-vocabulary speech recognition has turned out beneficial in comparison to more standard approaches.

As part of the work, Finnish and English linguistic segmentations have been produced as a basis for a quantitative assessment of the segmentation accuracy of Morfessor. In addition, Turkish and Egyptian Arabic evaluation data have recently become available. Some experiments have been carried out, and the results are comparable to those obtained for Finnish and English. Morfessor has also been tested on smaller data sets of Swedish, Russian, Estonian, as well as other languages, but a quantitative evaluation has not been possible. Visual inspection suggests that also here the results are on the same level as for Finnish and English. In future research, Morfessor should be applied more systematically to a larger collection of languages.

In addition to new languages, Morfessor can be applied in new natural language processing tasks. Currently, there are results from automatic speech recognition and information retrieval. Hopefully, Morfessor can prove itself useful in machine translation as well. Experiments can freely be carried out, since a downloadable software package is publicly available (Creutz and Lagus, 2005). However, the spread of Morfessor would be promoted if the program code for the Categories models were available in addition to the current package that only supports the Baseline versions.

The future development of Morfessor could focus on at least two important issues:

non-concatenative phenomena and the utilization of a larger context, which extends across word boundaries. Non-concatenative processes are common in most languages, e.g., sound changes within stems as in English "sing, s*a*ng, s*u*ng", German "Mann, M*ä*nn+er", and Finnish "käsi, kä*de*+n, kä*t*+tä". Such regular patterns could be learned in an unsupervised manner from language data, similarly to the learning of a segmentation. Modeling sound change is a necessary step toward a full model of allomorphy, i.e., a model that identifies complementary distributions of morphs that are realizations of the same underlying morpheme.

Currently Morfessor examines words in isolation. If word *sequences* were utilized instead, larger idiomatic segments could be discovered, such as multi-word geographical names: "San Francisco, New York City, New Zealand". Additionally, the use of cross-word contexts is necessary, if one wants to discover syntactic processes, such as the grammatical agreement of number in the phrases "this result shows" (singular) vs. "these results show" (plural).

A larger context window naturally contains more information, which means that a larger set of usage-based features could be extracted. A finer resolution of the input space makes it possible to acquire more fine-grained category distinctions. The current separation of morphs into three broad categories (prefix, stem, and suffix) is sufficient in some, but not all situations. Distinguishing between verb endings and nominal endings is an example of one, crude yet possible refinement.

So far, all information about the usage of a morph is based on its occurrences in text. If one were to produce richer semantic representations, one could *ground morph meaning* in additional modalities, by using *multimodal data* as input (e.g., image, sound, tactile sensations). Moreover, the current model family assumes the existence of distinct (albeit probabilistic) categories. Continuous latent representations would be an alternative, and one might draw inspiration from the conceptual spaces framework proposed by Gärdenfors (2000).

Regardless of how far one progresses from the current state of morphology learning toward a more advanced language acquiring system, there seem to be no compelling reasons to abandon the current modeling methodology. Probabilistic modeling provides a means for ranking competing hypotheses in a principled manner. The Bayesian framework, as well as the related Minimum Description Length and Minimum Message Length formalisms, additionally incorporate a mechanism for controlling model complexity. It has been suggested, e.g., by Kit (2003), that the functioning of the human brain is likely to rely on resembling general principles.

Appendices

A.1 Derivation of a Noninformative Prior for Morph Frequency

Suppose that there are a total number of N morph tokens in the segmented corpus and that these morphs represent M different morph types. What is the probability of coming up with a particular frequency distribution, i.e., a set of M frequencies that sum up to N?

Further suppose that the probability distribution is a noninformative prior, that is, all frequency distributions are equally likely. It follows that the probability of one particular distribution is one divided by the number of possible ways of choosing M positive integers (the M frequencies) that sum up to N.

Imagine that the N morph tokens are sorted into alphabetical order and each morph is represented by a binary digit. Since some morphs occur more than once, there will be sequences of several identical morphs in a row. Now, initialize all N bits to zero. Next, every location, where the morph *changes*, is switched to a one, whereas every location, where the morph is identical to the previous morph, is left untouched. There are $\binom{N}{M}$ possibilities of choosing M bits to switch in a string of N bits. However, as the value of the first bit is known to be one, it can be omitted, which leaves us with $\binom{N-1}{M-1}$ possible binary strings. Therefore the probability of the frequency distribution is:

$$P(\textit{frequency distribution}) = 1/\binom{N-1}{M-1} = \frac{(M-1)!(N-M)!}{(N-1)!}. \tag{A.1}$$

A.2 Derivation of a Zipfian Frequency Prior

Zipf has studied the relationship between the frequency of a word, f, and its rank, z. The rank of a word is the position of the word in a list, where the words have been sorted according to falling frequency. Zipf suggests that the frequency of a word is inversely proportional to its rank (Zipf, 1935). Mandelbrot has refined Zipf's formula,

and suggests a more general relationship; see, e.g., Baayen (2001):

$$f = C(z + b)^{-a}, \tag{A.2}$$

where C, a and b are parameters of a text.

