Signal Processing and Acoustics

Contributions to Morphology Learning using Conditional Random Fields

Teemu Ruokolainen

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 67/2016

Contributions to Morphology Learning using Conditional Random Fields

Teemu Ruokolainen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall S1 of the school on 3 June 2016 at 12.

Aalto University
School of Electrical Engineering
Signal Processing and Acoustics
Speech and Language Processing

Supervising professor
Professor Mikko Kurimo, Aalto University, Finland

Thesis advisor
Doctor Sami Virpioja, Aalto Univeristy, Finland

Preliminary examiners
Professor Suresh Manandhar, University of York, UK
Doctor Filip Ginter, University of Turku, Finland

Opponent
Assistant Professor Chris Dyer, Carnegie Mellon University, USA

Aalto University publication series
DOCTORAL DISSERTATIONS 67/2016

ISBN 978-952-60-6753-7 (printed)
ISBN 978-952-60-6754-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6754-4

Unigrafia Oy
Helsinki 2016

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Teemu Ruokolainen

Name of the doctoral dissertation
Contributions to Morphology Learning using Conditional Random Fields

Publisher School of Electrical Engineering

Unit Signal Processing and Acoustics

Series Aalto University publication series DOCTORAL DISSERTATIONS 67/2016

Field of research Language Technology

Manuscript submitted 7 December 2015 **Date of the defence** 3 June 2016

Permission to publish granted (date) 1 March 2016 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Natural language processing (NLP) refers to the study of systems performing natural language related tasks in an automatic manner, that is, without human supervision or interference. This thesis work considers NLP problems related to morphology analysis, that is, the description of internal structure of words. Acquiring knowledge of morphology is necessary in order for applications, such as search engines, machine translators, and speech recognizers, to successfully address rare and previously unseen word forms. In particular, we focus on two widely applied morphological analysis tasks, namely, morphological tagging and segmentation. In morphological tagging, the aim is to assign words in sentential contexts with word class labels describing their morphological properties. Meanwhile, morphological segmentation considers describing the inner word structure by splitting word forms into their smallest meaning-bearing units, morphemes.

In the scope of this thesis, we approach the morphological tagging and segmentation problems using statistical, data-driven machine learning methodology. Using this approach, the processing systems are learned (estimated) based on training data prepared manually by a human expert. In particular, we focus on the highly influential conditional random field (CRF) model proposed for sequence tagging and segmentation in the early 2000s.

As the first main contribution, the thesis discusses data-driven morphological segmentation employing the CRF model. A particular emphasis is placed on the semi-supervised learning setting, in which the available data consists of a small number of annotated segmentation examples and a large amount of unannotated raw word forms. The provided empirical evaluation on six languages shows that the proposed semi-supervised CRF-based approach is highly successful in the considered morphological segmentation task compared to earlier methods. In particular, the performed error analysis shows that closed class phenomena, such as suffixation of English and Finnish, can be learned already from a small number of annotated examples in a supervised manner. Meanwhile, open morpheme class phenomena, such as compounding of Finnish, can be learned by additionally exploiting the large unannotated word list using the semi-supervised approach.

As the second main contribution, the thesis contains a presentation of FinnPos, the first open-source statistical morphological tagging and lemmatization toolkit designed specifically for Finnish. The CRF-based FinnPos system is readily applicable for tagging and lemmatization of running text with models learned from the recently published Finnish Turku Dependency Treebank and FinnTreeBank.

Keywords natural language, morphology, conditional random fields, tagging, segmentation

ISBN (printed) 978-952-60-6753-7 **ISBN (pdf)** 978-952-60-6754-4

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2016

Pages 188 **urn** http://urn.fi/URN:ISBN:978-952-60-6754-4

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Teemu Ruokolainen

Väitöskirjan nimi
Kontribuutioita morfologian oppimiseen ehdollisilla satunnaiskentillä

Julkaisija Sähkötekniikan korkeakoulu

Yksikkö Signaalinkäsittely ja akustiikka

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 67/2016

Tutkimusala Kieliteknologia

Käsikirjoituksen pvm 7.12.2015 **Väitöspäivä** 3.6.2016

Julkaisuluvan myöntämispäivä 1.3.2016 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Luonnollisen kielen käsittelyssä (LKK) tutkitaan järjestelmiä, jotka suorittavat ihmiskieleen liittyviä tehtäviä automaattisesti ilman ihmisen valvontaa. Tässä väitöskirjassa tarkastellaan LKK-järjestelmiä, jotka liittyvät morfologiseen analyysiin eli sanojen sisäisen rakenteen kuvaukseen. Morfologiset kuvaukset ovat tarpeellisia monien sovellusten, kuten hakukoneiden, kielenkääntäjien ja puheentunnistimien, kannalta, jotta kyseiset sovellukset voivat käsitellä harvinaisia ja tuntemattomia sanamuotoja. Työssä keskitytään erityisesti kahteen yleisesti käytettyyn analyysimenetelmään, morfologiseen jäsennykseen ja pilkontaan. Morfologisessa jäsennyksessä sanamuodoille annetaan luokituksia niiden morfologisten ominaisuuksien mukaan. Morfologisessa pilkonnassa sanojen sisäistä rakennetta kuvaillaan pilkkomalla sanamuodot niiden pienimpiin merkitystä sisältäviin osiin, morfeemeihin.

Tässä väitöskirjassa morfologista jäsennystä ja pilkontaa lähestytään käyttäen tilastollista koneoppimismetodologiaa eli järjestelmät oppivat suorittamaan analyysin asiantuntijan muodostamien esimerkkien avulla. Erityisesti keskitytään ehdollisten satunnaiskenttien (ESK) soveltamiseen. 2000-luvun alussa julkaistua ESK-mallia on aikaisemmin sovellettu menestyksekkäästi useissa jäsennys- ja pilkontatehtävissä.

Väitöskirjan ensimmäisenä pääkontribuutiona työssä tarkastellaan morfologisen pilkonnan oppimista ESK-mallin avulla. Erityisesti tarkastellaan puoliohjattua oppimisasetelmaa, jossa käytettävissä oleva data muodostuu pienestä määrästä annotoituja pilkontaesimerkkejä ja suuresta määrästä annotoimattomia, "raakoja", sanamuotoja. Kokeelliset tulokset kuudella kielellä osoittavat, että ehdotettu puoliohjattu ESK-pohjainen lähestymistapa on erittäin kilpailukykyinen menetelmä aikasemmin julkaistuihin menetelmiin verrattuna. Erityisesti huomataan, että suljettujen luokkien ilmiöt, kuten suomen kielen suffiksaatio, voidaan oppia jo pienestä määrästä annotoituja esimerkkejä ohjatulla opetuksella. Toisaalta avoimen luokkien ilmiöt, kuten yhdyssanojen muodostaminen suomessa, voidaan oppia hyödyntämällä suurta määrää annotoimattomia sanamuotoja käyttäen puoliohjattua opetusta.

Toisena pääkontribuutiona väitöskirjassa esitellään FinnPos, ensimmäinen suomen kielelle julkaistu avoimen lähdekoodin tilastollinen morfologinen jäsennin. ESK-malliin pohjautuvaa FinnPos-järjestelmää voidaan soveltaa suomen kielisen tekstin morfologiseen jäsentämiseen käyttäen Turku Dependency Treebank- ja FinnTreebank-puupankkien avulla opettuja malleja.

Avainsanat luonnollinen kieli, morfologia, ehdolliset satunnaiskentät, jäsennys, pilkonta

ISBN (painettu) 978-952-60-6753-7 **ISBN (pdf)** 978-952-60-6754-4

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2016

Sivumäärä 188 **urn** http://urn.fi/URN:ISBN:978-952-60-6754-4

Preface

This thesis work was conducted in Aalto University in the automatic speech recognition research group led by Prof. Mikko Kurimo. The work began in 2011 at the Department of Information and Computer Science and continued at the Department of Signal Processing and Acoustics after the group relocated. The work was financially supported by Langnet (Finnish doctoral program in language studies) and the Academy of Finland under the grant no 251170 (Finnish Centre of Excellence Program (2012-2017)).

I am grateful to my supervisor Prof. Kurimo for the guidance and the opportunity to participate in language processing research. I also wish to thank my instructor Dr. Sami Virpioja whose prior work on morphology learning had a major influence on the topic choice and execution of the research presented in this thesis.

Moreover, I would like to thank my co-authors Krister Lindén, Kairit Sirts, and Stig-Arne Grönroos for their contributions in the publications. In particular, I wish to express my sincerest gratitude towards Oskar Kohonen and Miikka Silfverberg with whom I had the pleasure to share the majority of the daily workload.

In addition to those mentioned, I wish to thank my many friends and colleagues at the group and department: Ulpu Remes, Hande Topa, Onur Dikmen, Paul Wagner, Reima Karhila, Peter Smith, Matti Varjokallio, Seppo Enarvi, Kalle Palomäki, Heikki Kallasjoki, Sami Keronen, Andre Mansikkaniemi, and Ana Ramirez Lopez.

During the final phases of the work I received helpful comments from the pre-examiners of this thesis Prof. Suresh Manandhar and Dr. Filip Ginter.

Finally, I wish to thank my mom and dad and my big brother and big sisters for their love and support over the years.

Helsinki, April 12, 2016,

Teemu Ruokolainen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, pages 29-37, Sofia, Bulgaria. 2013.

II Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. Painless Semi-Supervised Morphological Segmentation using Conditional Random Fields. In *Proceedings of the Fourteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 24-29, Gothenburg, Sweden. 2014.

III Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. A Comparative Study of Minimally-Supervised Morphological Segmentation. *Computational Linguistics*, 42:1, pages 91-120. 2016.

IV Teemu Ruokolainen, Miikka Silfverberg, Mikko Kurimo, and Krister Linden. Accelerated Estimation of Conditional Random Fields using a Pseudo-Likelihood-inspired Perceptron Variant. In *Proceedings of the Fourteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 74-78, Gothenburg, Sweden. 2014.

V Miikka Silfverberg, Teemu Ruokolainen, Krister Linden, and Mikko Kurimo. Part-of-Speech Tagging using Conditional Random Fields: Exploiting Sub-Label Dependencies for Improved Accuracy. In *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 259-264, Baltimore, Maryland, USA. 2014.

VI Miikka Silfverberg, Teemu Ruokolainen, Krister Linden, and Mikko Kurimo. FinnPos: An Open-Source Morphological Tagging and Lemmatization Toolkit for Finnish. *Journal of Language Resources and Evaluation*, pages 16, accepted 2016.

Author's Contribution

Publication I: "Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields"

The paper describes work conducted on statistical learning of morphological segmentation from a small number, say hundreds, of annotated word forms. The work stemmed from the present author's observation that the previously proposed approaches to segmentation in this learning setting were based on generative semi-supervised models which utilize a large number of unannotated word forms in addition to the small annotated word set. In the presented paper, I wanted to study how these methods fared against a state-of-the-art supervised segmentation model learned from solely the small number of annotated word forms. To this end, I studied performing the task using the conditional random field model with a novel feature extraction scheme and, subsequently, carried out the experiments jointly with the other authors. The current author was the main writer of the article.

Publication II: "Painless Semi-Supervised Morphological Segmentation using Conditional Random Fields"

The paper presents a study on how to perform the morphological segmentation using the conditional random field model in a semi-supervised setting, that is, when utilizing both the annotated and unannotated word forms. To this end, I studied performing the task using the conditional random field model with a novel feature expansion scheme and, subsequently, carried out the experiments jointly with the other authors. The current author was the main writer of the article.

Publication III: "A Comparative Study of Minimally-Supervised Morphological Segmentation"

The paper describes a comparative study on morphological segmentation methodology in the learning setting applied in Publications I and II. The study comprises a literature survey and an in-depth empirical evaluation. The current author contributed to the literature survey, experiments with emphasis on the conditional random field model, overall coordination of the work, and writing of the article.

Publication IV: "Accelerated Estimation of Conditional Random Fields using a Pseudo-Likelihood-inspired Perceptron Variant"

The paper describes a heuristic approximative learning approach to conditional random field in presence of a large number of class labels. In this work, I wanted to study if the key idea of the classic pseudo-likelihood approximation was applicable in the perceptron algorithm framework. To this end, I formulated a variant of the perceptron algorithm for learning of conditional random fields and conducted experiments in morphological tagging of several morphologically rich languages jointly with the other authors. The current author was the main writer of the article.

Publication V: "Part-of-Speech Tagging using Conditional Random Fields: Exploiting Sub-Label Dependencies for Improved Accuracy"

The paper describes a means of improving the accuracy of morphological taggers using a novel feature extraction scheme for the conditional random field model. The scheme is applicable when the morphological labels are fine-grained, that is, have a rich inner structure. The current author participated in the methodology design, experiment design, and was the main writer of the article.

Publication VI: "FinnPos: An Open-Source Morphological Tagging and Lemmatization Toolkit for Finnish"

The paper describes FinnPos, the first open-source morphological tagging and lemmatization toolkit for Finnish. The current author participated in the methodology design, experiment design, and was the main writer of the article.

List of Abbreviations and Symbols

The following lists contain abbreviations and symbols employed throughout the thesis. The remaining notation is introduced and defined when necessary.

Abbreviations

CRF	conditional random fields
FTB	FinnTreeBank
HMM	hidden Markov model
MEMM	maximum entropy Markov model
TDT	Turku Dependency Treebank

Symbols

i, j, k	generic index
n	order of a sequence model
N	set size (cardinality)
T	length of sequence
x, y	sequence variable
z	highest scoring (most probable) sequence given a model
$\mathcal{D}$	annotated (labeled) data set
$\mathcal{U}$	unannotated (unlabeled) data set
$\mathcal{Y}$	tag (label) set
$\boldsymbol{w}$	parameter vector
$p(x)$	probability distribution over random variable x
$p(x, y)$	joint probability distribution over random variables x and y
$p(y \mid x)$	conditional probability distribution over random variable y given x
$p(y \mid x; \boldsymbol{w})$	conditional probability distribution over random variable y given x parametrized by $\boldsymbol{w}$
χ	feature function set
$\boldsymbol{\phi}$	feature vector
$\boldsymbol{w} \cdot \boldsymbol{\phi}$	inner (dot) product of $\boldsymbol{w}$ and $\boldsymbol{\phi}$

1. Introduction

1.1 Natural Language Processing and Scope of the Thesis

Natural language processing refers to the study of systems performing natural language related tasks in an automatic manner, that is, without human supervision or interference. Due to the rapid digitalization of the society during the past few decades, varying language processing systems have become an established feature of the modern everyday life, with such application examples as search engines, machine translators, and speech recognizers (speech-to-text translators).

For an automatized system, a machine, processing human language is an extremely challenging task. This is because the machine is inherently unaware of the rich structural and semantic aspects of the language humans most often take for granted. From the point of view of the machine, a search term written in the search bar of a web browser is merely a sequence of discrete symbols with no intrinsic meaning or purpose. In order to alleviate this problem, it is a common practice to pre-process the language prior to handing it over to the machine for subsequent processing. In pre-processing, the language is augmented with structural and semantic properties deemed important for the following applications, such as translation to another language or information retrieval from a database.

This thesis work examines pre-processing methods related to morphology, that is, the study of internal structure of words. As for language processing, acquiring knowledge of morphology is necessary in order for the applications to successfully address rare and previously unseen word forms. In essence, the aim is to deduce the syntactic and semantic properties of unknown word forms, fully or partially, from its inner components. In particular, we consider two widely applied morphological analysis tasks, namely, morphological tag-

ging and segmentation. In morphological tagging, the aim is to assign words in sentential contexts with word class labels describing their morphological properties. This type of augmentation is an integral pre-processing part of, for example, full parsing which in turn provides complete syntactical analyses for sentences. Meanwhile, morphological segmentation considers describing the inner word structure by splitting word forms into their smallest meaning-bearing units, morphemes. While a simplification of the diverse morphological phenomena present in languages, this type of analysis has nevertheless been found useful in a wide range of applications, including speech recognition, information retrieval, and machine translation.

Following the modern language processing discipline, the morphological tagging and segmentation problems are approached using statistical, data-driven machine learning methodology. Using this approach, the processing systems are learned (estimated) based on training data prepared manually by a human expert. Specifically, we focus on the highly influential conditional random field modeling framework proposed for sequence tagging and segmentation in the early 2000s. Since its introduction, this framework has been successfully applied to numerous language processing tasks, including parsing, information extraction, and word segmentation. Thus, the presented work extends this established field of literature by novel methodological and application contributions.

1.2 Contributions of the Thesis

This thesis contributes to the field of data-driven morphological segmentation in the following manner:

- A study of morphological segmentation using the conditional random field model in a supervised and semi-supervised learning settings (Publications I and II).

- A comparative study of data-driven morphological segmentation methodology, including a literature survey and an in-depth empirical evaluation (Publication III).

The contributions to data-driven morphological tagging are as follows:

- Heuristic methods for accelerated estimation of morphological taggers based

on conditional random fields in presence of large number of morphological labels (Publications IV and VI).

- Feature extraction scheme for improving the accuracy of a morphological tagger based on conditional random fields in presence of morphological labels with rich inner structure (Publications V and VI).

- FinnPos, the first open-source morphological tagging and lemmatization toolkit for Finnish (Publication VI).

1.3 Structure of the Thesis

The thesis is structured as follows. In Section 2, we discuss essential terminology related to morphology and statistical learning, and subsequently describe the morphological segmentation and morphological tagging tasks. Section 3 discusses graphical model methodology applicable for segmentation and tagging problems in language processing, including the influential hidden Markov and conditional random field models. Sections 4 and 5 provide summarizing presentations of the current author's contributions to the morphological segmentation and morphological tagging tasks based on Publications I-III and Publications IV-VI, respectively. Finally, conclusions on the thesis work are presented in Section 6.

2. Morphology Learning

This chapter provides a background discussion on morphology in natural language processing and statistical machine learning. We begin by providing an overview of the fundamental terminology and concepts related to morphology and statistical learning in Section 2.1. We then discuss the morphological segmentation and morphological tagging tasks in Sections 2.2 and 2.3, respectively.

2.1 Basic Terminology

This section provides an overview of terminology and concepts related to morphology and statistical learning.