Let us derive a probability distribution from Mandelbrot's formula. The rank of a word as a function of its frequency can be obtained by solving for z from Eq. A.2:

$$z = C^{\frac{1}{a}} f^{-\frac{1}{a}} - b. \tag{A.3}$$

Suppose that one wants to know the number of words that have frequency f rather than the rank of a word with frequency f. We denote this *frequency of frequency* f by $n(f)$. An estimate for $n(f)$ is obtained as the difference in rank between a word with frequency f and a word with frequency $f + 1$:

$$n(f) = z(f) - z(f + 1) = C^{\frac{1}{a}} \left(f^{-\frac{1}{a}} - (f + 1)^{-\frac{1}{a}}\right). \tag{A.4}$$

A probability distribution for f is obtained by dividing $n(f)$ by the total number of word tokens, which equals the sum of frequencies over all possible frequencies. The highest frequency is denoted by F. Thus,

$$P(f) = \frac{n(f)}{\sum_{f'=1}^{F} n(f')} = \frac{C^{\frac{1}{a}} \left(f^{-\frac{1}{a}} - (f + 1)^{-\frac{1}{a}}\right)}{\sum_{f'=1}^{F} C^{\frac{1}{a}} \left(f'^{-\frac{1}{a}} - (f' + 1)^{-\frac{1}{a}}\right)} = \frac{f^{-\frac{1}{a}} - (f + 1)^{-\frac{1}{a}}}{1 - (F + 1)^{-\frac{1}{a}}}. \tag{A.5}$$

When the highest frequency F is assumed to be big, one can make the approximation $F \approx \infty$ without any loss of accuracy that is of practical significance:

$$P(f) \approx \lim_{F \to \infty} \frac{f^{-\frac{1}{a}} - (f + 1)^{-\frac{1}{a}}}{1 - (F + 1)^{-\frac{1}{a}}} = f^{-\frac{1}{a}} - (f + 1)^{-\frac{1}{a}}. \tag{A.6}$$

Rather than setting a value for the parameter a, we want to shape the probability distribution according to our prior belief of the proportion of *hapax legomena* (h), i.e., the proportion of words occurring only once in the corpus:

$$h = P(1) = 1^{-\frac{1}{a}} - 2^{-\frac{1}{a}} = 1 - \left(\frac{1}{2}\right)^{\frac{1}{a}}. \tag{A.7}$$

Substituting a in Eq. A.6 by h yields:

$$P(f) = f^{\log_2(1-h)} - (f+1)^{\log_2(1-h)}. \tag{A.8}$$

The exponent $\log_2(1-h)$ is always negative. Therefore the resulting probability distribution follows a *power law* and it is represented by a straight line when plotted in a graph with logarithmic scales on both axes. It can be assumed that the derived probability distribution applies to morphs as well as to words.

A.3 Probability of the Frequency of Individual Morphs

It is difficult to compare the implications of the Zipfian frequency prior in Eq. A.8 to those of the noninformative prior in Eq. A.1. The Zipfian prior separately assigns a probability to the frequency of each morph, whereas the noninformative prior at once assigns a probability for the whole frequency distribution. In the following, an approximation will be derived for the probability of the frequency of an individual morph in the noninformative prior scheme. This facilitates a comparison between the Zipfian and noninformative prior approaches.

Suppose that there are N morph tokens and M morph types. Next, f occurrences of a new morph are added, which increases the number of morph tokens to $N+f$ and the number of morph types to $M+1$. We compute the conditional probability of adding a morph with frequency f when the initial position (N, M) is given:

$$P(f \mid N, M) = \frac{P(f, N, M)}{P(N, M)} = \frac{P(\textit{freq. distr.}(N+f, M+1))}{P(\textit{freq. distr.}(N, M))}. \tag{A.9}$$

According to Eq. A.1 this equals:

$$P(f \mid N, M) = \binom{N-1}{M-1} \Big/ \binom{N+f-1}{M} = \frac{(N-1)!M!(N-M+f-1)!}{(N+f-1)!(M-1)!(N-M)!}. \tag{A.10}$$

The factorials are rewritten using Stirling's approximation: $n! \approx (n/e)^n \sqrt{2\pi n}$:

$$P(f \mid N, M) = \frac{(N-1)^{N-1/2} M^{M+1/2} (N-M+f-1)^{N-M+f-1/2}}{(N+f-1)^{N+f-1/2} (M-1)^{M-1/2} (N-M)^{N-M+1/2}}. \tag{A.11}$$

The factors that are constant with respect to f are rewritten as C_1:

$$P(f \mid N, M) = C_1 \cdot \frac{(N-M+f-1)^{N-M+f-1/2}}{(N+f-1)^{N+f-1/2}}. \tag{A.12}$$

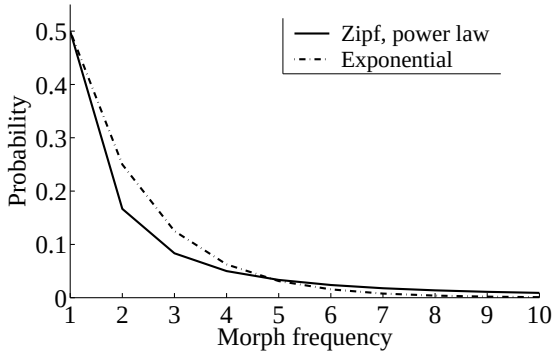


Figure A.1. Probabilities of morph frequencies according to (1) a pdf derived from Mandelbrot's correction of Zipf's law ($h = 0.5$) and (2) an approximately exponential pdf resulting from applying the non-informative frequency prior ($N = 10\,000, M = 5000$).

For f values that are much smaller than N and M the following approximately holds for the bases: $N - M + f - 1 \approx N - M$ and $N + f - 1 \approx N$. Thus,

$$P(f \mid N, M) = C_1 \cdot \frac{(N-M)^{N-M+f-1/2}}{N^{N+f-1/2}} = C_1 \cdot \frac{(N-M)^{N-M-1/2}(N-M)^f}{N^{N-1/2}N^f}. \tag{A.13}$$

The factors that are now constant with respect to f are combined with C_1 into C_2:

$$P(f \mid N, M) = C_2 \cdot \frac{(N-M)^f}{N^f} = C_2 \cdot \big(\frac{N-M}{N}\big)^f. \tag{A.14}$$

This results in an *exponential distribution*. That is, the probability decreases exponentially with the value of the frequency. (This only applies to f values that are small compared to the total number of tokens N and types M.) The exponential distribution can be directly compared to the power-law distribution that results from applying the Zipfian prior in the previous section. Figure A.1 shows that the curves are different, but not radically different for small frequency values, which may explain why neither approach performs significantly better than the other.

A.4 Hierarchical Dependencies for Morph Category Membership Probabilities

In the Categories-MAP model, a morph can have hierarchical structure. Section 3.3.5 describes the normal computation of category membership probabilities, such as $P(\text{PRE} \mid \mu_i)$, which denotes the probability that the morph μ_i functions as a prefix.

However, if a morph consists of submorphs, its category membership probabilities are affected by the category tagging of the submorphs. This prevents conflicts between the syntactic role of a morph itself and its substructure. The following rules apply:

1. If either submorph has been tagged as a non-morpheme, *none of the following dependencies apply*, because non-morphemes are considered as mere sound patterns without a syntactic (or semantic) function.

2. Stems need to consist of at least one (sub)stem: PRE + STM, STM + STM, or STM + SUF. Otherwise the probability of a stem is zero, that is, when both submorphs are are either prefixes or suffixes.

3. Suffixes can only consist of other suffixes and have zero probability otherwise. A morph μ consisting of two suffixes $\mu_1 + \mu_2$ is as likely to be a suffix as the one of its submorphs with lower suffix probability, i.e., $P(\text{SUF} \mid \mu) = \min\{P(\text{SUF} \mid \mu_1), P(\text{SUF} \mid \mu_2)\}$. The probability of μ being a non-morpheme is then decreased to $1 - P(\text{SUF} \mid \mu)$, if it should be higher than this value. (According to the other rules, prefixes and stems have zero probability in this case.)

4. Prefixes are treated analogously to the suffixes.

The above rules follow straightforwardly from the "semantics" of the morpheme categories. The second rule might be superfluous, but it seems natural that a sequence of morphs, all tagged as prefixes or suffixes, should not function as a stem when occurring together (e.g., English "ing/SUF + ly/SUF", occurring in words, such as "knowingly"). In this case, "ingly" should most logically be tagged as a suffix. However, the left perplexity of "ingly" is rather low, because as a whole it occurs in much fewer context than its submorphs. The third rule gives morph sequences, such as "ingly", a fair chance of being tagged as suffixes, even if their own left perplexity is low, but their constituents are good suffix candidates.

A.5 Experimental Results in Numbers

This section contains all numbers related to the experimental results presented in graphical form in Chapter 4.

A.5.1 Linguistic Evaluation: Morpheme Segmentation Performance

Tables A.1–A.6 contain the numerical values for the results of the experiments described in Section 4.1.3.

Finnish

Tables A.1–A.3 report the precision, recall, and F-measure of the placement of morpheme boundaries, as evaluated against the Finnish Hutmegs gold standard (see Sec. 4.1.3). Five algorithms have been tested on Finnish test sets of four different sizes (10 000 – 16 million words). The reported figures are means and standard deviations (std) from runs on five different sets of the same size, with the exception of the largest set (one run).

Table A.1: Precision of the placement of morpheme boundaries: five algorithms evaluated on Finnish data sets of four different sizes.

FINNISH:	10 000		50 000		250 000		16 000 000	
PRECISION [%]	mean	std	mean	std	mean	std	mean	std
Baseline	59.24	1.79	64.63	2.04	71.41	1.59	85.46	–
Baseline-Length	66.65	2.04	70.49	1.80	75.10	1.28	84.92	–
Categories-ML	70.00	1.96	70.89	1.46	73.15	1.22	75.08	–
Categories-MAP	68.25	2.58	70.46	1.40	73.66	0.81	77.35	–
Linguistica	79.53	0.59	77.96	0.58	72.64	0.53	–	–

Table A.2: Recall of the placement of morpheme boundaries: five algorithms evaluated on Finnish data sets of four different sizes.

FINNISH:	10 000		50 000		250 000		16 000 000	
RECALL [%]	mean	std	mean	std	mean	std	mean	std
Baseline	52.31	1.40	51.62	1.31	49.97	1.05	41.82	–
Baseline-Length	49.85	2.27	50.44	1.02	49.06	1.09	42.95	–
Categories-ML	51.69	2.66	59.45	0.51	61.57	1.02	67.57	–
Categories-MAP	52.09	1.92	57.58	0.83	60.92	0.49	62.21	–
Linguistica	25.80	2.88	38.51	1.66	37.25	1.43	–	–

Table A.3: F-measure of the placement of morpheme boundaries: five algorithms evaluated on Finnish data sets of four different sizes.

FINNISH:	10 000		50 000		250 000		16 000 000	
F-MEASURE [%]	mean	std	mean	std	mean	std	mean	std
Baseline	55.54	1.11	57.36	0.69	58.77	0.31	56.16	–
Baseline-Length	57.03	2.10	58.78	0.69	59.32	0.57	57.05	–
Categories-ML	59.46	2.43	64.66	0.69	66.85	0.72	71.13	–
Categories-MAP	59.05	1.80	63.36	0.82	66.68	0.41	68.96	–
Linguistica	38.88	3.37	51.54	1.54	49.23	1.34	–	–

English

Tables A.4–A.6 report the precision, recall, and F-measure of the placement of morpheme boundaries, as evaluated against the English Hutmegs gold standard (see Sec. 4.1.3). Five algorithms have been tested on English test sets of four different sizes (10 000 – 12 million words). The reported figures are means and standard deviations (std) from runs on five different sets of the same size, with the exception of the largest set (one run).

Table A.4: Precision of the placement of morpheme boundaries: five algorithms evaluated on English data sets of four different sizes.

ENGLISH:	10 000		50 000		250 000		12 000 000	
PRECISION [%]	mean	std	mean	std	mean	std	mean	std
Baseline	44.38	4.88	48.58	2.17	53.80	0.68	63.28	–
Baseline-Length	60.60	3.34	63.81	2.26	65.53	0.87	65.76	–
Categories-ML	73.70	4.50	79.41	2.62	80.28	2.55	73.42	–
Categories-MAP	70.24	4.33	74.22	2.36	76.30	1.82	84.00	–
Linguistica	89.49	3.38	82.91	2.64	77.85	1.87	68.32	–

Table A.5: Recall of the placement of morpheme boundaries: five algorithms evaluated on English data sets of four different sizes.