2.1.1 On Morphology

In the scope of this thesis, the fundamental building block of language is the word. In what follows, we discuss essential terminology related to morphology, that is, the study of inner structure of words. In particular, we consider such concepts as **part-of-speech**, **inflection**, **derivation**, **compounding**, and **morphemes**. We then discuss terminology related to **morphological typology**, that is, the classification of languages according to their morphological properties.

It should be noted that, given the scope of the thesis, providing extensive descriptions of the linguistic phenomena underlying the presented concepts is neither feasible nor necessary. Rather, the purpose of this section is to introduce the terminology required for the reader to follow the subsequent discussion, and cited literature, on morphology from the perspective of language processing and statistical learning.

	part-of-speech	examples
1.	adjective	beautiful, general, slow
2.	adverb	beautifully, generally, slowly
3.	conjunction	and, for, but
4.	determiner	a, an, the
5.	noun	cat, dog, house
6.	numeral	one, eleven, hundred
7.	preposition	at, in, to
8.	pronoun	I, he, them
9.	verb	say, eat, sleep

Table 2.1. English part-of-speech categories following Quirk et al. [1985].

Part-of-Speech Part-of-speech refers to a category of words with similar morphosyntactic (inner structure and functionality in sentence) properties. For example, Quirk et al. [1985] and Hakulinen et al. [2004] define 9 and 10 part-of-speech classes for English and Finnish, respectively. The categories are presented in Tables 2.1 and 2.2 along with example word forms.

The part-of-speech categories, such as those presented in Tables 2.1 and 2.2, differ in that some are more amenable to introduction of novel words than others. For example, in English and Finnish, the noun, adjective, and verb classes are constantly expanded with new words to accommodate new communicational needs. These type of categories are referred to as **open**. In contrast, categories containing functional words, such as the English and Finnish prepositions, tend to be expanded with new word forms rarely, if at all, and are referred to as **closed**.

Derivation and Inflection In morphology, derivation and inflection refer to processes of creating novel word forms from existing words. In the scope of this thesis, the processes of interest comprise **prefixation** and **suffixation** of word **stems**. The stem refers to the part of the word form which remains unchanged when applying the affixation. Other means of derivation and inflection include, but are not limited to, reduplication of word segments and changes in stress, pitch, and tone.

For an example of word derivation, consider the English word stem *build* and its derivatives *building* and *buildable* obtained by applying suffixes *-ing* and *-able*, respectively. It is commonly stated that the process of derivation changes the part-of-speech of the original word: in the previous examples, the verb *build* becomes the noun *building* and the adjective *buildable*. Meanwhile, inflection often does not modify the part-of-speech of the original word form, but rather affects other grammatical information, such as **tense**, **number**, **per-**

	part-of-speech	examples
1.	adjective	kaunis (beaufiful), yleinen (general), hidas (slow)
2.	adverb	kauniisti (beautifully), yleisesti (generally), hitaasti (slowly)
3.	comparative adjective and adverb	kauniimpi (more beautiful), kaunein (most beautiful), kauniimmin (more beautifully), kauneimmin (most beautifully)
4.	infinitive and participle	sanoa (to say), sanomassa (about to say), sanoen (by saying), sanova (saying)
5.	noun	kissa (cat), koira (dog), talo (house)
6.	number	yksi (one), yksitoista (eleven), sata (hundred)
7.	particle	ja (and), että (that), niin (so), aivan (indeed), ehkä (maybe), aina (always)
8.	pre-/postpositions	ennen (before), jälkeen (after), alla (below)
9.	pronoun	minä (I), sinä (you), he (they)
10.	verb	sanon (I say), sanoit (you said), sanoivat (they said)

Table 2.2. Finnish part-of-speech categories following Hakulinen et al. [2004].

son, and **case**. For example, consider the inflected form *played* of the English word form *play*, in which the suffix *-ed* changes the tense of the verb *play* from present to past.

Finally, consider a set of word forms composed using the process of inflection, for example *eat, eats, ate, eaten* or *cat, cats*. Then, consider choosing an index term to represent this group in a dictionary or other lexical resource. By convention, the term is chosen among the word form variants and is referred to as the **lemma**, that is, the base form. For example, in English, the lemma for verbs is defined to be the infinitive form which does not contain person or

	word form	structure	translation
1.	alue	stem	area
2.	asuin+alue	prefix + stem	residental area
3.	asuin+alue+i	prefix + stem + suffix	residental areas
4.	asuin+alue+i+lla	prefix + stem + suffix + suffix	in residential areas

Table 2.3. A reconstruction of the Finnish word *asuinalueilla* (*in residential areas*).

tense information (for example, *eat, run sleep*), whereas for nouns the base is the singular (for example, *cat, dog, house*). Meanwhile, in Finnish, the lemma for verbs is defined to be the first infinitive form (*sanoa (to say), syödä (to eat), nukkua (to sleep)*) and for nouns the singular nominative (*kissa (cat), koira (dog), talo (house)*).

Compounding In derivation and inflection, novel words are created by affixiating word stems. Meanwhile, the process of compounding creates new words by combining multiple stems into single words. For example, consider the English compound word *snowball* consisting of the noun stems *snow* and *ball*. Some languages, such as Finnish, frequently apply compounding of more than two stems, exemplified by *polttoainesäiliö* (*fuel tank*) consisting of three components *poltto (combustion)*, *aine (substance)*, and *säiliö (tank)*.

Morphemes We next discuss word formation via affixation and compounding utilizing the concept of morphemes. To this end, consider reconstructing the Finnish word form *asuinalueilla* (*in residental areas*) via affixation of the stem *alue* (*area*) as presented in Table 2.3.

We then make the important observation that all the individual segments (*asuin*, *alue*, *i*, and *lla*) contribute to the meaning of the resulting word form. In other words, none of the segments can be removed or replaced without modifying the meaning of the full word form. These segments are referred to as **morphs**, that is, the surface forms of the morphemes. By a commonly applied definition, morphemes correspond to the smallest meaning-bearing units of a language.

In some cases, such as the word form *asuinalueilla*, the individual morph components remain intact despite applying the concatenation. However, due to **morphophonical** properties of languages, this is not always the case. For example, consider the Finnish word form *lumi* (*snow*) and its plural form *lumet* with a segmentation analysis *lume+t*. In contrast to the form *asuinalueilla*, in which the stem *alue* remains intact, the stem *lumi* is modified to *lume* due to morphophonology of Finnish. Nevertheless, despite this change in the surface

form, the morphs *lumi* and *lume* are associated with the same morpheme, that is, the linguistic unit bearing the meaning of *snow*.

As in the case of part-of-speech discussed above, also the morphemes in languages tend to have the open and closed class distinction. For example, in English and Finnish, suffixes tend to serve syntactic purposes, such as denoting tense, person, number, or case: consider the English and Finnish suffixes *-s* and *-t*, respectively, which mark the plural number. In consequence of this syntactic functionality, the suffix classes in English and Finnish are closed and contain a much more smaller number of morphemes compared to the open stem category.

Morphological Typology Finally, we discuss morphological typology, that is, the classification of languages according to their morphological properties. To this end, consider the terms **isolative** and **synthetic** languages. In languages with high amount of isolating morphological properties, words tend to comprise their own morphemes. Meanwhile, in heavily synthetic languages, words tend to contain multiple morphemes. Synthetic languages can be described further according to their **agglutinative** (**concatenative**) and **fusional** properties. In the former, the morphs tend to have clear boundaries between them while in the latter, the morphs tend to be indistinguishable. For examples of agglutinative and fusional word formation, consider the English verbs *played* (past tense of *play*) and *sang* (past tense of *sing*). While the previous can be effortlessly divided into two segments as *play+ed* (word stem + suffix marking past tense), there are no such distinct boundaries in the latter. Generally, languages with synthetic properties mix concatenative and fusional schemes and contain agglutinative properties to varying degrees.

2.1.2 On Statistical Learning

Language processing systems, and artificial intelligence systems in general, can be roughly divided to two categories, **rule-based** and **statistical (data-driven)**. In essence, rule-based approaches consist of sets of manually prepared if-then statements which allow the system to provide desired responses, outputs, to given inputs. Meanwhile, instead of applying a set of rules, statistical approaches aim to learn the input-output mapping from data. In the scope of this thesis, we consider three statistical learning paradigms, namely, **supervised**, **unsupervised**, and **semi-supervised** learning. These approaches differ in their learning aims as well as in the type of data utilized.

In the supervised setting, the model learns to yield desired outputs to given

inputs based on a training data set of manually prepared exemplar input-output pairs. Examples of commonly applied supervised learning tasks include:

- **Classification**. In classification, one assumes a training data set of input-output pairs, $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$, where the input variables $\boldsymbol{x}$ are vectors of dimensionality $|\boldsymbol{x}|$ and the output variables y take values from a discrete class label set $\mathcal{Y}$ of size $N_\mathcal{Y}$. This type of manually prepared data is referred to as **annotated** or **labeled**. The problem is then to learn a classification model from $\mathcal{D}$ which maps input instances not seen during training to class labels. For example, in hand-written digit recognition, the input x contains the pixel information of the digit image and output y is the set of digits from 0 to 9. For a comprehensive overview of classification methodology, see, for example [Alpaydin, 2004, Chapter 2 and 14] and [Bishop, 2006, Chapter 4].

- **Sequence Tagging**. In sequence tagging, one assumes a training data set of input-output pairs, $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$, where the input variables x are sequences of variables $x = (x_1, x_2, \ldots, x_T)$, while the output comprises of a sequence of T multiple, interdependent variables $y = (y_1, \ldots, y_i, \ldots, y_T)$, where each variable y_i, $i \in 1, \ldots, T$, takes values from a discrete class label set $\mathcal{Y}$. Examples of sequence tagging include the morphological tagging and segmentation problems discussed in detail throughout this thesis. For an overview of tagging methodology utilizing particularly graphical modeling framework, see Chapter 3.

In contrast to supervised models, unsupervised models aim to learn from a data set of input instances, often with an intention of discovering a more compact representation of the original data. Examples of unsupervised learning include:

- **Clustering.** In clustering, one assumes a training set of data points, $\mathcal{U} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N_\mathcal{U}}$, where the instances $\boldsymbol{x}$ are vectors of dimensionality $|\boldsymbol{x}|$. This type of raw data set is referred to as **unannotated** or **unlabeled**. The clustering task is then to assign the instances into K groups (clusters) so that instances belonging to the same group are similar (according to a defined similarity measure) and instances belonging to different groups dissimilar. As an exemplar task, consider **part-of-speech induction** [Christodoulopoulos et al., 2010], in which word forms are assigned to clusters according to

their contextual similarity, where the context comprises neighboring word in running text. The learning relies on the insight that two words which can be replaced with each other in a sentence without breaking the grammaticality of the sentence belong to the same syntactic group (part-of-speech) with a high probability. For a comprehensive overview of clustering methodology, see, for example [Alpaydin, 2004, Chapter 7] and [Bishop, 2006, Chapter 9].

- **Dimensionality Reduction.** Similarly to the clustering problem, in dimensionality reduction, one assumes a training set of data points, $\mathcal{U} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N_\mathcal{U}}$, where the instances $\boldsymbol{x}$ are vectors of dimensionality $|\boldsymbol{x}|$. The problem is then to learn a transformation which projects $\boldsymbol{x}$ from the original high-dimensional space into a space with smaller dimension while preserving some defined property of the original data. This projection serves as a means of compressing the original data into a more compact representation in a lossy manner. Examples of widely applied dimensionality reduction methods include the singular value decomposition [Golub and Reinsch, 1970], principal component analysis [Wold et al., 1987], and random projections [Bingham and Mannila, 2001].

Finally, the aim of semi-supervised learning is to utilize both the available annotated and unannotated data. Therefore, semi-supervised learning lies somewhere in between the supervised and unsupervised learning approaches. Examples of learning using both annotated and unannotated data include:

- **Semi-Supervised Classification.** In semi-supervised classification, the aim is to learn a classification model from annotated data $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$ similarly to standard classification while additionally utilizing unannotated data $\mathcal{U} = \{\boldsymbol{x}^{(j)}\}_{j=1}^{N_\mathcal{U}}$ for information on the distribution of the input variable $\boldsymbol{x}$. The inclusion of the unannotated data is commonly motivated by the fact that obtaining annotated data requires manual labor by a human expert and can, therefore, be both time consuming as well expensive.

- **Semi-Supervised Clustering.** In semi-supervised clustering, the key idea is to group the data points in the unannotated set $\mathcal{U} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N_\mathcal{U}}$ similarly to standard clustering while guiding the assignment into desired direction using a handful of annotated examples.

For a comprehensive overview of semi-supervised learning methodology and

applications, see [Zhu and Goldberg, 2009].

2.2 Morphological Segmentation

In this section, we discuss statistical morphological segmentation learning. We first provide an overview on how the problem has been historically approached in Section 2.2.1. We then briefly describe the Morpho Challenge data set in Section 2.2.2. This data set is utilized extensively in the experimental work discussed in detail in Chapter 4 (Publications I-III).

2.2.1 Overview

Morphological segmentation considers the problem of segmenting word forms into morphs, the surface forms of morphemes. For example, consider the English word *houses* with a corresponding segmentation *house+s*, where the segment *house* corresponds to the word stem and the suffix *-s* marks the plural number. For more examples of segmentations of English and Finnish word forms, see Table 2.4. While a simplification of the diverse morphological phenomena present in languages, this type of analysis has nevertheless gathered substantial attention by the computational linguistics community, beginning with the pioneering work on morphology learning of Harris [1955]. As for automatic language processing, such segmentations have been found to be a useful pre-processing step in a wide range of applications, including speech recognition [Hirsimäki et al., 2006, Narasimhan et al., 2014], information retrieval [Turunen and Kurimo, 2011], machine translation [de Gispert et al., 2009, Green and DeNero, 2012], and word representation learning [Luong et al., 2013].

Since the early work of Harris [1955], most research on morphological segmentation has focused on unsupervised learning, in which case the segmentation is learned from a list of unannotated word forms. This approach and learning setting has received further popularity due to its close relationship with the unsupervised word segmentation problem which has been viewed as a reasonable setting for language acquisition study [Brent, 1999, Goldwater, 2006]. As for applications, the unsupervised methods are appealing as they can be applied to any language, for which there exists a sufficiently large set of unannotated words in electronic form. Consequently, such methods provide a means of acquiring a type of morphological analysis for under-resourced languages as motivated, for example, by Creutz and Lagus [2002].

word form	segmentation
play	play
plays	play+s
played	play+ed
replay	re+play
players	play+er+s
kissa (cat)	kissa
kissalla (with cat)	kissa+lla
kissoilta (from cats)	kisso+i+lta
kotissa (domestic cat)	koti+kissa
kissanminttu (catnip)	kissa+n+minttu

Table 2.4. Morphological segmentation analyses for exemplar English and Finnish word forms.

While development of novel unsupervised model formulations has remained a topic of active research [Snyder and Barzilay, 2008, Poon et al., 2009, Monson et al., 2010, Lee et al., 2011, Spiegler and Flach, 2010, Sirts and Goldwater, 2013], recent work has also shown a growing interest towards semi-supervised learning [Poon et al., 2009, Kohonen et al., 2010, Sirts and Goldwater, 2013, Grönroos et al., 2014]. In the previous work, the annotated data sets are commonly small, on the order of a few thousands of word forms. In consequence, similarly to ?Noname manuscript No. (will be inserted by the editor) unsupervised methods, the minimally-supervised techniques can be seen as a means of acquiring a type of morphological analysis for under-resourced languages. Learning morphological segmentation in this setting is discussed further in the author's contributions in Chapter 4 (Publications I-III).

Finally, despite its intuitiveness, it should be noted that the segmentation representation of word forms is not equally applicable to all languages. To this end, recall the distinction of agglutinative (concatenative) and fusional word formation discussed in Section 2.1. In particular, while the words formed using agglutinative processes can be effortlessly divided into non-overlapping segments, in fusional languages such distinct boundaries are scarce. Thus, morphological segmentation can be most naturally applied to languages with a high amount of agglutinative properties. For example, the languages discussed in author's contributions in Chapter 4 are English, Estonian, Finnish, and Turkish.

2.2.2 Data

The main morphological segmentation data set employed in this thesis work originates from the Morpho Challenge competitions organized since 2005 at Aalto University (formerly Helsinki University of Technology).[1] The current version of the data set [Kurimo et al., 2010] includes manually prepared morphological segmentations in English, Finnish, and Turkish. In addition to the annotated training and development sets, the data contains a large number of unannotated, raw word lists for each language. The number of word forms are shown in Table 2.5. Note the large difference between the amount of annotated and unannotated word forms. Lastly, the data set includes a held-out, publicly non-available test set for evaluation purposes.

set	English	Finnish	Turkish
unannotated training	878,036	2,928,030	617,298
annotated training	1,000	1,000	1,000
annotated development	694	835	763

Table 2.5. The number of unannotated and annotated word forms in the Morpho Challenge data set [Kurimo et al., 2010].

2.3 Morphological Tagging and Lemmatization

In this section, we discuss the problem of morphological tagging and lemmatization, in which the aim is to assign words with morphological labels and lemmas in sentential contexts. In particular, we consider tagging and lemmatization of Finnish, where the morphological labels and lemmas are provided by OMorfi, an open-source rule-based morphological analyzer for Finnish developed by Pirinen [2008].[2]

2.3.1 Overview

Consider the Finnish exemplar sentence *Kissani syövät ruokaa* (*My cats eat food*). Then, consider assigning each word form a morphological analysis given its sentential context, where the analyses are provided by OMorFi [Pirinen, 2008]. Each analysis, shown in Table 2.6, consists of word lemma, part-of-speech, and a label set presenting fine-grained morphological information, in-

[1]The competition website can be found at `http://research.ics.aalto.fi/events/morphochallenge/`.