ENGLISH:	10 000		50 000		250 000		12 000 000	
RECALL [%]	mean	std	mean	std	mean	std	mean	std
Baseline	71.98	4.68	75.74	4.17	75.76	2.22	71.13	–
Baseline-Length	67.28	2.97	71.76	2.81	74.17	1.72	69.65	–
Categories-ML	60.00	2.38	64.75	2.43	66.23	1.29	68.09	–
Categories-MAP	59.67	1.09	65.58	2.44	66.51	1.58	55.68	–
Linguistica	46.92	2.33	61.13	2.52	66.56	2.43	62.79	–

Table A.6: F-measure of the placement of morpheme boundaries: five algorithms evaluated on English data sets of four different sizes.

ENGLISH:	10 000		50 000		250 000		12 000 000	
F-MEASURE [%]	mean	std	mean	std	mean	std	mean	std
Baseline	54.56	2.98	59.08	1.12	62.90	0.62	66.97	–
Baseline-Length	63.72	2.77	67.50	1.71	69.57	0.86	67.65	–
Categories-ML	66.07	2.31	71.32	2.32	72.57	1.54	70.65	–
Categories-MAP	64.48	2.15	69.59	1.78	71.05	1.18	66.97	–
Linguistica	61.56	2.79	70.35	2.38	71.76	2.17	65.44	–

A.5.2 Language Modeling and Speech Recognition: Experiment II

The results from the second speech recognition experiment (Section 4.2.4) are collected into Tables A.7 (Book task) and Table A.8 (News task).

Table A.7: Comparison of three different types of language models in the recognition of a Finnish audio book (see Section 4.2.4). For each order of the n-gram model, the size of the model, together with the cross-entropy (H) as well as the word-error rate (WER) are reported (WER only available for $n = 3, 4, 5$).

	Statistical morphs			Grammatical morphs			Words		
n	size [MB]	H [bits]	WER [%]	size [MB]	H [bits]	WER [%]	size [MB]	H [bits]	WER [%]
2	19	16.11	–	14	16.71	–	241	16.15	–
3	80	14.95	22.85	53	15.10	24.20	285	15.59	33.04
4	190	14.41	21.24	148	14.51	21.33	317	15.07	32.11
5	323	14.40	21.76	290	14.36	22.46	349	14.67	29.55
6	441	14.41	–	445	14.37	–	385	14.45	–
7	538	14.43	–	545	14.42	–	422	14.39	–

Table A.8: Comparison of three different types of language models in the recognition of Finnish news broadcasts (see Section 4.2.4). For each order of the n-gram model, the size of the model, together with the cross-entropy (H) as well as the word-error rate (WER) are reported (WER only available for $n = 3, 4, 5$).

	Statistical morphs			Grammatical morphs			Words		
n	size [MB]	H [bits]	WER [%]	size [MB]	H [bits]	WER [%]	size [MB]	H [bits]	WER [%]
2	19	15.63	–	14	16.66	–	241	13.94	–
3	80	13.75	24.68	53	14.49	29.47	285	13.42	27.94
4	190	12.75	22.71	148	13.46	27.03	317	13.03	27.83
5	323	12.67	23.77	290	12.97	27.40	349	12.74	27.05
6	441	12.67	–	445	12.87	–	385	12.56	–
7	538	12.71	–	545	12.92	–	422	12.51	–

Bibliography

Adda-Decker, M. (2003). A corpus-based decompounding algorithm for German lexical modeling in LVCSR. In *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech)*, pages 257–260, Geneva, Switzerland.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723.

Allen, M., Badecker, W., and Osterhout, L. (2003). Morphological analysis in sentence processing: An ERP study. *Language and Cognitive Processes*, 18(4):405–430.

Altun, Y. and Johnson, M. (2001). Inducing SFA with ϵ-transitions using Minimum Description Length. In *Proc. Finite-State Methods in Natural Language Processing, ESSLLI Workshop*, Helsinki.

Anderson, S. R. (1992). *A-Morphous Morphology*. Cambridge University Press.

Ando, R. K. and Lee, L. (2000). Mostly-unsupervised statistical segmentation of Japanese: Applications to Kanji. In *Proc. 6th Applied Natural Language Processing Conference and 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL)*, pages 241–248.

Argamon, S., Akiva, N., Amir, A., and Kapah, O. (2004). Efficient unsupervised recursive word segmentation using Minimum Description Length. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*.

Arisoy, E. and Arslan, L. M. (2005). Turkish dictation system for broadcast news applications. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO 2005)*, Antalya, Turkey.

Baayen, R. H. (2001). *Word Frequency Distributions*. Kluwer Academic Publishers.

Baayen, R. H., Piepenbrock, R., and Gulikers, L. (1995). *The CELEX lexical database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC96L14`.

Baayen, R. H. and Schreuder, R. (2000). Towards a psycholinguistic computational model for morphological parsing. *Philosophical Transactions of the Royal Society (Series A: Mathematical, Physical and Engineering Sciences 358)*, pages 1–13.

Banerjee, S. and Rosenfeld, A. (1992). MAP estimation of context-free grammars. *Pattern Recognition Letters*, 13(2):95–101.

Baroni, M. (2003). Distribution-driven morpheme discovery: A computational/experimental study. *Yearbook of Morphology 2003*, pages 213–248.

Baroni, M., Matiasek, J., and Trost, H. (2002). Unsupervised learning of morphologically related words based on orthographic and semantic similarity. In *Proc. Workshop on Morphological & Phonological Learning of ACL'02*, pages 48–57.

Barron, A., Rissanen, J., and Yu, B. (1998). The Minimum Description Length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Bellegarda, J. R. (2000). Exploiting Latent Semantic Analysis in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. In *Proceedings of Neural Information Processing Systems (NIPS 2000)*, pages 932–938.