[2]The OMorFi analyzer is freely available at `http://code.google.com/p/omorfi/`.

word form	translation	full analysis
kissani	my cat	kissa+N+SG+NOM+SG1
	my cat's	kissa+N+SG+GEN+SG1
	my cats	kissa+N+PL+NOM+SG1
syövät	cancers	syöpä+N+PL+NOM
	eating (plural)	syödä+V+ACT+VA+PL+NOM
	eat (plural)	syödä+V+ACT+INDV+PRES+PL3
ruokaa	food	ruoka+N+SG+PAR

Table 2.6. All morphological analyses for word forms in the exemplar Finnish sentence *Kissani syövät ruokaa* (*My cats eat food*) provided by the OMorFi analyzer [Pirinen, 2008].

cluding tense, person, case, and number. We refer to the combination of the part-of-speech and the fine-grained information as the **morphological label**. Due to the ambiguity of the word forms *kissani* and *syövät*, the complete sentence has 9 possible analyses. However, from these, only the disambiguated sequence

Kissani	syövät	ruokaa
↓	↓	↓
kissa+N+PL+NOM+SG1	syödä+V+ACT+INDV+PRES+PL3	ruoka+N+SG+PAR

would be grammatically correct. This type of morphological annotation is an integral pre-processing step to, for example, full parsing which in turn returns complete morphosyntactic analyses for sentences [Haverinen et al., 2014].

An intuitive means of approaching this disambiguation problem from an automatic language processing perspective is to regard the morphological tagging and lemmatization tasks as two separate sub-problems. In morphological tagging, one performs the disambiguation for the morphological label part of the complete analysis. Currently, the state-of-the-art in statistical morphological tagging is achieved using variants of graphical models [Ratnaparkhi, 1996, Brants, 2000, McCallum et al., 2000, Lafferty et al., 2001, Toutanova et al., 2003]. These models learn to tag new sentences from a manually prepared corpus of sentences annotated with morphological labels. In order to perform the disambiguation, the graphical models exploit the following main information sources:

- **Lexical and Orthographic Information**. This information corresponds to the label distribution associated with a given a word form in the corpus. For example, if the word form *can* has been frequently assigned verb and noun labels in the training data, they are the most probable candidate labels for

can also in the test data. Meanwhile, for unknown word forms not observed in the training corpus, the label distribution can be associated with orthographic features (letter capitalization, prefixes, and suffixes, so forth) of the word form. For example, in English and Finnish, one can utilize the information that words beginning with a capital letter within sentence have an increased probability belonging to the (proper) noun class.

- **Label and Word Context**. The disambiguation information provided by the label context relies on the insight that some label transitions tend to occur much more often than others. For example, in English, a label sequence (Determiner, Noun, Verb) is extremely common (*a man walks*, *the cat sleeps*, and so forth), whereas a pair (Determiner, Verb) occurs rarely if at all since such word pairs as *a walks* and *the sleeps* are not grammatically correct sentence fragments. Meanwhile, the neighboring word context provide similar, although more fine-grained, disambiguation information to the label context.

Subsequent to assigning the morphological label, selecting the appropriate word lemma is straightforward given the set of full analyses provided by the OMorFi analyzer. However, OMorFi does not have a full vocabulary coverage, that is, for some word forms no analyses are returned. In these **out-of-vocabulary** cases, a simple baseline solution is to simply return the original word form as the lemma. However, a more appealing approach is to learn a lemmatization model in a data-driven manner and apply it to lemmatize the unknown word forms [Chrupala et al., 2008].

2.3.2 Treebanks

This section describes the recently published Finnish treebanks, namely, the Turku Dependency Treebank [Haverinen et al., 2009, 2014] and FinnTreeBank [Voutilainen, 2011]. These corpora contain manually prepared morphological annotations and are applicable for statistical learning of morphological taggers and lemmatizers. For reference, we first describe the classic English Penn Treebank [Marcus et al., 1993].

Penn Treebank The complete Penn Treebank is divided into 25 sections of newswire text extracted from the Wall Street Journal. It contains 47,287 sentences (1,127,315 word tokens) annotated using a set of 45 morphological labels. As an exemplar sentence fragment using the Penn annotation, consider

About	a	quarter	has	already	been	reallocated	.
↓	↓	↓	↓	↓	↓	↓	↓
IN	DT	NN	VBZ	RB	VBN	VBN	PUNCT

The Penn Treebank annotation scheme was largely influenced by the precursor Brown Corpus [Francis, 1964]. However, whereas the Brown Corpus utilized a tag set of 87 labels, the Penn Treebank reduced the number of tags to 45 by removing redundancy. This simplification was performed to make the tagging more amenable to statistical analysis and learning by reducing sparsity [Marcus et al., 1993, Section 2.1.1].

Turku Dependency Treebank and FinnTreeBank The Turku Dependency Treebank [Haverinen et al., 2014], or Turku Treebank for short, contains 13,572 sentences (183,118 word tokens) with texts from ten varying domains, such as Wikipedia articles, blog entries, and financial news.[3] The morphological analyses of word tokens are based on the outputs of OMorFi [Pirinen, 2008]. The annotation for each word token consists of word lemma and a morphological label, as discussed above. The morphological labels amount to a set of 2,355 different tags in total. Evidently, compared to Penn Treebank, the morphological labels of Turku Treebank are substantially more fine-grained. Also, the corpora differ in that Penn Treebank does not contain lemma annotation.

The FinnTreeBank [Haverinen et al., 2014] contains 19,121 sentences (162,028 word tokens) from varying domains including news and grammar examples.[4] Similarly to Turku Treebank, the morphological analyses of word tokens are based on the outputs of OMorFi [Pirinen, 2008] consisting of word lemma and a morphological label. The morphological labels amount to a set of 1,399 tags in total.

Discussion As mentioned above, compared to Penn Treebank, the morphological labels of Turku Treebank and FinnTreeBank are substantially more fine-grained. In general, the granularity of annotation depends on several factors, such as the preferences and choices made by the human expert providing the ”correct” linguistic gold standard. However, in case of English versus Finnish, the difference in granularity is arguably due to the fact that Finnish is morphologically much richer language compared to English. Specifically, in Finnish, a major part of the syntactic (sentence-level) grammatical information is encoded within the word forms through the process of inflection. For example, consider the English sentence fragment *also in residential areas* which can be

[3]The treebank is freely available at `http://bionlp.utu.fi/fintreebank.html`.

[4]The treebank is freely available at `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/treebank/`.

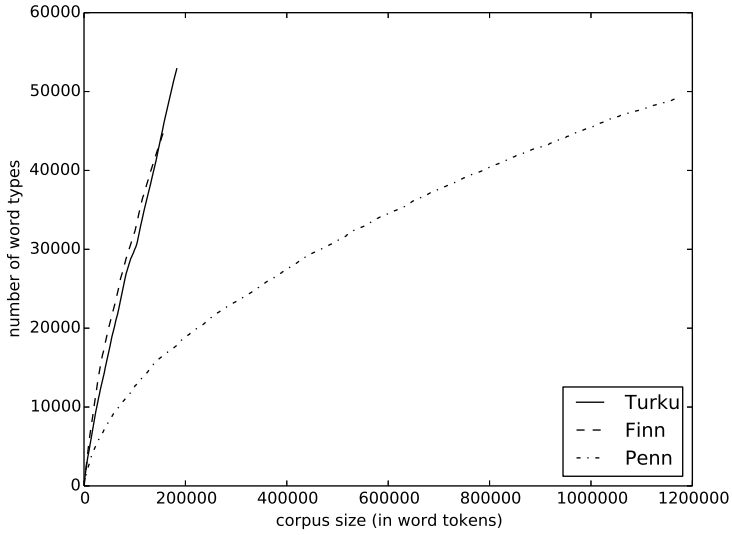

Figure 2.1. Number of encountered word forms as a function of corpus size (in word tokens) in Turku Treebank, FinnTreeBank, and Penn Treebank.

translated to Finnish with a single word *asuinalueillakin*. As a means of illustrating the difference, consider the number of encountered word forms as a function of corpus size (in word tokens) presented in Figure 2.1: although the Penn Treebank contains multiple times more word tokens, the total number of word types remains lower compared to Turku and FinnTreeBanks.

From statistical learning point of view, the substantially more fine-grained label sets of Turku and FinnTreeBank yield somewhat different challenges compared to the Penn Treebank. In particular, utilizing the disambiguation information provided by the label context as discussed in 2.3.1 will become increasingly difficult due to the sparser observed transition statistics. To illustrate this, consider the number of average label transition types presented in Table 2.7. What this table shows is that, on average, a randomly picked label in Penn Treebank is observed over 225 times more frequently compared to Turku and FinnTreeBanks. Meanwhile, on average, a randomly picked label pair is observed roughly 111 times more often, and so forth. Counteracting this sparsity problem is discussed in the author's contributions in Section 5.1.1 (Publications V and VI). Moreover, yet another sparsity problem rises from the higher growth rate of vocabulary: when annotating Finnish sentences, the taggers learned from Turku and FinnTreeBanks inevitably encounter word forms not seen during training more often compared to English. This phenomenom is referred to as the **out-of-vocabulary** problem.

order	Turku	Finn	Penn
0	90.9	115.8	26,084
1	7.2	7.0	812.0
2	2.2	2.1	65.6
3	1.3	1.3	12.2

Table 2.7. The average occurrence count of label transition types: the rows correspond to label types, label pair types, label triplet types, and label quadruplet types, respectively.

In addition to the increased sparsity, the high amount of labels also has a substantial impact on the computational complexity of learning of (discriminative) graphical tagging models, such as the conditional random fields discussed in detail in Section 3.4. Counteracting the increased computational burden using approximative learning approaches is discussed in Section 3.4.3 and in author's contributions in Section 5.1.2 (Publications IV and VI).

Finally, we note that the Turku Treebank and FinnTreeBank are the first freely available treebanks published for Finnish. Thus, for the first time, it has become possible to design statistical, data-driven morphological taggers and lemmatizers specifically for Finnish. Nevertheless, there exists some earlier work on Finnish morphological tagging conducted by Silfverberg and Linden [2011] who learned and evaluated their tagger using newspaper texts with an annotation provided by a commercial morphological tagging toolkit. This setting is problematic since the evaluation essentially measures the ability of the model to learn the errors of the commercial tagger. Meanwhile, in the scope of this thesis work, we are able to study learning morphological taggers for Finnish from manually annotated data as discussed in Section 5.2 (Publication VI).

2.4 Summary

In this section, we discussed the morphological segmentation and tagging problems in addition to the basic terminology related to word forming and statistical (data-driven) learning. In morphological segmentation, one considers the problem of segmenting word forms into morphs, the surface forms of morphemes. While a simplification of the diverse morphological phenomena present in languages, this type of analysis has nevertheless been found useful in a range of language processing applications. On the other hand, we discussed morphological tagging and lemmatization, in which the aim is to assign words with morphological labels and lemmas in sentential contexts. This type

of annotation is an integral pre-processing part of, for example, full parsing which in turn provides complete syntactical analyses for sentences.

3. Graphical Models for Tagging and Segmentation

This chapter provides an overview of statistical **graphical models** for sequence tagging and segmentation, including the highly influential hidden Markov models (HMMs) and their discriminative counterpart, the conditional random fields (CRFs). Throughout the section, we use the problem of morphological tagging for English as an example task to provide some concreteness to the formal presentation.

3.1 On Graphical Models

Consider a sequence $y = (y_1, \ldots, y_T)$ of T variables, where each variable $y_i, 1 \leq i \leq T$, takes values from a discrete label set $\mathcal{Y}$ of size $N_{\mathcal{Y}}$. Then, consider estimating the **joint** probability distribution $p(y_1, \ldots, y_T)$ based on a set of training instances, $\mathcal{D} = \{(y_1^{(i)}, \ldots, y_T^{(i)})\}_{i=1}^{N_{\mathcal{D}}}$, using, for example, the simple maximum likelihood estimator

$$p(y_1, \ldots, y_T) = \frac{count\Big((y_1, \ldots, y_T) \in \mathcal{D}\Big)}{N_{\mathcal{D}}}, \tag{3.1}$$

where $count(.)$ denotes the number of observed instances of specific value configurations $(y_1, \ldots, y_T)$ in $\mathcal{D}$.

Given T and $N_{\mathcal{Y}}$, a sequence $(y_1, \ldots, y_T)$ can take up to $N_{\mathcal{Y}}^T$ different label configurations. If the number of variables and labels are sufficiently small, the maximum likelihood estimates can be obtained from the training data with a high confidence. However, in case T and/or $N_{\mathcal{Y}}$ is increased, the estimation can become infeasible. This is simply because we may not able to acquire enough training instances to reliably estimate the counts of $(y_1, \ldots, y_T)$, that is, the estimates have excessively high variance. For example, assuming a label set size of 45 (the amount of morphological labels in the Penn Treebank), a sequence of 20 variables (the average sentence length in the Penn Treebank) can already take up to roughly $1.16 \cdot 10^{33}$ different value configuations. In statistical model-

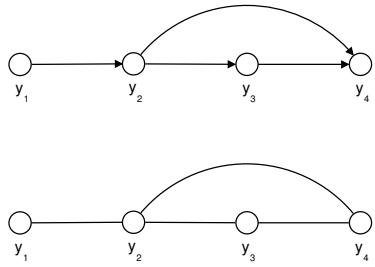

Figure 3.1. Example of a directed (top) and an undirected graph (bottom) defined over a sequence of four variables $y = (y_1, y_2, y_3, y_4)$.

ing, this phenomenom is commonly referred to as the **data sparsity** problem.

In order to combat the sparsity, one can utilize **factorizations** of the joint distribution $p(y_1, \ldots, y_T)$ defined over the complete set of variables. The central idea of the factorizations is to model the dependency structure within variables only partially, thus resulting in denser estimated statistics. In sequence modeling, it is common to model structure between temporally adjacent variables with the strongest dependencies, while ignoring the weaker, long-distance relations.

A well-established means of defining the factorizations is to use graphical models, in which we associate each variable y_i with a **node** in a **graph** (**network**), and define dependencies between nodes using **edges**. The graphical models come in two flavors, **directed** and **undirected**, depending on the applied edge types (arrows and lines for directed and undirected edges, respectively, as depicted in Figure 3.1). Next, we provide a compact description of the factorization properties yielded by the two graphical presentations.

3.1.1 Directed Graphs (Bayesian Networks)

We begin by defining the **parent-child** relationship between graph nodes. Using the graphical notation exemplified by the network in Figure 3.1, the node at the tail-end of the edge arrow is defined to be the parent, and the node at the head-end the child. We then denote the set of parental nodes of y_i as $\text{pa}(y_i)$. Thus, in the example in Figure 3.1, the parental node sets are $\text{pa}(y_1) = \emptyset$, $\text{pa}(y_2) = \{y_1\}$, $\text{pa}(y_3) = \{y_2\}$, and $\text{pa}(y_4) = \{y_2, y_3\}$. The directed graphs factorize the joint distribution over all variables as

$$p(y_i, \ldots, y_T) = \prod_i p\Big(y_i \,|\, pa(y_i)\Big)\,. \tag{3.2}$$

Applying this rule to the exemplar directed graph in Figure 3.1 would result in a factorization

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_2)p(y_4 \mid y_2, y_3)\,. \tag{3.3}$$

3.1.2 Undirected Graphs (Markov Random Fields)

In order to present the factorization properties of the undirected graphs, we first define the term **node clique** which corresponds to a fully connected set of variables (nodes). Then, a clique is referred to be **maximal** in case it is not a sub-clique of any larger cliques in the graph. Given these definitions, the undirected graphs, exemplified by the network in Figure 3.1, factorize as

$$p(y_1, \ldots, y_T) = \frac{1}{Z} \prod_c \Psi(y_c)\,, \tag{3.4}$$

where c indexes the maximal node cliques, y_c denotes variables in each maximal clique c, $\Psi(y_c)$ a function operating on y_c with $\Psi(y_c) > 0$, and Z a **normalization constant** (**partition function**) required to guarantee that $p(.)$ is a true probability distribution. This factorization follows from the theorem presented originally by Hammersley and Clifford [1971] in an unpublished manuscript.[1] Applying the rule (3.4), the exemplar undirected graph in Figure 3.1 would factorize as

$$p(y_1, y_2, y_3, y_4) = \frac{1}{Z}\Psi(y_1, y_2)\Psi(y_2, y_3, y_4)\,. \tag{3.5}$$

In contrast to the conditional probabilities $p\Big(y_i \mid pa(y_i)\Big)$, the factors $\Psi(y_c)$ do not have a straightforward probabilistic interpretation. Instead, we define them as **log-linear** functions using **parameter vector** $\boldsymbol{w}$ and **feature extraction function** $\boldsymbol{\phi}(y_c)$. Formally we write them as

$$\Psi(y_c; \boldsymbol{w}) = \exp\Big(\sum_i w_i \phi_i(y_c, c)\Big) = \exp\Big(\boldsymbol{w} \cdot \boldsymbol{\phi}(y_c, c)\Big)\,. \tag{3.6}$$

3.2 Hidden Markov Models

In this section, we discuss hidden Markov models (HMMs) for modeling the joint distribution $p(x, y)$ over two sequences of variables, $x = (x_1, x_2, \ldots, x_T)$ and $y = (y_1, y_2, \ldots, y_T)$. We refer to x and y as **input** and **output** sequences, respectively. In morphological tagging, the input x would be a sequence of words and the output y a sequence of morphological labels.

[1] At the time of writing, the manuscript is available at `http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf`

3.2.1 Model Definition

We begin by defining a first-order HMM using a directed graph over the variables x and y depicted in Figure 3.2. Employing this graph, the joint probability distribution over the sequences x and y factorizes as

$$p(y, x) = \prod_{i=2}^{T} p(x_i|y_i)p(y_i|y_{i-1})\,. \tag{3.7}$$

This first-order model (3.7) can be generalized to higher, arbitrary order n as

$$p(y, x) = \prod_{i=n+1}^{T} p(x_i|y_i)p(y_i|y_{i-1})\dots p(y_i|y_{i-n}, \dots, y_{i-1})\,. \tag{3.8}$$

The model order n is typically set to range from 1 to 3. This is because the dependencies are commonly strong within nearby output variables, but deteriorate quickly when n is increased. In case the model order is set to zero, $n = 0$, the HMM ignores the dependencies within the output variables y_i and predicts the labels based on solely the input x.

The distributions $p(x_i|y_i)$ and $p(y_i|y_{i-1})$, $\dots$, $p(y_i|y_{i-n}, \dots, y_{i-1})$ are commonly referred to as **emission** and **transition** probabilities, respectively. The purpose of the emission probabilities is to model the co-occurrence behavior of the input symbols x_i and the labels y_i. For example, according to maximum likelihood statistics estimated from the Penn Treebank, the emission probability $p(x_i = \text{house}\,|\,y_i = \text{Noun Singular}) = 129/134 \approx 0.963$ and $p(x_i = \text{house}\,|\,y_i = \text{Verb Base Form}) = 5/134 \approx 0.037$. In other words, in the Penn Treebank, the word form *house* occurs much more often as a noun compared to verb. Meanwhile, the transition probabilities capture the dependency structure between adjacent output variables. For example, $p(y_i = \text{Verb 3rd Person Singular}\,|\,y_{i-1} = \text{Noun Singular}) = 7174/26436 \approx 0.271$ and $p(y_i = \text{Personal Pronoun}\,|\,y_{i-1} = \text{Noun Singular}) = 702/21357 \approx 0.033$, that is, after observing a singular noun, it is much more probable to observe a transition to a 3rd person singular verb than to a personal pronoun.

3.2.2 Learning and Decoding

The distributions $p(x_i|y_i)$ and $p(y_i|y_{i-1})$, $\dots$, $p(y_i|y_{i-n}, \dots, y_{i-1})$ in (3.8) can be learned from training data using (smoothed) maximum likelihood estimates. In consequence, estimation of the HMMs is fast since obtaining the maximum likelihood estimates essentially reduces to **counting** event occurrences from the training data. This counting can be usually done in mere seconds even for large data and/or labels sets.

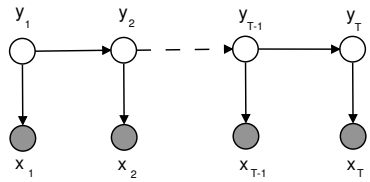

Figure 3.2. The first-order hidden Markov model for input and output sequences $x = (x_1, x_2, \ldots, x_T)$ and $y = (y_1, y_2, \ldots, y_T)$, respectively. The input variables $x = (x_1, \ldots, x_T)$ is assumed observed during both estimation and decoding, and is denoted with a darkened node.

Subsequent to estimation of the distributions $p(x_i \,|\, y_i)$ and $p(y_i \,|\, y_{i-n}, \ldots, y_{i-1})$, the HMM model can be used to assign output sequence y to any input sequence x, in a step commonly referred to as **decoding**. The "best guess" assignment z is obtained by using **maximum a posteriori** (MAP) graph inference, that is, solving an optimization problem

$$z = \arg\max_{y'} p\left(y', x\right), \tag{3.9}$$

where the input x is assumed observed (fixed). This maximization is solved using the dynamic programming algorithm referred to as the **Viterbi search** [Manning and Schütze, 1999, Section 10.2.2]. However, in case the number of labels is large, one might prefer to instead employ the approximative **beam search** method [Brants, 2000]. Beam search is discussed further in Section 3.4.3.

3.2.3 Sensitivity to Rich Features

The HMMs are a well-established framework for sequence modeling and have been particularly popular in morphological tagging [Brants, 2000]. Its main advantage is the speed of model estimation, which can be carried out in mere seconds even in case the task at hand involves large data and label sets. However, while fast, the HMMs suffer from a critical shortcoming, namely, sensitivity to rich, interdependent features. In what follows, we discuss this problem in more detail.

We begin by noting that emission probabilities $p(x_i \,|\, y_i)$ in the model definition (3.8) model the distribution over the input symbol x_i itself. However, in many cases, it would be beneficial to describe the input using more complex features. For example, in morphological tagging, one might exploit a feature which provides information about the capitalization of the word form, say, $p(\text{word form } x_i \text{ begins with a capital letter} \,|\, y_i = \text{Noun})$. Then, in order to preserve the ability to estimate the distributions using counting despite extended

feature set, one typically employs the **Naive Bayes** assumption, that is, the variables $x_{i,j}$ are presumed to be independent given label variable y_i. This is written formally as

$$p(x_i \,|\, y_i) = \prod_j p(x_{i,j} \,|\, y_i)\,. \tag{3.10}$$

While the Naive Bayes assumption successfully conserves the ease of model estimation, it inconveniently also makes the HMMs notoriously sensitive to the dependent features causing potentially bad performance on test data. For a broader discussion on the problems caused by the Naive Bayes assumption, see, for example, [Sutton and McCallum, 2011, Section 2.2].

3.3 Maximum Entropy Markov Models

The HMMs are a well-established technique for sequence tagging but suffer from the sensitivity to overlapping features as discussed above in Section 3.2.3. To this end, this section discusses another graphical model proposed for sequence tagging, the maximum entropy Markov model (MEMM) [Ratnaparkhi, 1996, McCallum et al., 2000]. In contrast to HMMs which model the joint distribution over the input and output variables x and y, the MEMMs aim to directly estimate the conditional distribution of the output y *given* the input x. This conditional modeling is viable since the input sequence x is observed during both estimation and decoding stages. In consequence of the conditioning and applied discriminative learning, the MEMMs are able to incorporate rich, overlapping features in contrast to HMMs.

3.3.1 Model Definition

The first-order MEMMs are defined using the directed graph depicted in Figure 3.3 and factorize as

$$p(y \,|\, x; \boldsymbol{w}) = \prod_{i=1}^{T} p(y_i \,|\, y_{i-1}, x, i; \boldsymbol{w})\,, \tag{3.11}$$

where the factor distributions $p(y_i \,|\, y_{i-1}, x, i; \boldsymbol{w})$ are defined to be of the log-linear form as

$$p(y_i \,|\, y_{i-1}, x, i; \boldsymbol{w}) = \frac{\exp\Big(\boldsymbol{w} \cdot \boldsymbol{\phi}(y_{i-1}, y_i, x, i)\Big)}{Z(x, y_{i-1})}\,. \tag{3.12}$$

with $\boldsymbol{w} \cdot \boldsymbol{\phi}(.)$ denoting the inner product between a parameter vector $\boldsymbol{w}$ and a vector-valued **feature extraction function** $\boldsymbol{\phi}$. The normalization (partition) function $Z(.)$ ensures that the factors are probability distributions.The first-

order MEMM (3.11) can be generalized to nth order as

$$p(y \mid x; \boldsymbol{w}) = \prod_{i=n}^{T} p(y_i \mid y_{i-n}, \ldots, y_{i-1}, x, i; \boldsymbol{w}) \,, \tag{3.13}$$

in which case the factor distribution is written as

$$p(y_i \mid y_{i-n}, \ldots, y_{i-1}, x, i; \boldsymbol{w}) = \frac{\exp\Big(\boldsymbol{w} \cdot \boldsymbol{\phi}(y_{i-n}, \ldots, y_{i-1}, y_i, x, i)\Big)}{Z(x, y_{i-n}, \ldots, y_{i-1})} \,. \tag{3.14}$$

The purpose of the feature vector $\boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i)$ is to capture the co-occurrence behavior of the label transitions $(y_{i-n}, \ldots, y_i)$ and a set of features describing position i of the input x. Typically, one exploits **emission** and **transition** type of features analogous to the emission and transition probabilities of the HMM model (3.8). The emission feature elements of $\boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i)$ associate properties of the sequence at position i with the corresponding label and are of the form

$$\phi(y_{i-n}, \ldots, y_i, x, i) = \chi_k(x, i) \mathbb{1}(y_i = y_i') \quad \text{for} \quad \forall k \,, \forall y_i' \in \mathcal{Y} \,, \tag{3.15}$$

where the function $\mathbb{1}(y_i = y_i')$ returns one if and only if $y_i = y_i'$ and zero otherwise, that is

$$\mathbb{1}(y_i = y_i') = \begin{cases} 1 & \textbf{if} \quad y_i = y_i' \\ 0 & \textbf{otherwise} \end{cases} \,, \qquad \forall y_i, y_i' \in \mathcal{Y} \,, \tag{3.16}$$

and $\{\chi_k(x, i)\}_{k=1}^{N_\chi}$ is the set of functions characterizing the input sequence position i. For example, consider a feature

$$\chi_{k'}(x, i) = \begin{cases} 1 & \textbf{if} \quad \text{the word form } x_i = \text{Smith} \\ 0 & \textbf{otherwise} \end{cases} \,. \tag{3.17}$$

Meanwhile, the transition features are of the form

$$\begin{aligned} \phi(y_{i-j}, .., y_i, x, i) = \mathbb{1}(y_{i-j} = y_{i-j}') \mathbb{1}(y_{i-j+1} = y_{i-j+1}') \ldots \mathbb{1}(y_i = y_i') \\ \text{for} \quad \forall y_{i-j}, y_{i-j}', .., y_i, y_i' \in \mathcal{Y} \,, \forall 1 \leq j \leq n \end{aligned} \tag{3.18}$$

and capture the dependency structure between adjacent labels irrespective of the input x.

3.3.2 Learning and Decoding

The model parameters $\boldsymbol{w}$ in the factor distributions (3.14) are chosen to maximize the model likelihood. Given the log-linear form, maximizing the likelihood is equivalent to the maximum entropy solution.[2] However, in contrast

[2]The equivalence of maximum likelihood and maximum entropy solutions is generally true for models of the log-linear family.

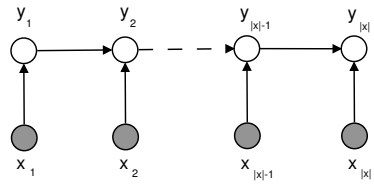

Figure 3.3. The first-order maximum entropy Markov model (MEMM) for input and output sequences $x = (x_1, x_2, \ldots, x_T)$ and $y = (y_1, y_2, \ldots, y_T)$, respectively. The input variables $x = (x_1, x_2, \ldots, x_T)$ are assumed observed during both estimation and decoding and are denoted with darkened nodes.

to HMMs, this learning problem can not be performed in a closed form and must be solved using iterative algorithms, such as gradient descent methods [Byrd et al., 1995]. In consequence, MEMM training is computationally more costly compared to HMMs. Subsequent to training, the MEMMs decode test instances using the Viterbi search identically to the HMMs.

3.3.3 Label Bias

In contrast to HMMs, the MEMMs can exploit heavily dependent and overlapping features defining arbitrarily complex functions of the input x. This is because the discriminative estimation procedure of the factors (3.14) effectively ignores overly noisy, or non-informative, features. However, the MEMMs suffer from a profound limitation in that they can be biased towards states with few successor states. For a detailed discussion on this **label bias** problem, see [Lafferty et al., 2001].

3.4 Conditional Random Fields

The HMMs are a well-established techique for sequence tagging but suffer from the sensitivity to overlapping features as discussed in Section 3.2.3. Meanwhile, the MEMMs are able to accommodate the rich feature presentations but suffer from the label bias problem. Therefore, we next consider a third sequence modeling approach based on conditional random fields (CRFs). The CRF model was originally presented for tagging and segmentation in natural language processing by Lafferty et al. [2001] to address the problems associated with HMMs and MEMMs.

The model definition of the CRFs differ from HMMs and MEMMs in two critical aspects. First, instead of modeling the joint distribution $p(x, y)$ as in the generative HMM case, the CRFs aim to directly model the **conditional**

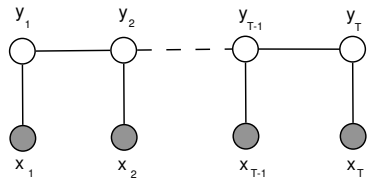

Figure 3.4. The first-order conditional random field model for input and output sequences $x = (x_1, x_2, \ldots, x_T)$ and $y = (y_1, y_2, \ldots, y_T)$, respectively. The input variables $x = (x_1, x_2, \ldots, x_T)$ are assumed observed during both estimation and decoding and are denoted with darkened nodes.

distribution $p(y|x)$ similarly to the MEMMs. Second, while the HMMs and MEMMs were defined using a directed graph, a Bayesian network, the CRFs utilize an undirected graph, a Markov random field.

Similarly to MEMMs, the advantage of employing the discriminative learning approach is that the resulting CRF model becomes very robust against noisy features. On the other hand, they avoid the label bias problem inherent to the MEMM approach. As a downside, the CRF model estimation is computationally more complex compared to both HMMs and MEMMs.

3.4.1 Model Definition

We begin by describing the first-order CRFs, also known the linear-chain CRFs, the discriminative equivalent of the first-order HMMs in (3.7). In what follows, we again denote the input sequence $x = (x_1, x_2, \ldots, x_T)$ and a corresponding sequence of output variables $y = (y_1, y_2, \ldots, y_T)$.

Given the input x and output y, the first-order CRF model is obtained by defining a first-order Markov random field over the output y and by conditioning the output globally on the input x. The model is presented graphically in Figure 3.4, and factorizes as

$$p\left(y \,|\, x; \boldsymbol{w}\right) = \frac{1}{Z(x)} \prod_{i=2}^{T} \exp\Big(\boldsymbol{w} \cdot \boldsymbol{\phi}(y_{i-1}, y_i, x, i)\Big)\,, \tag{3.19}$$

where $\boldsymbol{w} \cdot \boldsymbol{\phi}(.)$ denotes the inner product between a parameter vector $\boldsymbol{w}$ and a vector-valued **feature extraction function** $\boldsymbol{\phi}$. In contrast to the MEMMs (3.13), the normalization is performed over the complete input x. The first-order CRF model (3.19) can be generalized to nth order as

$$p\left(y \,|\, x; \boldsymbol{w}\right) \propto \prod_{i=n+1}^{T} \exp\Big(\boldsymbol{w} \cdot \boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i)\Big)\,. \tag{3.20}$$

The feature vector $\boldsymbol{\phi}(y_{i-n}, \ldots, y_{i-1}, x, i)$ is constructed identically to the MEMMs as described in Section 3.3.

3.4.2 Exact Learning and Decoding

Given some model parameters $\boldsymbol{w}$, the CRF model (3.20) decodes input instances x by solving the MAP optimization problem

$$z = \arg\max_{y'} p\left(y' \mid x; \boldsymbol{w}\right). \tag{3.21}$$

using Viterbi search [Lafferty et al., 2001] identically to the HMMs and MEMMs. In the remainder of this section, we discuss two criteria for learning the parameters $\boldsymbol{w}$ from an annotated data set $\mathcal{D}$.

Conditional Maximum Likelihood Learning We begin by discussing CRF estimation using the (conditional) maximum likelihood criterion. This approach was employed in the original work of Lafferty et al. [2001]. The maximum likelihood solution is attained by solving a minimization problem

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}'} -\sum_{i=1}^{N_{\mathcal{D}}} \log p(y^{(i)} \mid x^{(i)}; \boldsymbol{w}'). \tag{3.22}$$

This minimization problem is **convex**, that is, it has a unique solution, and can be solved using standard optimization algorithms, such as **iterative scaling** [Lafferty et al., 2001] or **gradient descent** [Malouf, 2002, Vishwanathan et al., 2006] methods. In order to avoid overfitting to training data, the minimized objective function in (3.22) is augmented with additive l_1 or l_2 regularization terms [Vail et al., 2007]. These penalty terms are associated with a hyper-parameter C which controls the amount of performed regularization.

Perceptron Learning This section describes estimation of CRF model parameters employing the **perceptron algorithm**, an online learning algorithm for supervised classification. Perceptron learning has a long history in machine learning beginning from the original work of Rosenblatt [1958]. A perceptron algorithm variant for CRF learning was first formulated by Collins [2002]. Note that the CRFs correspond to discriminatively trained HMMs and Collins [2002] employs the latter terminology.

In the previous section, we employed the factorized CRF notation (3.20).In what follows, we write the model in an equivalent but more compact manner as

$$p\left(y \mid x; \boldsymbol{w}\right) \propto \exp\Big(\boldsymbol{w} \cdot \boldsymbol{\Phi}(x, y)\Big), \tag{3.23}$$

where the equivalence holds when $\boldsymbol{\Phi}(x, y) = \sum_{i=n+1}^{T} \boldsymbol{\phi}(y_{i-n}, \ldots, y_i, x, i)$. This notation reflects the cited literature on perceptron learning [Collins, 2002, Collins and Roark, 2004, Huang et al., 2012].

Input: training data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$
Output: model parameters $\boldsymbol{w}$

1: **repeat until convergence**
2: **for** (x, y) **in** $\mathcal{D}$ **do**
3: $z \leftarrow \arg\max_u \boldsymbol{w} \cdot \boldsymbol{\Phi}(x, u)$
4: **if** $z \neq y$ **then**
5: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\Phi}(x, y) - \boldsymbol{\Phi}(x, z)$

Figure 3.5. The structured perceptron algorithm [Collins, 2002].

The **structured perceptron** algorithm, originally presented by Collins [2002], is depicted in Figure 3.5. Intuitively, the algorithm operates by tagging the training instances a single instance at a time using the Viterbi search (line 3 in Figure 3.5) and modifies the parameters using a simple additive update in case the prediction is incorrect (line 5). As shown by Collins [2002], if the data set $\mathcal{D}$ is linearly separable given the feature presentation $\boldsymbol{\Phi}$, the algorithm is guaranteed to find parameters which separate the data, that is, yield no incorrect predictions on the data set. The perceptron algorithm will generalize well to test instances with a high probability if it is able to find parameters which yield a small number of erroneous predictions on the training set [Freund and Schapire, 1999, Collins, 2002].

In order to reduce overfitting the training data, the perceptron algorithm employs a parameter averaging approach. The averaging was originally proposed by Freund and Schapire [1999] as an approximation to the voting scheme presented in the same work. In averaging, the parameter vector returned by the perceptron algorithm is the average of the parameters obtained after each update during learning. More formally, denoting the parameter vector after ith update as $\boldsymbol{w}^{(i)}$, the returned parameters are

$$\boldsymbol{w} = \frac{1}{N_{up}} \sum_{i=0}^{N_{up}} \boldsymbol{w}^{(i)} \,, \tag{3.24}$$

where N_{up} is the total number of performed updates before termination. An efficient implementation technique for the approach can be found in [Daumé III, 2006, page 19]. Appealingly, the straightforward averaging requires no introduction or tuning of hyper-parameters in contrast to using regularization terms as in maximum likelihood learning.