Bilmes, J. A. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference (HLT/NAACL)*, pages 4–6, Edmonton, Canada.

Bingham, E. (2003). *Advances in Independent Component Analysis with Applications to Data Mining*. PhD thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Computer and Information Science.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pages 245–250, San Francisco, CA, USA.

Bloomfield, L. (1933). *Language*. George Allen & Unwin Ltd, London.

Bordag, S. (2005). Unsupervised knowledge-free morpheme boundary detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2005)*, Borovets, Bulgaria.

Brent, M. R. (1999). An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

Brent, M. R. and Cartwright, T. A. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1–2):93–125.

Broman, S. and Kurimo, M. (2005). Methods for combining language models in speech recognition. In *Proceedings of Interspeech 2005*, pages 1317–1320, Lisbon, Portugal.

Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Byrne, W., Hajič, J., Ircing, P., Jelinek, F., Khudanpur, S., Krbec, P., and Psutka, J. (2001). On large vocabulary continuous speech recognition of highly inflectional language — Czech. In *Proceedings of EUROSPEECH 2001*, pages 487–489, Aalborg, Denmark.

Byrne, W. J., Hajič, J., Krbec, P., Ircing, P., and Psutka, J. (2000). Morpheme based language models for speech recognition of Czech. In *Proceedings of the Third International Workshop on Text, Speech, Dialogue (TSD-2000)*, pages 211–216.

Carlson, L. (2005). Inducing a morphological transducer from inflectional paradigms. In Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., and Yli-Jyrä, A., editors, *Inquires into Words, Constraints and Contexts. Festschrift in the Honour of Kimmo Koskenniemi on his 60th Birthday*, pages 18–24. Gummerus Printing, Saarijärvi, Finland.

Cetinoglu, O. (2000). Prolog based natural language processing infrastructure for Turkish. Master's thesis, Bogazici University, Istanbul, Turkey.

Chang, J.-S., Lin, Y.-C., and Su, K.-Y. (1995). Automatic construction of a Chinese electronic dictionary. In *Proc. Third workshop on very large corpora*, pages 107–120, Somerset, New Jersey.

Chelba, C. and Jelinek, F. (2000). Structured language modeling. *Computer Speech and Language*, 14:283–332.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. The M.I.T. Press.

Christiansen, M. H., Allen, J., and Seidenberg, M. S. (1998). Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, 13(2):221–268.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of EUROSPEECH'97*, pages 2707–2710, Rhodes, Greece. `http://mi.eng.cam.ac.uk/~prc14/toolkit.html`.

Cohen, P., Heeringa, B., and Adams, N. (2002). Unsupervised segmentation of categorical time series into episodes. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, pages 99–106.

Creutz, M. (2006). Morfessor in the Morpho Challenge. In Kurimo, M., Creutz, M., and Lagus, K., editors, *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy.

Creutz, M. and Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology.

Croft, W. and Cruse, D. A. (2004). *Cognitive Linguistics*. Cambridge University Press.

de Beaugrande, R. (1991). *Linguistic Theory: The Discourse of Fundamental Works*. Longman, London. http://beaugrande.bizland.com/LINGTHERLinguistic%20Theory%20Title.htm.

de Beaugrande, R. (2004). *A New Introduction To The Study Of Text And Discourse: Discursivism and Ecologism*. Published on the Internet. http://beaugrande.bizland.com/new_intro_to_study.htm.

de Marcken, C. G. (1996). *Unsupervised Language Acquisition*. PhD thesis, MIT.

de Saussure, F. (1916). *Cours de linguistique générale*. Éditions Payot & Rivages. With comments by Tullio de Mauro, 1967.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407.

Déjean, H. (1998). Morphemes as necessary concept for structures discovery from untagged corpora. In *Workshop on Paradigms and Grounding in Natural Language Learning*, pages 295–299, Adelaide.

Deligne, S. and Bimbot, F. (1997). Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23:223–241.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Dutagaci, H. (2002). Statistical language models for large vocabulary continuous speech recognition of Turkish. Master's thesis, Bogazici University, Istanbul, Turkey.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness – A Connectionist Perspective on Development*. The MIT Press.

Engström, S. (2005). Information retrieval using unsupervisedly segmented morphemes. Special assignment, Laboratory of Computer and Information Science, Helsinki University of Technology.

Erjavec, T. and Džeroski, S. (2004). Machine learning of morphosyntactic structure: Lemmatizing unknown Slovene words. *Applied Artificial Intelligence*, 18(1):17–41.

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Statistical learning theory: A primer. *International Journal of Computer Vision*, 38(1):9–13.

Feng, H., Chen, K., Kit, C., and Deng, X. (2004). Unsupervised segmentation of Chinese corpus using accessor variety. In *Proc. First International Joint Conference on Natural Language Processing (IJCNLP)*, pages 255–261, Sanya, Hainan. (extended abstract).

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. Reprinted 1968 in Palmer, F., editor, *Selected Papers of J. R. Firth*, Longman, Harlow.

Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Gärdenfors, P. (2000). *Conceptual Spaces*. MIT Press.

Gaussier, E. (1999). Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 24–30, University of Maryland.

Ge, X., Pratt, W., and Smyth, P. (1999). Discovering Chinese words from unsegmented text. In *Proc. SIGIR*, pages 271–272.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis*. Chapman & Hall/CRC.

Geutner, P., Finke, M., and Scheytt, P. (1998). Adaptive vocabularies for transcribing multilingual broadcast news. In *Proceedings of ICASSP*, volume 2, pages 925–928, Seattle, Washington.

Gibbs, P. (1996). What is Occam's razor? In *Usenet Physics FAQ*. Updated 1997 by Sugihara Hiroshi. `http//math.ucr.edu/home/baez/physics/General/occam.html`. Retrieved November 7, 2005.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *Trends in Cognitive Science*, 7(5):219–224.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Goldsmith, J. (2005). An algorithm for the unsupervised learning of morphology. Technical Report TR-2005-06, University of Chicago. `http://humfs1.uchicago.edu/~jagoldsm/Papers/Algorithm.pdf`.