Compared to the conditional maximum likelihood estimation criterion, the perceptron algorithm has some advantages. First, the algorithm is inherently online (stochastic) which usually results in accelerated learning on large, re-

dundant data sets as noted by Vishwanathan et al. [2006]. Second, the parameter updates in the perceptron algorithm are inherently sparse, resulting in additional savings in running time as well as compact final parameter settings. Third, the perceptron algorithm is simple to implement and contains only a single tunable hyper-parameter, namely, the number of passes made over the training set. On the other hand, the perceptron algorithm does have an apparent shortcoming in that it lacks an explicit means of pushing down the probabilities of incorrect (or unlikely) label assignments. This is because the algorithm stops updating the model parameters as soon as the probability for the correct assignment y is higher than any other assignment. Meanwhile, the maximum likelihood criterion has the advantage of inherently assigning low probabilities for unlikely label sequences. Moreover, the maximum likelihood parameters of the CRF model coincide with the maximum entropy solution [Jaynes, 1957], which provides additional theoretical appeal for the approach.

Finally, the perceptron algorithm can be interpreted as a classification algorithm in itself. However, it can also be seen as an alternative learning approach to maximum likelihood CRF estimation optimizing the zero-one loss, that is, for each training instance $x^{(i)}, i = 1 \dots N_{\mathcal{D}}$, the loss is one if $z^{(i)} \neq y^{(i)}$ and zero otherwise. The latter interpretation is employed throughout this thesis. This, however, can lead to potential confusion since the parameters yielded by the perceptron algorithm do not have a similar probabilistic interpretation as the maximum likelihood parameters. On the other hand, since subsequent to training we are merely interested in finding the highest scoring label sequences provided by the model, the lack of probabilistic interpretation is not a major issue.

3.4.3 Approximative Learning and Decoding

As discussed in Section 3.4.2, the CRF model parameters are estimated using iterative algorithms. These algorithms operate by performing repeated inference using forward-backward (maximum likelihood estimation) or Viterbi (perceptron estimation) algorithms over training instances in addition to making multiple passes over the training data set. Inconveniently, in some cases, learning can become infeasible or impractically slow due to the costly inference procedure. For example, consider morphological tagging in presence of large label sets. Therefore, in what follows, we discuss **approximative** parameter estimation approaches, the purpose of which is to lower the complexity of learning by searching for approximative solutions instead of exact ones.

Beam Search We begin by discussing the beam search, a well-known heuristic optimization technique employed to search the highest scoring sequence z with a lower computational cost compared to the exact search (using the Viterbi algorithm). The key idea is to preserve only the highest scoring (most probable) histories for each sequence position i while pruning out the rest. Typically, the number of preserved histories, referred to as the **beam width**, is fixed. However, as presented by Pal et al. [2006], it is also possible to adapt the beam width during the search by minimizing the Kullback-Leibler divergence between the inferred marginal distributions and true marginals over label configurations. Moreover, by applying this **minimum divergence beam** method to the standard forward-backward algorithm, one obtains the **sparse forward-backward** algorithm. Pal et al. [2006] then apply the sparse forward-backward inference to accelerate the search for approximative maximum likelihood parameters. Subsequent to learning, test instances are decoded using the beam search with adapted beam widths.

The beam search has also been extensively employed in combination with the perceptron algorithm of Collins [2002] (Figure 3.5) and found useful in several syntactic language processing applications [Collins and Roark, 2004, Zhang and Clark, 2011, Huang et al., 2012]. While directly applying beam search in the structured perceptron algorithm in Figure 3.5 can work well in practice, this approach has some theoretical drawbacks. In particular, the perceptron convergence guarantee provided by Collins [2002] breaks down. This is because the correct label sequence y may be dropped out from the beam during the search although having the highest score given model parameters $\boldsymbol{w}$. However, in order to accommodate the inexact beam search, one can employ the **violation-fixing** perceptron algorithm presented by Huang et al. [2012] which preserves the perceptron convergence guarantee under the assumption of **beam-separable** data. The approach of Huang et al. [2012] encompasses the original algorithm of Collins [2002] in case the beam width is set to infinite, that is, exact search is applied. Again, subsequent to learning, test instances are decoded using the beam search with adapted beams widths.

Cascaded Models In a recent work, Weiss et al. [2012] discussed acceleration of structured predictors, such as CRFs, by learning a **cascade** of classifiers, in which the input is passed through a series of models of increasing complexity until the final prediction is produced. In this general cascade system, the computational cost of the complete system remains low as the search space of each complex model is restricted using the predictions of the less complex models. The cascade is applied during the learning as well as the decoding phase,

thus making both phases computationally viable. Moreover, Weiss et al. [2012] provide generalization bounds for both accuracy and efficiency of their method.

In an independent work, Müller et al. [2013] presented an approximative high-order CRF estimation technique utilizing a cascade of CRF models utilizing a coarse-to-fine decoding technique [Charniak and Johnson, 2005, Rush and Petrov, 2012]. Similarly to the more general framework of Weiss et al. [2012], in the approach of Müller et al. [2013] the input is passed through a series of CRFs models of increasing complexity, beginning from a zeroth-order model, until the final prediction is produced. Unlike Weiss et al. [2012], Müller et al. [2013] do not provide nor discuss any performance guarantees to their approximation. Nevertheless, the method was shown to achieve empirical success in morphological tagging in presence of high number of labels.

Pseudo-Likelihood Learning Finally, we discuss the classic pseudo-likelihood estimation approach originally presented by Besag [1975]. Intuitively, pseudo-likelihood estimation operates by performing the necessary inference over single graph nodes while keeping the remaining nodes fixed to their true values. Formally, the pseudo-likelihood solution is attained by solving an optimization problem

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}'} \prod_{i=1}^{N_{\mathcal{D}}} \prod_{j=1}^{T^{(i)}} p(y_j^{(i)} \mid y_{-j}^{(i)}, x^{(i)}; \boldsymbol{w}') \,, \tag{3.25}$$

where y_{-j} denotes all the variables in sequence y apart from y_j. The appeal of this approach is that, since the inference is performed over single variables, the complexity of learning is reduced to linear in the number of labels in label set. Similarly to the exact maximum likelihood estimation (3.22), the pseudo-likelihood optimization problem (3.25) is convex and can be solved using standard optimization algorithms.

The pseudo-likelihood is known to be a consistent estimator [Mozeika et al., 2014], that is, the pseudo-likelihood solution coincides with the exact maximum likelihood solution given infinite amount of data. This asymptotic property, however, does not guarantee success on finite-sized real life data sets. For example, in the experiments reported by Sutton and McCallum [2009], the pseudo-likelihood approach was shown to be successful in morphological tagging but fail in segmentation tasks including named entity recognition and phrase chunking.

Finally, subsequent to model estimation using the pseudo-likelihood criterion, the model is applied to training instances in a standard manner using the Viterbi search.

3.4.4 Semi-Supervised Learning and Decoding

The discussion on CRF model estimation in Sections 3.4.2 and 3.4.3 assumed a training set of input-output pairs, $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$. Next, we discuss extending this supervised learning setting by additionally utilizing an unannotated data set $U = \{x^{(j)}\}_{j=1}^{N_\mathcal{U}}$. The inclusion of the unannotated data is commonly motivated by the fact that obtaining annotated data requires manual labor by a human expert and can, therefore, be both time consuming as well expensive. The first notable work on this learning setting for language processing were presented by Yarowsky [1995] and Blum and Mitchell [1998] in the form of the self-training and co-training algorithms, respectively. Since the early work, semi-supervised learning has been applied extensively in numerous applications using a wide range of proposed techniques [Zhu and Goldberg, 2009].

As for semi-supervised learning of the CRF model, the most straightforward manner of utilizing U is through a **feature expansion** scheme exploiting unsupervised learning. For example, consider morphological tagging of English using the Penn Treebank annotation:

About	a	quarter	has	already	been	reallocated	.
↓	↓	↓	↓	↓	↓	↓	↓
IN	DT	NN	VBZ	RB	VBN	VBN	PUNCT

Then, consider the unsupervised learning task of part-of-speech induction, in which word forms are assigned to clusters according to their contextual similarity, where the context comprises neighboring word in running text. This type of learning problem has received much attention within the language processing community [Brown et al., 1992, Honkela et al., 1995, Schütze, 1995, Clark, 2003, Christodoulopoulos et al., 2010, Lamar et al., 2010, Lee et al., 2010, Christodoulopoulos et al., 2011]. The learning relies on the insight that two words which can be replaced with each other in a sentence without breaking the grammaticality of the sentence belong to the same part-of-speech class with a high probability. Assuming M induced clusters, we can define a set of M functions $\{\upsilon_m(x, i)\}_{m=1}^{M}$, where $\upsilon_m(i)$ returns 0 or 1 if the word at position i in sentence x belongs to the cluster m, as in

i	1	2	3	4	5	6	7	
x	About	a	quarter	has	already	been	reallocated	.
$\upsilon_1(x,i)$	0	1	0	0	1	0	0	0
$\upsilon_2(x,i)$	1	0	0	0	0	0	0	0
...								
$\upsilon_M(x,i)$	0	0	0	1	0	0	0	0

Now, given a set of the M functions $\{\upsilon_m(i)\}_{m=1}^{M}$, we can define variants of the

emission features in (3.15) as

$$\phi(y_{i-n}, \dots, y_i, x, i) = \upsilon_m(x, i)\chi_k(x, i)\mathbb{1}(y_i = y'_i)$$
$$\text{for} \quad \forall m \in 1..M\,, \forall k \in 1..N_\chi\,, \forall y'_i \in \mathcal{Y}\,. \tag{3.26}$$

By adding the expanded features of form (3.26), the CRF model learns to associate the output of the unsupervised algorithms in relation to the input feature set $\{\chi_k(x, i)\}_{k=1}^{N_\chi}$ and label set $\mathcal{Y}$. Similarly, expanded transition features can be written as

$$\phi(y_{i-n}, \dots, y_i, x, i) = \upsilon_m(x, i)\mathbb{1}(y_{i-1} = y'_{i-1})\mathbb{1}(y_i = y'_i)$$
$$\text{for} \quad \forall m \in 1..M\,, \forall y'_i, y'_{i-1} \in \mathcal{Y}\,. \tag{3.27}$$

Note that the functions $\upsilon_m(x, i)$ in (3.26) and (3.27) do not need to be one-hot nor binary, that is, for a given word form, υ_m can be non-zero for $m = 1 \dots M$. To this end, instead of hard clusters, one could employ **word embeddings** which correspond to these types of distributed representations [Bengio et al., 2006, Collobert and Weston, 2008, Luong et al., 2013].

After defining the augmented feature set, the CRF model parameters can be estimated in a standard manner on the annotated training data set. Subsequent to CRF training, the unsupervised model is applied on the test instances in order to allow the feature set augmentation and standard decoding with the estimated CRF model. Feature expansion schemes similar to the one described here have been shown to yield state-of-the-art performance in several language processing sequence tagging and segmentation tasks, including morphological tagging [Östling, 2012], named entity recognition and phrase chunking [Turian et al., 2010], and Chinese word segmentation [Wang et al., 2011, Sun and Xu, 2011]. Nevertheless, it should be mentioned that there does exist numerous other approaches proposed for semi-supervised learning of CRFs, exemplified by minimum entropy regularization [Jiao et al., 2006], generalized expectations criteria [Mann and McCallum, 2008], and rate distortion approach [Wang et al., 2009]. In addition, CRFs are amenable to general purpose semi-supervised learning approaches, such as the self-training algorithm of Yarowsky [1995].

3.5 Summary

In this section, we discussed graphical models for sequence tagging and segmentation, including hidden Markov models (HMMs), maximum entropy Markov models (MEMMs), and conditional random fields (CRFs). A specific

emphasis was put on the CRFs which enjoy several theoretical and practical advantages over the HMM and MEMM frameworks in addition to wide popularity. In the remainder of this work, we discuss the author's contributions to morphological segmentation and morphological tagging prolems employing the CRF modeling framework.

4. Contributions to Morphological Segmentation

This chapter provides a summarizing discussion on the current author's contributions to statistical morphological segmentation. In particular, we consider performing the segmentation using the conditional random field (CRF) model in a semi-supervised learning setting. In this setting the available training data consists of a small number of annotated segmentation examples and a large number of unannotated raw word forms. In what follows, we first review methodology in Section 4.1 including a literature overview of previous work. Subsequently, we discuss empirical results in Section 4.2. The presentation is based on Publications I-III.

4.1 Methodology

In this section, we review methods proposed for statistical morphological segmentation. The focus of the discussion is on a learning setting, in which the available data consists of a small number of annotated word forms $\mathcal{D}$ and a large number of unannotated word forms $\mathcal{U}$. Given this setting, one can in principle employ unsupervised, supervised, or semi-supervised learning approaches. However, by learning in an unsupervised manner from $\mathcal{U}$, one neglects information provided by the annotated set $\mathcal{D}$. Similarly, by learning from $\mathcal{D}$ in a supervised manner, one neglects information extractable from $\mathcal{U}$. Therefore, in order to utilize all the available data, one must employ semi-supervised learning. In what follows, we provide an overview of existing semi-supervised morphological segmentation approaches in Section 4.1.1. Subsequently, in Section 4.1.2, we discuss how to perform supervised and semi-supervised morphological segmentation using the conditional random field (CRF) model.

4.1.1 Literature Overview

In what follows, we review existing morphological segmentation methods in a chronological order. We will first provide brief descriptions of the individual techniques and then make general observations on the core properties of the systems.

Morfessor. The Morfessor method family [Creutz and Lagus, 2005, Kohonen et al., 2010, Grönroos et al., 2014] is based on a generative probabilistic model which defines a joint probability distribution over the unannotated word forms $\mathcal{U}$ and the corresponding segmentations. The model generates the observed word forms by concatenating morphs. The morphs are stored in a **morph lexicon** which defines the probability of each morph given some model parameters. The Morfessor learning problem is to find a morph lexicon which yields an optimal balance between encoding the observed word forms concisely and, at the same time, having a concise morph lexicon. To this end, most Morfessor variants utilize a prior distribution over morph lexicons, derived from the Minimum Description Length principle [Rissanen, 1989] which favors lexicons containing a small number of short morphs. The earliest Morfessor method, referred to as Morfessor Baseline [Creutz and Lagus, 2005], was originally proposed for unsupervised learning. Subsequently, it has been extended to semi-supervised learning by Kohonen et al. [2010]. Meanwhile, the most recent Morfessor variant, coined the Morfessor FlatCat [Grönroos et al., 2014], extends the unsupervised Morfessor CatMAP model [Creutz and Lagus, 2005] to semi-supervised learning in a similar manner to [Kohonen et al., 2010].

Generative Log-Linear Approach. Similarly to the Morfessor model family, the generative log-linear model of [Poon et al., 2009] defines a joint probability distribution over the unannotated word forms $\mathcal{U}$ and the corresponding segmentations. The distribution is log-linear in form. Again, similarly to the Morfessor framework, Poon et al. [2009] learn a morph lexicon which is subsequently used to generate segmentations for new word forms. The learning is controlled using prior distributions on both corpus and lexicon (motivated by the MDL principle), which penalize exceedingly complex morph lexicon and exceedingly segmented corpus, respectively. The model can be trained in an unsupervised or semi-supervised manner.

Promodes. The Promodes system of Spiegler and Flach [2010] defines a family of generative probabilistic models for recovering segment boundaries in an unsupervised fashion. The Promodes models define a joint distribution over words and segmentations and can be trained utilizing both annotated and unannotated data, $\mathcal{D}$ and $\mathcal{U}$.

Hidden Markov Model. The algorithm of Kılıç and Bozsahin [2012] is based on a generative HMMs, in which the HMMs learn to generate morph sequences for given word forms in a semi-supervised fashion. The algorithm learns mainly from unannotated data $\mathcal{U}$ and incorporates supervision from the annotated corpus in the form of manually selected statistics.

Adaptor Grammars. Sirts and Goldwater [2013] presented work on minimally-supervised morphological segmentation using the Adaptor Grammar approach of Johnson et al. [2006]. The Adaptor Grammars constitute a non-parametric Bayesian modeling framework applicable for learning latent tree structures over an input corpus of strings. They can be used to define morphological grammars of different complexity, starting from the simplest grammar where each word is just a sequence of morphs and extending to more complex grammars, where each word consists, for example, of zero or more prefixes, a stem, and zero or more suffixes. The actual forms of the morphs are learned from the data and, subsequent to learning, employed to generate segmentations for new word forms. In this general approach the Adaptor Grammars are similar to the Morfessor family. A major difference, however, is that the morphological grammar is not hard-coded but instead specified as an input to the algorithm. This allows different grammars to be explored in a flexible manner. As another point of difference, the Morfessor variants return single models corresponding to the MAP point-estimates, while Adaptor Grammars operate with full posterior distributions over all possible models. Finally, prior to the work by Sirts and Goldwater [2013], the Adaptor Grammars were successfully applied in a related task of segmenting utterances into words [Johnson, 2008, Johnson and Goldwater, 2009, Johnson and Demuth, 2010]. While the Adaptor Grammar framework was originally designed for the unsupervised learning setting, Sirts and Goldwater [2013] showed improved performance using a semi-supervised extension.

All the methods discussed above [Creutz et al., 2007, Poon et al., 2009, Spiegler and Flach, 2010, Kılıç and Bozsahin, 2012, Sirts and Goldwater, 2013] model

the joint distribution of word forms and their corresponding segmentations, that is, these methods generate both word forms and segmentations. Such generative models are readily applicable for unsupervised learning from large unannotated word lists. The authors then seek improvements in accuracy using semi-supervised learning, that is, by extending the methods to utilize (a small amount of) annotated word forms in addition to the unannotated data. To this end, [Creutz et al., 2007, Poon et al., 2009, Spiegler and Flach, 2010, Kılıç and Bozsahin, 2012, Sirts and Goldwater, 2013] contain a wide range of varying techniques. In the most straightforward approach, the segmentation is fixed to its correct value for the labeled word forms $\mathcal{D}$, as exemplified by [Poon et al., 2009, Spiegler and Flach, 2010, Sirts and Goldwater, 2013]. In addition, the availability of the annotated data $\mathcal{D}$ makes it possible to apply discriminative learning techniques to generative models. In particular, model hyper-parameters can be selected to optimize segmentation performance, rather than some generative objective, such as likelihood. Special cases of hyper-parameter selection include the weighted objective function [Kohonen et al., 2010], data selection [Virpioja et al., 2011a, Sirts and Goldwater, 2013], and grammar template selection [Sirts and Goldwater, 2013].