Goldsmith, J. and Hu, Y. (2004). From signatures to finite state automata. In *Midwest Computational Linguistics Colloquium*, Bloomington, Indiana, USA.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3):237–264.

Goodman, J. T. (2001a). A bit of progress in language modeling. *Computer Speech and Language*, 15:403–434.

Goodman, J. T. (2001b). A bit of progress in language modeling, extended version. Technical Report MSR–TR–2001–72, Microsoft Research.

Grünwald, P., Myung, I. J., and Pitt, M. A. (2005). *Advances in Minimum Description Length: Theory and Applications*. MIT Press.

Hacioglu, K., Pellom, B., Ciloglu, T., Ozturk, O., Kurimo, M., and Creutz, M. (2003). On lexicon creation for Turkish LVCSR. In *Proc. Eurospeech'03*, pages 1165–1168, Geneva, Switzerland.

Hafer, M. A. and Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385.

Hagen, A. and Pellom, B. (2005). Data driven subword unit modeling for speech recognition and its application to interactive reading tutors. In *Proceedings of INTERSPEECH 2005*, pages 2757–2760, Lisbon, Portugal.

Hakulinen, L. (1979). *Suomen kielen rakenne ja kehitys (The structure and development of the Finnish language)*. Kustannus-Oy Otava, 4th edition.

Harris, Z. (1991). *A Theory of Language and Information, A Mathematical Approach*. Clarendon Press.

Harris, Z. S. (1951). *Methods in Structural Linguistics*. The University of Chicago Press.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(1):146–162.

Harris, Z. S. (1955). From phoneme to morpheme. *Language*, 31(2):190–222. Reprinted 1970 in *Papers in Structural and Transformational Linguistics*, Reidel Publishing Company, Dordrecht, Holland.

Harris, Z. S. (1967). Morpheme boundaries within words: Report on a computer test. *Transformations and Discourse Analysis Papers*, 73. Reprinted 1970 in *Papers in Structural and Transformational Linguistics*, Reidel Publishing Company, Dordrecht, Holland.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Inc., 2nd edition.

Hockett, C. F. (1954). Two models of grammatical description. *Word*, 10(1):210–234.

Hockett, C. F. (1958). *A Course in Modern Linguistics*. Macmillan Publishing Co., Inc.

Honkela, T. (1997). *Self-Organizing Maps in Natural Language Processing*. PhD thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Computer and Information Science.

Honkela, T. and Hyvärinen, A. (2004). Linguistic feature extraction using Independent Component Analysis. In *Proc. International Joint Conference on Neural Networks (IJCNN 2004)*, pages 279–284.

Honkela, T., Pulkki, V., and Kohonen, T. (1995). Contextual relations of words in Grimm tales, analyzed by Self-Organizing Map. In *Proceedings of International Conference on Artificial Neural Networks, ICANN-95*, pages 3–7, Paris, France.

Hu, Y., Matveeva, I., Goldsmith, J., and Sprague, C. (2005a). The SED heuristic for morpheme discovery: a look at Swahili. In *Proc. 2nd Workshop of Psychocomputational Models of Human Language Acquisition*, pages 28–35, Ann Arbor, Michigan.

Hu, Y., Matveeva, I., Goldsmith, J., and Sprague, C. (2005b). Using morphology and syntax together in unsupervised learning. In *Proc. 2nd Workshop of Psychocomputational Models of Human Language Acquisition*, pages 20–27, Ann Arbor, Michigan.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons, Inc.

Iisa, K., Piehl, A., and Oittinen, H. (2002). *Kielenhuollon käsikirja*. Yrityskirjat Oy.

Iyer, R. M. and Ostendorf, M. (1999). Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39.

Jacquemin, C. (1997). Guessing morphology from terms and corpora. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 156–165, Philadelphia, Pennsylvania, USA.

Johnson, H. and Martin, J. (2003). Unsupervised learning of morphology for English and Inuktitut. In *Human Language Technology and North American Chapter of the Association for Computational Linguistics Conference (HLT-NAACL'03)*, Edmonton, Canada.

Juang, B. H., Rabiner, L. R., Levinson, S. E., and Sondhi, M. M. (1985). Recent developments in the application of Hidden Markov Models to speaker-independent isolated word recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'85)*, pages 9–12, San Francisco, California, USA.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Inc.

Karlsson, F. (1998). *Yleinen kielitiede (General linguistics)*. Yliopistopaino/Helsinki University Press.

Karttunen, L. and Beesley, K. R. (2005). Twenty-five years of finite-state morphology. In Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., and Yli-Jyrä, A., editors, *Inquires into Words, Constraints and Contexts. Festschrift in the Honour of Kimmo Koskenniemi on his 60th Birthday*, pages 1–13. Gummerus Printing, Saarijärvi, Finland.

Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of IJCNN'98, International Joint Conference on Neural Networks*, pages 413–418.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.

Kazakov, D. and Manandhar, S. (2001). Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Machine Learning*, 43:121–162.

Kirby, S. (2002). Natural language from artificial life. *Artificial Life*, 8(2):185–215.

Kirchhoff, K., Bilmes, J., Das, S., Duta, N., Egan, M., Ji, G., He, F., Henderson, J., Liu, D., Noamany, M., Schone, P., Schwartz, R., and Vergyri, D. (2003). Novel approaches to Arabic speech recognition: Report from the 2002 John-Hopkins workshop. In *Proceedings of ICASSP-2003*, volume 1, pages 344–347.

Kit, C. (2003). How does lexical acquisition begin? A cognitive perspective. *Cognitive Science*, 1(1):1–50.

Kit, C., Pan, H., and Chen, H. (2002). Learning case-based knowledge for disambiguating Chinese word segmentation: A preliminary study. In *Proceedings of the COLING'02 workshop SIGHAN-1*, pages 33–39, Taipei, Taiwan.