4.1.2 Morphological Segmentation using Conditional Random Fields

The work presented in this section stemmed from the observation that the existing segmentation systems [Creutz et al., 2007, Poon et al., 2009, Spiegler and Flach, 2010, Kılıç and Bozsahin, 2012, Sirts and Goldwater, 2013] were based on generative semi-supervised modeling frameworks. However, what was lacking from the literature was a study on how these methods would fare against a straightforward segmentation approach using a state-of-the-art supervised segmentation model learned discriminatively solely from the annotated training data $\mathcal{D}$. The purpose of Publication I was to extend the existing literature from this point of view. In contrast to employing a range of discriminative learning "tricks", such as weighted objective functions or data selection, for the generative models discussed above, the key idea of this work was to employ a CRF model to directly estimate a conditional distribution of segmentation *given* a word form in a discriminative manner from $\mathcal{D}$. This approach is viable since the earlier work explicitly assume that word forms are observed during both model learning and segmentation of novel word forms. Subsequently, in Publication II, the CRF segmentation approach was extended from supervised to semi-supervised learning. In what follows, we will review the methodology discussed in these publications.

Supervised Learning In general, a sequence segmentation problem can be represented as a sequence tagging task by employing a number of varying label set schemes. In the simplest case, segmentation can be performed by assigning each symbol x_i, such as characters or words, in an input sequence $x = (x_1, x_2, \ldots, x_T)$ to one of two labels

B	beginning of a segment
M	middle of a segment

However, the label set can be chosen to be more fine-grained. For example, in their CRF-based Chinese word segmentation system, Zhao et al. [2006] assign each input character into one of the following six labels:

S	beginning of a single character word
B	beginning of a multi character word
M_1	first character position of a multi character word
M_2	second character position of a multi character word (if not last position)
E	last character position of a multi character word
M	middle of a segment (if not first, second, or last position)

Essentially, by defining more fine-grained sets, one captures increasingly eloquent structure but may overfit model to the training data due to increasingly sparser statistics. Therefore, the appropriate granularity depends largely on the amount of available training data and the alphabet size (the set from which input symbols x_i are drawn). Based on Publications I–III, a suitable compromise between granularity and data density for the morphological segmentation task in the considered learning setting is provided by, for example, a set of three

B	beginning of a multi-character morph
M	middle of a multi-character morph
S	single-character morph

or four labels:

B	beginning of a multi-character morph
M	middle of a multi-character morph
E	end of a multi-character morph
S	single-character morph

For instance, using the latter set, one can represent the segmentation of the Finnish word *asuinalueilla* (*in residential areas*) with a correct analysis *asuin+alue+i+lla* as

a	s	u	i	n	a	l	u	e	i	l	l	a
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
B	M	M	M	E	B	M	M	E	S	B	M	E

left substrings	position	right substrings
^	t	t, ta, tal, talo, talot, talot$
^t, t	a	a, al, alo, alot, alot$
^ta, ta, a	l	l, lo, lot, lot$
^tal, tal, al, l	o	o, ot, ot$
^talo, talo, alo, lo, o	t	t, t$

Table 4.1. Example of left and right substring feature extraction from the word form *talot* (*houses*). The symbols ^ and $ mark word beginning and end, respectively.

Subsequent to defining the label set, the modeling problem consists of choosing the emission features included in the feature set χ in Equation (3.15). Intuitively, one could describe each letter position using overlapping left and right substring contexts. For example, consider extracting the substrings from the Finnish word form *talot* (*houses*) as presented in Table 4.1. Indeed, as discussed in Publication I, this simple and straightforward approach works well in practice given that the maximum substring length δ_{max} is restricted to avoid overfitting to training data.

Finally, from early on, the CRF model was shown to perform well in tasks related to morphological segmentation, including Chinese word segmentation [Peng et al., 2004] and chunking [McCallum and Li, 2003]. Thus, the CRFs were also a natural choice for the morphological segmentation task. However, it should be noted that the learning setting applied in these tasks differ somewhat. In particular, in phrase chunking and Chinese word segmentation, the number of annotated training instances is typically hundreds of thousands or millions [Tjong Kim Sang and Buchholz, 2000, Sproat and Emerson, 2003, Emerson, 2005]. In contrast, consider the Morpho Challenge data set [Kurimo et al., 2010] which contains a set of 1,000 annotated word instances. On the other hand, the number of different input symbols x_i in morphological segmentation is equal to the number of letters in the alphabet and thus rather small. Meanwhile, for example in Chinese word segmentation, the alphabet typically contains thousands different characters resulting in more severe data sparsity issues. Due to these issues, it was not completely clear whether the CRF model would be successful in the morphological segmentation task prior to work presented in Publication I.

Semi-Supervised Learning In what follows, we discuss how to extend the CRF-based segmentation approach described above to utilize unannotated data $\mathcal{U}$ in a semi-supervised manner instead of learning the model in a supervised manner solely from the annotated set $\mathcal{D}$. As mentioned in Section 3.4.4, a range

of feature expansion schemes discussed in the same section have been shown to work well for several segmentation tasks including Chinese word segmentation [Wang et al., 2011, Sun and Xu, 2011] and chunking [Turian et al., 2010]. Therefore, in this section, we study how to extend the CRF-based morphological segmentation model to semi-supervised learning similarly to [Turian et al., 2010, Wang et al., 2011, Sun and Xu, 2011] using the feature expansion scheme described in Section 3.4.4.

We begin by expanding the CRF feature set utilizing predictions of the unsupervised Morfessor [Creutz et al., 2007] and Adaptor Grammar [Sirts and Goldwater, 2013] methods. To this end, one must first learn the Morfessor and Adaptor Grammar models from the unannotated training data, and then apply the learned models to the word forms in the annotated training set $\mathcal{D}$. Assuming the annotated training data includes the Finnish word *epäkypsät* (*immature (plural)*), the Morfessor and Adaptor Grammar algorithms might, for instance, return (partially correct) segmentations *epä+kypsät* and *epäkypsä+t*, respectively. We present these segmentations by defining a set of two functions υ_1, υ_2 corresponding to the Morfessor and Adaptor Grammar models, respectively. Subsequently, υ_m, $m \in \{1, 2\}$, is then defined to return 0 or 1 if the position i is in the middle of a segment or in the beginning of a segment, respectively, as in

i	1	2	3	4	5	6	7	8	9
x_i	e	p	ä	k	y	p	s	ä	t
$\upsilon_1(i)$	1	0	0	1	0	0	0	0	0
$\upsilon_2(i)$	1	0	0	0	0	0	0	0	1

These functions can then be incorporated into the CRF model as described in Section 3.4.4. In consequence, the CRF model learns to associate the outputs of these unsupervised algorithms in relation to the substring contexts defined by the features described in previous section.

Next, we make use of the fact that the output of the unsupervised algorithm does not have to be binary (zeros and ones). To this end, we consider the classic letter successor variety (LSV) scores presented originally by Harris [1955]. The LSV scores utilize the insight that the predictability of successive letters should be high within morph segments and low at the boundaries. In consequence, a high variety of letters following a prefix indicates a high probability of a boundary. While the LSV scores track predictability given prefixes, the same idea can be utilized for suffixes, providing the letter predecessor variety scores (LPV). Subsequent to augmenting the feature set using the LSV and LPV scores, the CRF model learns to associate high successor and predecessor

	Arabic	English	Estonian	Finnish	Hebrew	Turkish
train (an.)	1,000	1,000	1,000	1,000		
train (unan.)	384,903	3,908,820	2,206,719	617,298		
devel.	694	800	835	763		
test	10×1,000	10×1,000	10×1,000	10×1,000		

Table 4.2. Number of word types in the data sets.

variety values (low predictability) to high probability of a segment boundary. Note that, instead of employing the plain LSV/PSV scores, it is preferable to utilize the improved variants presented by Çöltekin [2010], in which one first normalizes the scores by the average score at each position i, and subseqently logarithmizes the normalized values.

The computational cost of the feature set augmentation approach described above is dominated by the computational overhead of the unsupervised methods. This is because the CRF parameter estimation is still based on the small amount of labeled examples as described in Section 3.4.

4.2 Experiments

In this section, we discuss empirical results obtained using the conditional random field approach described in Section 4.1.2. The presentation is based on Publications I and III.

4.2.1 Data

We discuss experiments conducted on six languages, namely, Arabic, English, Estonian, Finnish, Hebrew, and Turkish. The English, Finnish, and Turkish data are from the Morpho Challenge 2009/2010 data set [Kurimo et al., 2009, 2010] discussed in Section 2.2.2. The annotated Estonian data set is acquired from a manually annotated, morphologically disambiguated corpus.[1] Meanwhile, the unannotated word forms are gathered from the Estonian Reference Corpus [Kaalep et al., 2010]. The Arabic and Hebrew data sets are incorporated in the Hebrew Bible parallel corpus introduced by Snyder and Barzilay [2008] and applied later by Poon et al. [2009]. It contains 6,192 parallel phrases in Hebrew, Arabic, Aramaic, and English. Table 4.2 shows the total number of instances available for model estimation and testing.

[1] Available at `http://www.cl.ut.ee/korpused/morfkorpus/index.php?lang=en`

4.2.2 Methods

For English, Estonian, Finnish, and Turkish, we compare the supervised and semi-supervised CRF models with semi-supervised variants of the Morfessor method family, namely, the semi-supervised Morfessor Baseline [Kohonen et al., 2010] and the semi-supervised Morfessor FlatCat [Grönroos et al., 2014], and semi-supervised variants of the Adaptor Grammar method, namely, semi-supervised Adaptor Grammar and Adaptor Grammar Select [Sirts and Goldwater, 2013]. The supervised CRF model follows the presentation in Section 4.1.2. For semi-supervised learning of CRFs, we utilize log-normalized successor and predecessor variety scores and binary Morfessor Baseline and Adaptor Grammar features following the presentation in Section 4.1.2. The CRF model parameters are estimated using the standard averaged perceptron algorithm [Collins, 2002]. The presentation is based on the experimental section of Publication III.

For Arabic and Hebrew, we compare the CRF model with the semi-supervised log-linear model of Poon et al. [2009] and the semi-supervised Morfessor Baseline [Kohonen et al., 2010]. The presentation on these languages is based on the experimental section of Publication I. Note that Arabic and Hebrew results do not include semi-supervised CRFs. This is because initial experiments with this learning approach indicated that the small unannotated sets available for unsupervised learning did not yield useful successor and predecessor variety scores nor Morfessor Baseline segmentations. In other words, for Arabic and Hebrew, we only report the supervised CRF learning results presented in Publication I.

4.2.3 Evaluation

The word segmentations are evaluated by comparison with reference segmentations using **boundary precision**, **boundary recall**, and **boundary F1-score**. The boundary F1-score, or F1-score for short, equals the harmonic mean of precision (the percentage of correctly assigned boundaries with respect to all assigned boundaries) and recall (the percentage of correctly assigned boundaries with respect to the reference boundaries):

$$\text{Precision} = \frac{C(\text{correct})}{C(\text{proposed})} \tag{4.1}$$

$$\text{Recall} = \frac{C(\text{correct})}{C(\text{reference})} \tag{4.2}$$

For English, Estonian, Finnish, and Turkish, we follow Virpioja et al. [2011b] and use type-based macro-averaged F1-scores, that is, compute the F1-scores

for each word type individually and average over all word types. However, we handle word forms with alternative analyses in a different fashion. Instead of penalizing algorithms that propose an incorrect number of alternative analyses, we take the best match over the alternative reference analyses (separately for precision and recall). This is because all the methods considered in the experiments provide a single segmentation per word form.

Because of the different treatment of alternative analyses, the results reported here are not directly comparable to the boundary F1-scores reported for the Morpho Challenge competitions [Kurimo et al., 2009, 2010]. However, the best boundary F1-scores for all languages reported in Morpho Challenge have been achieved with the semi-supervised Morfessor Baseline algorithm [Kohonen et al., 2010] which is included in the current experiments.

For the experiments on Arabic and Hebrew data sets, we apply token-based micro-averaged F1-scores. Employing this variant, the results obtained using CRFs are comparable with those reported by Poon et al. [2009].

For English, Estonian, Finnish, and Turkish, we establish statistical significance with confidence level 0.95 according to the standard 2-sided Wilcoxon signed-rank test performed on 10 random subsets of 1000 word forms drawn from the complete test sets (subsets may contain overlapping word forms). For Arabic and Hebrew, we compare the to the results reported by Poon et al. [2009] and, therefore, statistical significance testing is not feasible.

4.2.4 Error analysis

This section describes the error analysis following the presentation of Publication III. The purpose of the error analysis is to gain a more detailed understanding into what kind of errors the methods make, and how the error types affect the overall F1-scores. To this end, we employ a categorization of morphs into the categories PREFIX, STEM, and SUFFIX, in addition defining a separate category for DASH. For the English and Finnish sections of the Morpho Challenge data set, the segmentation gold standard annotation contain additional information for each morph, such as part-of-speech for stems and morphological categories for affixes, that allows us to assign each morph into one of the morph type categories. In some rare cases the tagging is not specific enough, and we choose to assign the tag UNKNOWN. However, as we are evaluating segmentations, we lack the morph category information for the proposed analyses. Consequently, we cannot apply a straightforward category evaluation metric, such as category F1-score. In what follows, we instead show how to use the categorization on the gold standard side to characterize the segmentation

errors.

We first observe that errors come in two kinds, **over-segmentation** and **under-segmentation**. In over-segmentation, boundaries are incorrectly assigned within morph segments, while in under-segmentation, the segmentation fails to uncover correct morph boundaries. For example, consider the English compound word form *girlfriend* with a correct analysis *girl+friend*. Then, an under-segmentation error occurs in case the model fails to assign a boundary between the segments *girl* and *friend*. Meanwhile, over-segmentation errors take place if any boundaries are assigned within the two compound segments *girl* and *friend*, such as *g+irl* or *fri+end*.

As for the relationship between these two error types and the precision and recall measures in Equations (4.1) and (4.2), we note that over-segmentation solely affects precision, whereas under-segmentation only affects recall. This is evident as the measures can be written equivalently as:

$$\text{Precision} = \frac{C(\text{proposed}) - C(\text{over-segm.})}{C(\text{proposed})} = 1 - \frac{C(\text{over-segm.})}{C(\text{proposed})} \tag{4.3}$$

$$\text{Recall} = \frac{C(\text{reference}) - C(\text{under-segm.})}{C(\text{reference})} = 1 - \frac{C(\text{under-segm.})}{C(\text{reference})} \tag{4.4}$$

In the error analysis, we employ these equivalent expressions as they allow us to examine the effect of reduction in precision and recall caused by over-segmentation and under-segmentation, respectively. We then decompose the precision reduction in Equation (4.3) as

$$\text{Precision} = 1 - \sum_{c} \frac{C(\text{over-segm.}(c))}{C(\text{proposed})} \tag{4.5}$$

where the index c takes values from the error category set {PREFIX, STEM, SUFFIX, UNKNOWN} since the segments in the category DASH cannot be segmented and do, therefore, not contribute to over-segmentation errors. The recall reduction in Equation (4.4) is decomposed as

$$\text{Recall} = 1 - \sum_{d} \frac{C(\text{under-segm.}(d))}{C(\text{reference})} \tag{4.6}$$

where the index d takes values from the error category set {STEM-SUFFIX, STEM-STEM, PREFIX-STEM, ...} since the recall is affected by the morph boundaries.

Table 4.3 shows the occurrence frequency of each boundary category, averaged over alternative analyses. Evidently, we expect the total precision scores to be most influenced by over-segmentation of STEM and SUFFIX segment types due to their high frequencies. Similarly, the overall recall scores are expected to be most impacted by under-segmentation of STEM-SUFFIX and SUFFIX-SUFFIX

Category	English		Finnish	
STEM	38608.8	(82.2%)	72666.0	(81.3%)
SUFFIX	7172.9	(15.3%)	15384.9	(17.2%)
PREFIX	1152.8	(2.5%)	946.5	(1.1%)
UNKNOWN	54.5	(0.1%)	414.0	(0.5%)
STEM-SUFFIX	5349.2	(62.6%)	9889.9	(45.8%)
SUFFIX-SUFFIX	1481.0	(17.3%)	5917.5	(27.4%)
STEM-STEM	613.4	(7.2%)	3538.0	(16.4%)
SUFFIX-STEM	n/a	n/a	1501.0	(6.9%)
CONTAINS DASH	458.0	(6.5%)	426.0	(2.0%)
PREFIX-STEM	554.3	(5.4%)	235.2	(1.1%)
OTHER	91.0	(1.1%)	105.4	(0.5%)

Table 4.3. Absolute and relative frequencies of the boundary categories in the error analysis. The numbers are averaged over the alternative analyses in the reference annotation.

boundaries. Finnish is also substantially influenced by the STEM-STEM boundary indicating that Finnish employs compounding frequently.

For simplicity, when calculating the error analysis, we forgo the sampling procedure of taking 10×1000 word forms from the test set, employed for the overall F1-score, for statistical significance testing, by Virpioja et al. [2011b]. Rather, we calculate the error analysis on the union of these sampled sets. As the sampling procedure may introduce the same word form in several samples, the error analysis precisions and recalls are not necessarily identical to the ones reported for the overall results.

In summary, while we cannot apply category F1-scores, we can instead categorize each error to either over-segmentation or under-segmentation. These categories then map directly to either reduced precision or recall.

4.2.5 Results

Here we summarize the results obtained using the experiment setup described above.

Boundary Precisions, Recalls, and F1-scores Segmentation results for English, Estonian, Finnish, and Turkish are presented in Table 4.4. The semi-supervised CRF approach yielded highest boundary F1-scores for all considered languages. The improvements over other models are statistically significant according to the applied 2-sided Wilcoxon signed-rank test. Segmentation results for Arabic and Hebrew are presented in Table 4.5. The supervised CRF approach yielded highest boundary F1-scores for both languages.

Method	Train (ann.)	Train (unann.)	Pre.	Rec.	F1
English					
MORFESSOR BASELINE (SSV)	1,000	384,903	84.4	83.9	84.1
MORFESSOR FLATCAT (SSV)	1,000	384,903	86.9	85.2	86.0
AG (SSV)	1,000	384,903	69.8	87.1	77.5
AG SELECT (SSV)	1,000	384,903	76.7	82.3	79.4
CRF (SV)	1,000	0	91.6	81.2	86.1
CRF (SSV)	1,000	384,903	89.3	87.0	**88.1**
Estonian					
MORFESSOR BASELINE (SSV)	1,000	3,908,820	80.6	80.7	80.7
MORFESSOR FLATCAT (SSV)	1,000	3,908,820	84.7	82.0	83.3
AG (SSV)	1,000	3,908,820	67.1	88.8	76.4
AG SELECT (SSV)	1,000	3,908,820	62.8	90.3	74.1
CRF (SV)	1,000	0	88.4	76.7	82.1
CRF (SSV)	1,000	3,908,820	90.2	86.3	**88.2**
Finnish					
MORFESSOR BASELINE (SSV)	1,000	2,206,719	76.0	78.0	77.0
MORFESSOR FLATCAT (SSV)	1,000	2,206,719	81.6	80.2	80.9
AG (SSV)	1,000	2,206,719	69.7	77.6	73.4
AG SELECT (SSV)	1,000	2,206,719	69.4	74.3	71.8
CRF (SV)	1,000	0	88.3	79.7	83.8
CRF (SSV)	1,000	2,206,719	89.3	87.9	**88.6**
Turkish					
MORFESSOR BASELINE (SSV)	1,000	617,298	85.1	89.4	87.2
MORFESSOR FLATCAT (SSV)	1,000	617,298	84.9	92.2	88.4
AG (SSV)	1,000	617,298	77.0	90.9	83.4
AG SELECT (SSV)	1,000	617,298	70.5	80.4	75.1
CRF (SV)	1,000	0	90.0	87.3	88.6
CRF (SSV)	1,000	617,298	89.3	92.0	**90.7**

Table 4.4. Precision, recall, and F1-scores for English, Estonian, Finnish, and Turkish.

Error Analysis In what follows, we examine how different error types contribute to the obtained precision and recall measures, and consequently, the overall F1-scores. To this end, we discuss the error analyses for English and Finnish presented in Tables 4.6 and 4.7, respectively. In order to provide a concise presentation, the tables consider the four most common boundary types individually and merge the rest into an OTHER category.

Compared to the Morfessor and Adaptor Grammar method families, the su-

Method	Train (ann.)	Train (unann.)	Pre.	Rec.	F1
Arabic					
MORFESSOR BASELINE (SSV)	782	3,130	78.7	79.7	79.2
POON ET AL. (SSV)	782	3,130	84.9	85.5	85.2
CRF (SV)	782	0	95.5	93.1	**94.3**
Hebrew					
MORFESSOR BASELINE (SSV)	692	2,770	71.5	85.3	77.8
POON ET AL. (SSV)	692	2,770	78.7	73.3	75.9
CRF (SV)	692	0	90.5	90.6	**90.6**

Table 4.5. Precision, recall, and F1-scores for Arabic and Hebrew.

pervised CRF approach has two distinct advantages: first, it produces distinctly less over-segmentation errors, and second, it is highly successful at capturing the SUFFIX-SUFFIX boundaries. On the other hand, its most evident weakness is its incapability of correctly detecting the STEM-STEM, that is compound word, boundaries. However, the Morfessor and Adaptor Grammar models appear to uncover the STEM-STEM boundary positions relatively well. In consequence, the semi-supervised CRF extension, which employs the predictions of the unsupervised Morfessor and Adaptor Grammar variants as features, uncovers the STEM-STEM boundaries with a substantially better accuracy compared to supervised variant while successfully preserving the strengths of the supervised approach. As a minor shortcoming, we note that improving recall (reducing under-segmentation) means that the semi-supervised extension is required to segment more compared to the supervised variant. For English, this increased segmentation results in a slight increase in over-segmentation of STEM, that is, the model trades off the increase in recall for precision.

4.2.6 Discussion

The CRF-based segmentation approach yielded the highest segmentation accuracies for all considered languages. This indicates that, in general, the CRF approach is able to utilize the available data more efficiently compared to the reference methods. In the remainder of this section, we discuss potential explanations for the empirical success of the discriminatively trained CRF approach in more detail.

We begin by noting that discriminative training has the advantage of directly optimizing segmentation accuracy with few assumptions about the data gener-

	Over-Segmentation					Under-Segmentation					
Method	STEM	SUFFIX	PREFIX	UNKNOWN	**PRE / TOTAL**	STEM-SUFFIX	SUFFIX-SUFFIX	STEM-STEM	PREFIX-STEM	OTHER	**REC / TOTAL**
WORDS	0.0	0.0	0.0	0.0	100.0	55.1	8.6	5.9	4.4	3.1	23.1
LETTERS	71.1	11.8	1.7	0.3	15.1	0.0	0.0	0.0	0.0	0.0	100.0
MORF.BL (SSV)	14.3	1.3	0.1	0.0	84.4	9.8	0.6	2.8	2.1	0.4	84.3
MORF.FC (SSV)	11.2	1.7	0.0	0.1	87.1	8.6	0.5	2.2	2.5	0.5	85.5
AG (SSV)	27.8	2.1	0.1	0.1	70.0	10.1	1.4	0.2	0.6	0.4	87.3
AG SEL. (SSV)	18.4	4.8	0.0	0.1	76.6	8.2	1.4	2.2	4.1	1.9	82.2
CRF (SV)	7.3	0.9	0.1	0.0	91.8	10.4	0.5	4.2	2.9	0.5	81.5
CRF (SSV)	9.6	0.8	0.0	0.1	89.5	8.4	0.5	1.4	1.9	0.4	87.4

Table 4.6. Error analysis for English. Over-segmentation and under-segmentation errors reduce precision and recall, respectively. For example, the total precision of MORF. BL (SSV) is obtained as $100.0 - 14.3 - 1.3 - 0.1 - 0.0 = 84.4$. The lines MORF. BL (SSV) and MORF. FC (SSV) correspond to the semi-supervised Morfessor Baseline and semi-supervised Morfessor FlatCat models, respectively.

ating process. Meanwhile, generative models can be expected to perform well only if the model definition matches the data generating process adequately. In general, discriminative approaches should generalize well under the condition that sufficient amount of training data is available. Given the empirical results, this condition appears to be fulfilled for morphological segmentation in the considered learning setting.

As the second point, we note that the Morfessor and Adaptor Grammar methods, the model of Poon et al. [2009], as well as the majority of earlier work discussed in Section 4.1.1, rely on **lexicon learning**, in which the model aims to learn segmentation by identifying lexical units. In the case of Morfessor [Creutz et al., 2007] and [Poon et al., 2009], the lexical units correspond to morphs while in Adaptor Grammars [Sirts and Goldwater, 2013] the units are parse-trees. Meanwhile, the CRFs implement a **boundary detection** approach, in which the aim is to identify morph boundary positions utilizing sub-

	Over-Segmentation					Under-Segmentation					
Method	STEM	SUFFIX	PREFIX	UNKNOWN	**PRE / TOTAL**	STEM-SUFFIX	SUFFIX-SUFFIX	STEM-STEM	SUFFIX-STEM	OTHER	**REC / TOTAL**
WORDS	0.0	0.0	0.0	0.0	100.0	49.2	21.8	17.2	4.8	2.1	4.1
LETTERS	65.2	13.8	0.7	0.6	19.7	0.0	0.0	0.0	0.0	0.0	100.0
MORF.BL (SSV)	20.8	2.9	0.0	0.2	76.1	13.6	5.9	1.9	0.5	0.2	78.0
MORF.FC (SSV)	15.3	2.9	0.0	0.1	81.7	12.2	5.2	1.5	0.6	0.3	80.2
AG (SSV)	27.9	2.1	0.1	0.2	69.7	14.7	6.5	0.7	0.2	0.3	77.6
AG SEL.(SSV)	24.2	6.1	0.0	0.1	69.5	13.2	7.8	2.4	1.1	1.1	74.4
CRF (SV)	9.3	2.3	0.0	0.0	88.3	10.7	2.2	5.8	1.1	0.6	79.7
CRF (SSV)	9.2	1.4	0.0	0.1	89.3	8.0	2.3	1.2	0.4	0.4	87.8

Table 4.7. Error analysis for Finnish. Over-segmentation and under-segmentation errors reduce precision and recall, respectively. The lines MORF. BL (SSV) and MORF. FC (SSV) correspond to the the semi-supervised Morfessor Baseline and semi-supervised Morfessor FlatCat models, respectively.

string contexts.[2] As the substrings are more frequent than lexical units, their use enables more efficient utilization of sparse data. For example, consider a training data that consists of a single labeled word form *kato+lla* (*on roof*). When segmenting an unseen word form *matolle* (*onto rug*), with the correct segmentation *mato+lle*, the CRFs can utilize the familiar left and right substrings *ato* and *ll*, respectively. In contrast, a lexicon-based model has a lexicon of two morphs {*kato, lla*}, neither of which match any substring of *matolle*.

Finally, we discuss how the varying approaches differ when learning to split affixes and compounds based on the performed error analysis. To this end, we first point out that, in the examined English and Finnish corpora, suffixes tend to serve syntactic purposes, such as marking case, tense, person or number. For example, consider the English suffix *-s* marking tense and person in *he plays* and number in *houses*. Thus, using the terminology introduced in Sec-

[2]It should be noted, however, that the log-linear model of Poon et al. [2009] also incorporates substring features at morph boundaries.

tion 2.1, the suffix class is considered closed and has only a small number of morphemes compared to the open prefix and stem categories. In consequence, a large coverage of suffixes should be achievable already with a relatively small annotated data set. This observation is supported by the evident success of the fully supervised CRF method in learning suffix splitting for both English and Finnish. On the other hand, while superiorly efficient at learning suffix splitting, the supervised CRF approach is apparently poor at detecting compound boundaries. Intuitively, learning compound splitting in a supervised manner seems infeasible since majority of stem forms are simply not present in the available small annotated data set. Meanwhile, the semi-supervised CRF extension and the generative Morfessor and Adaptor Grammar families, which do utilize the large unannotated word lists, capture the compound boundaries with an appealing high accuracy. This result again supports the intuition that in order to learn the open categories, one is required to utilize large amounts of word forms for learning. It then appears that the necessary information can be extracted from unannotated word forms.

4.3 Summary

In this chapter, we discussed the current author's contributions to statistical morphological segmentation. In particular, we focused on employing the CRF model in the semi-supervised learning setting, in which the available data consists of a small number of annotated segmentation examples and a large amount of unannotated raw word forms. The work on CRF-based morphological segmentation originally stemmed from the observation that the previously proposed systems for the considered learning setting were based on generative semi-supervised modeling frameworks. In contrast, the key idea of the CRFs is to directly estimate a conditional distribution of segmentation given a word form in a discriminative manner from the annotated data. Subsequently, it was shown that the CRF-based approach can be extended from supervised to semi-supervised learning in a straightforward manner using a simple feature expansion approach utilizing predictions of unsupervised segmentation algorithms.

The empirical evaluation on six languages showed that the semi-supervised CRF-based approach is highly successful in the morphological segmentation task. In general, discriminative learning approaches should generalize well under the condition that sufficient amount of annotated data is available during learning. Given the empirical results, this condition appears to be fulfilled

for morphological segmentation in the considered learning setting. In particular, the performed error analysis showed that closed class phenomena, such as suffixation of English and Finnish, can be learned already from a small number of annotated examples in a supervised manner. Meanwhile, open morpheme class phenomena, such as compounding of Finnish, can be learned by additionally exploiting the large unannotated word list using the semi-supervised approach.

5. Contributions to Morphological Tagging

In this chapter, we discuss the current author's contributions to statistical morphological tagging. The focus of discussion is on learning taggers based on the conditional random field (CRF) model for Finnish. To this end, the chapter describes methodological issues related to improving tagging accuracy and accelerating CRF model estimation in presence of large label sets with rich inner structures. These topics are covered in Sections 5.1.1 and 5.1.2, respectively. Section 5.2 then describes FinnPos, the first statistical morphological tagging and lemmatization toolkit designed specifically for Finnish. Empirical results obtained using the FinnPos system are discussed in Section 5.3. The presentation is based on Publications IV-VI.

5.1 Methodology

This section first discusses a feature expansion scheme for improving tagging accuracy of a CRF-based (or MEMM-based) tagger in presence of large label sets with rich inner structures. Subsequently, we discuss heuristics for accelerating CRF model learning in presence of large number of label sets.

5.1.1 Exploiting Sub-Label Dependencies for Improved Accuracy

The appeal of the MEMM and CRF models discussed in Sections 3.3 and 3.4, respectively, lies in their capability of utilizing rich, overlapping feature sets. For example, the classic work of Ratnaparkhi [1996] on morphological tagging utilizes a feature extraction scheme, in which each individual label position is associated with the following features describing the corresponding sentence position:

1. Bias (always active irrespective of input).

2. Word forms $x_{i-2}, \ldots, x_{i+2}$.

3. Prefixes and suffixes of the word form x_i up to length δ_{affix}.

4. If the word form x_i contains (one or more) capital letter, hyphen, dash, or digit.

In addition, the transition features capture dependencies between adjacent labels irrespective of the input x.

The [Ratnaparkhi, 1996] feature set described above can carry out the tagging with high accuracy given a conveniently simple label set, such as when tagging the Penn Treebank [Marcus et al., 1993] with a morphological tag set of 45 labels. However, it does to a certain extent overlook some beneficial dependency information in case the labels have a rich inner structure. In this section, we consider an expanded feature set which aims to exploit this inner structure of the Turku Treebank and FinnTreeBank labels described in Section 2.3.2.

We begin by considering the word form *kissat* (*cats*) where the suffix *-t* denotes plural number. Now, instead of associating the suffix *-t* solely with an exemplar compound label (Nominative, Plural), we also want to relate it with the sub-label (Plural). This is because one can exploit the suffix *-t* to predict the plural number also in words such as *vihreät* (*plural of green*) with an analysis (Adjective, Plural). Formally, we first define a function $\mathcal{P}(y_i)$ which partitions any label y_i into its sub-label components and returns them in an unordered set. For example, we could define $\mathcal{P}(\text{Noun, Plural}) = \{\text{Noun, Plural}\}$. We denote the set of all sub-label components as $\mathcal{S}$. Then, given $\mathcal{P}(y_i)$, instead of defining only features (3.15), we additionally associate the input x with all sub-labels s by defining features of the form

$$\phi(y_{i-n}, \ldots, y_i, x, i) = \chi_j(x, i)\mathbb{1}(s \in \mathcal{P}(y_i)) \quad \text{for} \quad \forall j \in 1 \ldots |\mathcal{X}|, \forall s \in \mathcal{S}, \qquad (5.1)$$

where $\mathbb{1}(s \in \mathcal{P}(y_i))$ returns one in case $\mathcal{P}(y_i)$ contains s and zero otherwise.

Furthermore, we can exploit transitional behavior of the sub-labels. For example, consider the sentence fragment *kissat juovat* (*cats drink*) where the words *kissat* and *juovat* have compound analyses (Nominative, Plural) and (Verb, 3rd person, Plural, Present tense, Active), respectively. Then, instead of merely modeling the transitional dependency between the compound labels, we can also model the **congruence**, that is, both analyses need to contain the sub-label denoting plural number. Formally, these transitions between sub-

labels are captured by features of the form

$$\phi(y_{i-n}, \ldots, y_i, x, i) = \mathbb{1}(s_{i-k} \in \mathcal{P}(y_{i-k})) \ldots \mathbb{1}(s_i \in \mathcal{P}(y_i)) \quad \text{for} \\ \forall s_{i-k}, \ldots, s_i \in \mathcal{S}\,, \forall k \in 1 \ldots m\,. \tag{5.2}$$

Note that we define the sub-label transitions up to order m, $1 \leq m \leq n$, that is, an nth-order CRF model is not obliged to utilize sub-label transitions all the way up to order n. This is because employing high-order sub-label transitions may potentially cause overfitting to training data due to substantially increased number of features (equivalent to the number of model parameters, $|\boldsymbol{w}| = |\boldsymbol{\phi}|$). For example, in Publication V, it was found for several languages that, in a second-order ($n = 2$), model it was beneficial to employ the sub-label emission feature set (5.1) and first-order sub-label transitions while discarding second-order sub-label transitions.

5.1.2 Approximative Heuristics for Learning and Decoding

In this section, we discuss two heuristics for accelerated estimation of CRF-based morphological tagging models in presence of high number of labels. The first approach is a straightforward cascade system of an orthography-based label guesser and a single, high-order CRF model inspired by the cascade model of Müller et al. [2013]. The second approach, coined simply the pseudo-perceptron algorithm in Publication IV, is obtained as combination of the structured perceptron algorithm [Collins, 2002] and the pseudo-likelihood estimator [Besag, 1975].

A Simple Cascade As discussed in Section 3.4.3, Müller et al. [2013] presented an approximative high-order CRF estimation technique utilizing a cascade of CRF models of increasing orders. In a general cascade system for structured prediction, one learns a series of increasingly complex models by restricting the search space of each model using the predictions of the less complex models [Weiss and Taskar, 2010]. Müller et al. [2013] implement this approach for CRFs using a coarse-to-fine decoding technique [Charniak and Johnson, 2005, Rush and Petrov, 2012] and show large savings in the computational cost of maximum likelihood training.

Here, we consider another cascading variant utilizing characteristics of the morphological tagging problem. In particular, our cascade is based on a series of two models, an orthography-based label guesser and a conventional nth-order CRF model. In this approach, the idea is simply to utilize the minimalistic, orthography-based label guesser to narrow down the label search space.

Input: training data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_\mathcal{D}}$
Output: model parameters $\boldsymbol{w}$
Let: $\hat{\mathcal{Y}}(j,T) = \{y_1\} \times \cdots \times \{y_{j-1}\} \times \mathcal{Y} \times \{y_{j+1}\} \times \cdots \times \{y_T\}$ (y_j is free)

1: **repeat until convergence**
2: **for** (x, y) **in** $\mathcal{D}$ **do**
3: **for** $j \in 1 \ldots T$ **do**
4: $z \leftarrow \arg\max_{u \in \hat{\mathcal{Y}}(j,T)} \boldsymbol{w} \cdot \boldsymbol{\Phi}(x, u)$
5: **if** $z \neq y$ **then**
6: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\Phi}(x, y) - \boldsymbol{\Phi}(x, z)$

Figure 5.1. The pseudo-perceptron algorithm.