Kit, C. and Wilks, Y. (1999). Unsupervised learning of word boundary with description length gain. In *Proc. CoNLL99 ACL Workshop*, Bergen.

Klami, M. (2005). Unsupervised discovery of morphs in children's stories and their use in self-organizing map -based analysis. Master's thesis, University of Helsinki, Department of General Linguistics.

Kneissler, J. and Klakow, D. (2001). Speech recognition for huge vocabularies by using optimized sub-word units. In *Proc. 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 69–72, Aalborg, Denmark.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'95)*, pages 181–184.

Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, third extended edition.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585.

Kontorovich, L., Ron, D., and Singer, Y. (2003). A Markov model for the acquisition of morphological structure. Technical Report CMU-CS-03-147, School of Computer Science, Carnegie Mellon University.

Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production*. PhD thesis, University of Helsinki.

Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 22nd Annual Meeting of the Association for Computational Linguistics*, pages 178–181.

Koskenniemi, K. (1991). A discovery procedure for two-level phonology. *Computational Lexicology and Lexicography: A Special Issue Dedicated to Bernard Quemada*, VI(I):451–465.

Kurimo, M., Creutz, M., Varjokallio, M., Arisoy, E., and Saraclar, M. (2006a). Unsupervised segmentation of words into morphemes – Challenge 2005: An introduction and evaluation report. In Kurimo, M., Creutz, M., and Lagus, K., editors, *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy.

Kurimo, M. and Lagus, K. (2002). An efficiently focusing large vocabulary language model. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*, pages 1068–1073, Madrid, Spain.

Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimäki, T., Pylkkönen, J., Alumäe, T., and Saraclar, M. (2006b). Unlimited vocabulary speech recognition for agglutinative languages. In *Human Language Technology Conference/North American chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL'06)*, New York, USA.

Kurimo, M. and Turunen, V. (2005). To recover from speech recognition errors in spoken document retrieval. In *Proceedings of Interspeech 2005*, pages 605–608, Lisbon, Portugal.

Kwon, O.-W. and Park, J. (2003). Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Communication*, 39(3–4):287–300.

Lagus, K., Airola, A., and Creutz, M. (2002). Data analysis of conceptual similarities of Finnish verbs. In *Proceedings of CogSci 2002, the 24th annual meeting of the Cognitive Science Society*, Fairfax, Virginia, USA.

Lagus, K., Creutz, M., and Virpioja, S. (2005). Latent linguistic codes for morphemes using Independent Component Analysis. In *Proceedings of the 9th Neural Computation and Psychology Workshop, NCPW9*, Plymouth, UK. (in print).

Lagus, K. and Kurimo, M. (2002). Language model adaptation in speech recognition using document maps. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing (NNSP'02)*, pages 627–636, Martigny, Switzerland.

"Language" (2005). Encyclopedia Britannica Online. Search path: "language > meaning and style in language > types of meaning > lexical meaning". `http://search.eb.com/eb/article-27173?query=lexical%20meaning&ct=`. Retrieved August 11.

Lari, K. and Young, S. J. (1990). The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4(1):35–56.

Li, W. (1992). Random texts exhibit Zipf's-Law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38(6):1842–1845.

Lindén, K. (2003). Word sense disambiguation with THESSOM. In *Proceedings of WSOM'03 – Intelligent Systems and Innovational Computing*, Kitakyushu, Japan.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Language Processing*. The MIT Press.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19.

Matthews, P. H. (1991). *Morphology*. Cambridge Textbooks in Linguistics, 2nd edition.

McKinnon, R., Allen, M., and Osterhout, L. (2003). Morphological decomposition involving non-productive morphemes: ERP evidence. *Cognitive Neuroscience and Neuropsychology*, 14(6):883–886.

McTait, K. and Adda-Decker, M. (2003). The 300k LIMSI German broadcast news transcription system. In *Proceedings of EUROSPEECH 2003*, pages 213–216, Geneva, Switzerland.

Mohri, M. and Riley, M. (2002). Weighted finite-state transducers in speech recognition (tutorial). In *International Conference on Spoken Language Processing (ICSLP'02)*, Denver, Colorado, USA. `http://www.research.att.com/projects/mohri/fsm/tut.html`.

Muggleton, S. and Raedt, L. D. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679.

Nagata, M. (1997). A self-organizing Japanese word segmenter using heuristic word identification and re-estimation. In *Proc. Fifth workshop on very large corpora*, pages 203–215.

Neuvel, S. and Fulop, S. A. (2002). Unsupervised learning of morphology without morphemes. In *Proc. Workshop on Morphological & Phonological Learning of ACL'02*, pages 31–40.

Nilsson, N. J. (1996). *Introduction to Machine Learning (Draft of Incomplete Notes)*. Stanford University. `http://ai.stanford.edu/people/nilsson/mlbook.html`.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273.

Ordelman, R., van Hessen, A., and de Jong, F. (2003). Compound decomposition in Dutch large vocabulary speech recognition. In *Proceedings of EUROSPEECH 2003*, pages 225–228, Geneva, Switzerland.

Peng, F. and Schuurmans, D. (2001). Self-supervised Chinese word segmentation. In *Proc. Fourth International Conference on Intelligent Data Analysis (IDA)*, pages 238–247. Springer.

Peterson, M. A. (2001). Lecture on language as a system of symbols (Structural Linguistics). The American University in Cairo. `http://www.aucegypt.edu/academic/anth/anth352/lectures.htm`.

Pirrelli, V., Calderone, B., Herreros, I., and Virgilio, M. (2004). Non-locality all the way through: Emergent global constraints in the Italian morphological lexicon. In *Proc. 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 11–19, Barcelona, Spain.

Pylkkönen, J. and Kurimo, M. (2004). Using phone durations in Finnish large vocabulary continuous speech recognition. In *Proceedings of the 6th Nordic Signal Processing Symposium (Norsig)*, pages 324–326, Espoo, Finland.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman, Essex.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall International (UK) Limited.

Redlich, A. N. (1993). Redundancy reduction as a strategy for unsupervised learning. *Neural Computation*, 5:289–304.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Series in Computer Science, Singapore.

Rissanen, J. J. (1996). Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47.

Rosenfeld, R. (1996). A Maximum Entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228.

Rosenfeld, R. (1997). A whole sentence Maximum Entropy language model. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, California, USA.

Rosenfeld, R. (2000). Two decades of language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Rumelhart, D. E. and McClelland, J. L. (1986). On learning the past tenses of English verbs. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2, pages 216–271. MIT Press, Cambridge, CA, USA.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Inc.

Saeed, J. I. (1997). *Semantics*. Blackwell Publishers.

Saffran, J. R., Newport, E. L., and Aslin, R. N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35:606–621.

Sapir, E. (1921). *Language – An Introduction to the Study of Speech*. Harcourt, Brace and Company, Inc. Renewed 1949 by Jean V. Sapir.

Schalkoff, R. J. (1992). *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc.

Schone, P. and Jurafsky, D. (2000). Knowledge-free induction of morphology using Latent Semantic Analysis. In *Proc. CoNLL-2000 & LLL-2000*, pages 67–72.

Schone, P. and Jurafsky, D. (2001). Knowledge-free induction of inflectional morphologies. In *Proc. NAACL-2001*.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–168.

Siivola, V., Kurimo, M., and Lagus, K. (2001). Large vocabulary statistical language modeling for continuous speech recognition in Finnish. In *Proceedings of EUROSPEECH 2001*, pages 737–740, Aalborg, Denmark.

Siivola, V. and Pellom, B. L. (2005). Growing an n-gram language model. In *Proceedings of Interspeech 2005*, Lisbon, Portugal.

Snover, M. G. and Brent, M. R. (2001). A Bayesian model for morpheme and paradigm identification. In *Proc. 39th Annual Meeting of the ACL*, pages 482–490.

Snover, M. G., Jarosz, G. E., and Brent, M. R. (2002). Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *Proc. Workshop of Morphological & Phonological Learning of ACL'02*, pages 11–20.

Steels, L. (1997). The origins of syntax in visually grounded robotic agents. In *Proceedings of IJCAI97*.

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*, pages 901–904. `http://www.speech.sri.com/projects/srilm/`.

Szarvas, M. and Furui, S. (2003). Evaluation of the stochastic morphosyntactic language model on a one million word Hungarian task. In *Proceedings of EUROSPEECH 2003*, pages 2297–2300, Geneva, Switzerland.

Theron, P. and Cloete, I. (1997). Automatic acquisition of two-level morphological rules. In *Proc. Fifth Conference on Applied Natural Language Processing*, pages 103–110.

Ullman, M. T. (2001). The declarative/procedural model of lexicon and grammar. *Journal of Psycholinguistic Research*, 30(1).

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York.

Varis, T. (2004). Baum-Welch -algoritmin käyttö kategorisen sanojen segmentointimallin parametrien ohjaamattomassa oppimisessa. Special assignment, Laboratory of Computer and Information Science, Helsinki University of Technology.

Vasilakopoulos, A. (2003). Improved unknown word guessing by decision tree induction for POS tagging with TBL. In *Proceedings of the 6th Annual CLUK Research Colloquium*, Edinburgh, UK.

Väyrynen, J. J. and Honkela, T. (2005). Comparison of Independent Component Analysis and Singular Value Decomposition in word context analysis. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 135–140.

Venkataraman, A. (2001). A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):352–372.

Vergyri, D., Kirchhoff, K., Duh, K., and Stolcke, A. (2004). Morphology-based language modeling for Arabic speech recognition. In *Proceedings of ICSLP-04*, volume 3, pages 2245–2248, Jeju, Korea.

Viikki, O. (1999). *Adaptive Methods for Robust Speech Recognition*. PhD thesis, Tampere University of Technology.

Virpioja, S. (2005). New methods for statistical language modeling. Master's thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Computer and Information Science.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Volk, N. (2004). Suomenkielisen tekstin laventaminen puhesynteesin laadun parantamiseksi. Master's thesis, University of Helsinki, Department of General Linguistics. `http://www.ling.helsinki.fi/suopuhe/lavennin/index.shtml`.

Wallace, C. S. and Boulton, D. M. (1968). An information measure for classification. *The Computer Journal*, 11(2):185–194.

Wallace, C. S. and Dowe, D. L. (1999). Minimum Message Length and Kolmogorov Complexity. *The Computer Journal*, 42(4):270–283.

Wallace, C. S. and Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society*, 49(3):240–265.

Whittaker, E. and Woodland, P. (2000). Particle-based language modelling. In *Proceedings of ICSLP-2000*, pages 170–173, Beijing, China.

Wicentowski, R. (2002). *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. PhD thesis, The Johns Hopkins University, Baltimore, Maryland, USA.

Wicentowski, R. (2004). Multilingual noise-robust supervised morphological analysis using the WordFrame model. In *Proc. 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 70–77, Barcelona.

Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell Publishers, Oxford.

Xue, N., Xia, F., Chiou, F.-D., and Palmer, M. (2004). The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 10(4):1–30. `http://www.cis.upenn.edu/%7Echinese/`.

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*, pages 161–168.

Yarowsky, D. and Wicentowski, R. (2000). Minimally supervised morphological analysis by multimodal alignment. In *Proc. ACL-2000*, pages 207–216.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1999). *The HTK Book*. Entropic / Cambridge University, version 2.2 edition. `http://htk.eng.cam.ac.uk/`.

Yu, H. (2000). Unsupervised word induction using MDL criterion. In *Proc. ISCSL*, Beijing.

Zipf, G. K. (1935). *The Psycho-Biology of Language – An Introduction to Dynamic Philology*. Houghton Mifflin Company, Boston.