In order to apply the cascade, we first learn a label guesser from the training data. The guesser ranks morphological tags according to their probability for any given word form. We then use the guesser to limit the candidate label set for each word x_i in sentence x and, subsequently, perform required inference among the limited candidate sequences. In Publication VI, the cascade was combined with the perceptron algorithm of Collins [2002], in which case the inference corresponds to performing the beam-search to obtain the (approximatively) highest scoring label sequences.

Perceptron Learning and Pseudo Search This section discusses the pseudo-perceptron algorithm, a straightforward heuristic estimation method for CRFs inspired by the perceptron algorithm [Collins, 2002] and the pseudo-likelihood estimator [Besag, 1975]. As in pseudo-likelihood, the key idea of the pseudo-perceptron approach is to perform the predictions over search space which differs from the correct sequence only at a single position. Meanwhile, the subsequent parameter updates are performed similarly to the perceptron algorithm using simple additive updates. The resulting algorithm is depicted in Figure 5.1 and can be combined with the parameter averaging approach similarly to the perceptron algorithm of Collins [2002]. Subsequent to learning, test instances are decoded using the standard Viterbi search in analogy to pseudo-likelihood.

Since the pseudo-perceptron variant performs the maximization over a single variable at a time (line 4 in Figure 5.1), its time complexity is linear in the number of labels in label set.

Next, we will discuss how the parameters yielded by the pseudo-perceptron algorithm are expected to generalize to test instances, that is, when decoding novel instances using Viterbi search. To this end, consider a set

of training instances (x, y) for which exact search yields correct solutions given some parameter vector $\boldsymbol{w}^*$. In other words, for each pair (x, y) it holds $\boldsymbol{\Phi}(x, y) - \boldsymbol{\Phi}(x, z_{exact}) = 0$, where $z_{exact} = \arg\max_{u \in \mathcal{Y}(x)} \boldsymbol{w}^* \cdot \boldsymbol{\Phi}(x, u)$. Then it necessarily holds that $\boldsymbol{\Phi}(x, y) - \boldsymbol{\Phi}(x, z_{pseudo}) = 0$, where $z_{pseudo} = \arg\max_{u \in \hat{\mathcal{Y}}(x,j)} \boldsymbol{w}^* \cdot \boldsymbol{\Phi}(x, u)$ for all $j \in 1 \ldots T$, because if this would not be the case, z_{exact} would not be the Viterbi path. Evidently, if $\boldsymbol{\Phi}(x, y) - \boldsymbol{\Phi}(x, z_{pseudo}) = 0$ holds for each pair (x, y) in the training set, the pseudo-perceptron has converged since parameter updates are no longer performed. What this means is that, given a linearly separable training data, it is *always possible* for the pseudo-perceptron to converge to parameters which linearly separate the data resulting in comparable generalizing performance to standard perceptron training. However, this is *not* in anyway guaranteed since, even in the case of linearly separable data, the algorithm may converge to parameters which do not linearly separate the set. For example, consider a training data set consisting of a single training instance:

x	a	a	a	a	x	a	a	a	a
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X	A	A	A	A	X	B	B	B	B

Then, consider a first-order CRF model (3.19) with emission features defined so that input characters $x_i \in \{a, b, c, x\}$ at each position i are associated with the corresponding labels $y_i \in \{A, B, C, X\}$. Using this feature extraction scheme, the training data is linearly separable. However, after five iterations, the pseudo-perceptron algorithm (initialized with a zero vector) converges to parameters which yield an incorrect Viterbi search result:

x	a	a	a	a	x	a	a	a	a
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X	A	A	A	A	X	A	A	A	A

In other words, the pseudo-perceptron algorithm has converged to a parameter setting which does not linearly separate the training set.

As discussed above, the pseudo-perceptron algorithm does not have convincing generalization guarantees in contrast to the pseudo-likelihood which is a consistent estimator. Nevertheless, in Publication IV, the approach was shown to work well in morphological tagging on several languages with large label sets. On the other hand, as supported by earlier empirical work [Sutton and McCallum, 2009], one could argue that the asymptotic consistency of pseudo-likelihood does not necessarily indicate high performance on real-life data sets either since these are always of finite sizes. However, in the current author's opinion, the real inconvenience of pseudo learning does not stem from the gen-

eralization problems but rather from the asymmetry between learning and decoding: while pseudo estimation of the model parameters may be feasible, applying the model to test instances using the exact Viterbi search may turn out to be computationally impractical. For example, in Publication IV, the decoding was performed using beam search instead of exact search. If this is indeed the case, it would appear most sensible to simply rely on the same approximative search procedure during learning and decoding. This is exactly what is accomplished if one performs learning and decoding utilizing the beam search or model cascades as discussed in Sections 3.4.3 and above.

5.2 FinnPos: A Morphological Tagging and Lemmatization Toolkit for Finnish

In this section, we discuss FinnPos, an open-source morphological tagging and lemmatization toolkit for Finnish.[1] The toolkit is readily applicable for tagging and lemmatization of running text with models learned from the recently published Finnish Turku Dependency Treebank and FinnTreeBank discussed in Section 2.3.2.

In the FinnPos system, we regard the morphological tagging and lemmatization tasks as two separate sub-problems. Given a sentence, each word form is assigned a morphological label by the morphological tagger based on a second-order CRF model. Subsequent to assigning the morphological label, selecting the appropriate word lemma is, in principle, straightforward given the set of full analyses (morphological labels and lemmas) provided by the OMorFi analyzer [Pirinen, 2008]. The output of the OMorFi analyzer was discussed in detail Section 2.3. However, OMorFi does not have full vocabulary coverage, that is, for some word forms no analyses are returned. In these cases, a simple baseline solution would be to simply return the original word form as the lemma. However, a more appealing approach is to learn a lemmatization model in a data-driven manner and apply it to lemmatize the unknown word forms [Chrupala et al., 2008]. In what follows, we discuss the applied CRF feature extraction scheme, the model learning and decoding, and the data-driven lemmatizer in Sections 3.4 and 5.2.3, respectively.

[1]The toolkit is freely available at `https://github.com/mpsilfve/FinnPos`.

5.2.1 Feature Extraction

We follow the classic work of Ratnaparkhi [1996] on morphological tagging and include the following input feature set:

1. Bias (always active irrespective of input).

2. Word forms $x_{i-2}, \ldots, x_{i+2}$.

3. Prefixes and suffixes of the word form x_i up to length δ_{affix}.

4. If the word form x_i contains (one or more) capital letter, hyphen, dash, or digit.

In addition, we use the following binary functions:

5. The lower-cased word form x_i.

6. The word pairs (x_{i-1}, x_i) and (x_i, x_{i+1}).

Given the output of the OMorFi analyzer, we include:

7. Each morphological label of word x_i returned by OMorFi.

Finally, in order to exploit the rich inner structure of the morphological labels, we include:

8. The sub-label feature extraction scheme discussed in Section 5.1.1.

5.2.2 Model Learning and Decoding

The CRF parameters are estimated from training data using a combination of averaged structured perceptron learning, beam search with minimum divergence beams [Pal et al., 2006], and the two-stage cascade described in Section 5.1.2. The label guesser component of the cascade is based on the lexical model for OOV words used by Brants [2000]. It assigns a probability $p(y|x)$ for any label $y \in \mathcal{Y}$ and an arbitrary word x based on the suffixes of x. The suffix-based approach is motivated by the fact that Finnish words are mostly inflected at

the end. Appealingly, the guesser can be trained and applied in mere seconds even when using large data sets. We use the label guesser to extract the minimal set of highest ranking label guesses y_i whose combined probability mass $\sum_{i=0}^{n} p(y_i|x)$ exceeds a threshold $\kappa \in [0,\ 1]$. The threshold is considered a hyperparameter of the learning procedure tuned on a held-out development set. Essentially, if one employs too small a κ, the model will underfit the training data, while increasing the threshold results in increasingly accurate approximations of the original learning problem.

Subsequent to parameter estimation, the resulting tagger can be applied to any given word sequence. In this decoding stage, the model assigns the highest scoring label sequence to a given word sequence using the same search procedure as during training, that is, using a combination of minimum divergence beams and the two-stage model cascade. The label guesser employs the same threshold value κ.

5.2.3 Lemmatizer

As stated above, subsequent to assigning the morphological label, selecting the appropriate word lemma is, in principle, straightforward given the set of full analyses (morphological labels and lemmas) provided by the OMorFi analyzer. However, OMorFi does not have full vocabulary coverage, that is, for some word forms no analyses are returned. Therefore, in order to lemmatize words unknown to the OMorFi analyzer, the FinnPos system follows Chrupala et al. [2008] and treats the lemmatization problem as a classification task, in which each class corresponds to a *suffix edit script*. For example, consider $[ies \rightarrow y]$, which removes a suffix "-ies" from the end of a word form, such as the English word form "beauties", and replaces it with another suffix "-y", thus producing the lemma "beauty". While Chrupala et al. [2008] use rather general edit scripts which can additionally modify prefixes and infixes of the word, we rely on the suffix-based approach because Finnish words mostly inflect at the end. The task of the lemmatizer is then to find the most appropriate edit script based on features extracted from the word form, its morphological label and its context. The script is chosen among minimal edit scripts, where the removed suffix is as short as possible [Chrupala et al., 2008].

5.3 Experiments

In this section, we present an empirical evaluation of the FinnPos system on two Finnish treebanks. The evaluation considers tagging and lemmatization accuracy and computational efficiency of learning and decoding. For comparison, we provide results using three reference toolkits. The presentation is based on Publication VI.

5.3.1 Data

The experiments are conducted on the Turku Dependecy Treebank (TDT) [Haverinen et al., 2009, 2014] and FinnTreeBank (FTB) [Voutilainen, 2011] described in Section 2.3.2. The treebanks do not have default partitions to training and test sets. Therefore, from each 10 consecutive sentences, we assign the 9th and 10th to the development sest and the test sets, respectively. The remaining sentences are assigned to the training sets. Statistics for the data splits are given in Table 5.1.

	TDT	FTB
train	10,858 sent. (145,775 tok.)	15,297 sent. (129,374 tok.)
dev	1,357 sent. (18,060 tok.)	1,912 sent. (16,579 tok.)
test	1,357 sent. (19,283 tok.)	1,912 sent. (16,075 tok.)
OOV in test	21.9%	22.1%

Table 5.1. Sizes of the training, development and test sets for FTB and TDT. The last row indicates the amount of tokens in the test set that are not found in the train set.

Tables 5.2 and 5.3 show the distributions of main POS classes for the test sets of TDT and FTB, respectively. Although the morphological labeling schemes in both FTB and TDT follow the labeling scheme of the OMorFi morphological analyzer, they are based on different versions of OMorFi. Therefore, the treebanks have differing main POS inventories. For example, the class Particle in FTB overlaps with the classes Conjunction and Adverb in TDT.

The encoding of nouns in FTB and TDT differs with regard to coordinated compounds. In Finnish, a coordination of two compound words which share an identical part can be written in an abbreviated manner. For example, *iso-tuloiset ja pienituloiset* (*people with high income and people with low income*) can be abbreviated as *iso- ja pienituloiset* because the coordinated compounds share the final part *-tuloiset*. FTB denotes the compound prefix *iso-* by a separate main POS Truncated. In contrast, TDT labels these prefixes as regular

nouns or adjectives.

Both FTB and TDT group common and proper nouns under the main POS Noun. The distinction is, however, denoted by an additional subcategory label.

		all words		OOV words	
		absolute	relative	absolute	relative
label	example	frequency	frequency (%)	frequency	frequency (%)
Noun	talo (a house)	6565	34.0	2656	62.9
Verb	istua (to sit)	3810	19.8	872	20.6
Punctuation	. ” ,	2897	15.0	0	0.0
Adverb	nopeasti (quickly)	1407	7.3	79	1.9
Adjective	hidas (slow)	1243	6.4	447	10.6
Pronoun	sinä (you)	1241	6.4	36	0.9
Conjunction	kun (when)	1096	5.7	2	0.0
Numeral	kolme (three)	652	3.4	73	1.7
Adposition	alla (under)	285	1.5	6	0.1
Foreign	live (live)	37	0.2	29	0.7
Symbol	:D	32	0.2	19	0.4
Interjection	nam (yum)	18	0.1	5	0.1

Table 5.2. The main POS distributions of all and out-of-vocabulary (OOV) words for the test set of Turku Dependency Treebank.

		all words		OOV words	
		absolute	relative	absolute	relative
label	example	frequency	frequency (%)	frequency	frequency (%)
Noun	talo (a house)	4354	27.1	2079	58.6
Verb	istua (to sit)	3831	23.8	755	21.3
Punctuation	. " ,	2302	14.3	0	0.0
Particle	näin (thus)	1502	9.3	23	0.6
Pronoun	sinä (you)	1437	8.9	37	1.0
Adverb	nopeasti (quickly)	1040	6.5	112	3.2
Adjective	hidas (slow)	1033	6.4	431	12.1
Numeral	kolme (three)	278	1.7	74	2.1
Adposition	alla (under)	273	1.7	16	0.5
Unknown	live (live)	16	0.1	14	0.4
Truncated	iso- (big)	9	0.1	8	0.2

Table 5.3. The main POS distributions of all and out-of-vocabulary (OOV) words for the test set of FinnTreeBank.

5.3.2 Reference Systems

This section summarizes the reference systems, namely, Morfette [Chrupala et al., 2008], MarMot [Müller et al., 2013], and HunPos [Halácsy et al., 2007].

Morfette Morfette is a toolkit for learning a morphological tagging and lemmatization model from annotated training data.[2] Given a corpus of sentences annotated with lemmas and morphological labels, and optionally a morphological analyzer, Morfette learns to assign analyses for new sentences. The Morfette tagging model is based on the CRF framework utilizing averaged perceptron learning. Meanwhile, lemmatization is handled as a classification task, in which each lemmatization class corresponds to a set of string edit operations required to transform the inflected word form into the corresponding lemma.

MarMot MarMot is a CRF-based morphological tagging toolkit.[3] Given a corpus of sentences annotated with morphological labels, and optionally a morphological analyzer, MarMot learns to assign morphological tags for new sentences. The model estimation of MarMot is based on the maximum likelihood criterion utilizing a pruning approach which enables efficient learning of high-order models. In contrast to FinnPos and Morfette systems, MarMot is solely a morphological tagging toolkit and does not perform lemmatization.

HunPos HunPos is an improved, open-source implementation of the morphological TnT tagger of Brants [2000].[4] Given a corpus of sentences annotated with morphological labels, and optionally a morphological analyzer, HunPos learns to assign morphological tags for new sentences. Similarly to MarMot, HunPos is solely a morphological tagging toolkit and does not perform lemmatization. The HunPos tagger is based on the generative HMM framework which makes it sensitive to rich feature sets compared to the discriminatively trained CRFs. On the other hand, due to the generative estimation procedure and simple feature sets, the system is extremely fast to both train and apply. While the HunPos system was originally designed for morphological tagging of Hungarian, it is a natural choice for a Finnish morphological tagger due to the relatedness of Hungarian and Finnish languages: Hungarian and Finnish are both agglutinative and morphologically rich languages belonging to the Finno-Ugric family.

[2] Available at `https://sites.google.com/site/morfetteweb/`.
[3] Available at `https://code.google.com/p/cistern/wiki/marmot`.
[4] Available at `http://code.google.com/p/hunpos/`.

5.3.3 Evaluation

Test performances in tagging and lemmatization (when applicable) are evaluated using *per-token* accuracies. These accuracies are reported separately for all words and words not seen in the training data. We establish statistical significance (with confidence level 0.95) using the standard 2-sided Wilcoxon signed-rank test performed on 10 randomly divided, non-overlapping subsets of the complete test sets.

5.3.4 Hardware

The experiments are run on a desktop computer (Intel Core i5-4300U with 1.90 GHz and 16 GB of memory).

5.3.5 Results

Obtained tagging and lemmatization accuracies, training times, and decoding speeds for TDT and FTB are presented in Tables 5.4 and 5.5, respectively. In what follows, we will compare the FinnPos system individually with the reference systems.

FinnPos versus Morfette. We begin by comparing FinnPos and Morfette, both of which perform morphological tagging and lemmatization. The FinnPos system outperforms the Morfette with respect to both tagging and lemmatization accuracy. The differences in accuracies are statistically significant. Furthermore, compared to FinnPos, the training time of Morfette is substantially higher and decoding speed substantially lower.

FinnPos versus MarMot. The FinnPos system outperforms MarMot with respect to tagging accuracy. However, the differences in accuracies are not statistically significant. Compared to FinnPos, the training time of MarMot is substantially higher and decoding speed substantially lower. Finally, MarMot does not perform lemmatization.

FinnPos versus HunPos. The training time of the HunPos system is substantially lower compared to FinnPos or any other system. While faster to estimate and apply, however, the tagging accuracy of HunPos is significantly lower compared to FinnPos on both data sets. The HunPos system does not perform lemmatization.

	tag acc.		lemma acc.		train. time		dec. speed (tok/s)
toolkit	all	OOV	all	OOV	tagger	lemmatizer	
HunPos	91.64	76.07	-	-	2 s	-	101,000
MarMot	96.29	91.04	-	-	38 min	-	1,000
Morfette	93.91	82.19	89.33	72.04	203 min	16 min	40
FinnPos	**96.31**	**91.64**	**93.29***	**84.28**	4 min	5 min	16,000

Table 5.4. Results for Turku Dependency Treebank.

	tag acc.		lemma acc.		train. time		dec. speed (tok/s)
toolkit	all	OOV	all	OOV	tagger	lemmatizer	
HunPos	93.65	82.55	-	-	2 s	-	141,000
MarMot	96.21	91.46	-	-	24 min	-	1,000
Morfette	95.03	86.81	95.66	83.12	128 min	8 min	60
FinnPos	**96.23**	**92.34**	**96.37**	**89.10**	3 min	3 min	18,000

Table 5.5. Results for FinnTreeBank.

5.3.6 Error Analysis

In this section, we present and discuss the distribution of the errors yielded by the FinnPos system. In particular, we examine how the errors are distributed across the main POS classes. In addition, we examine individual error types, that is, which categories are most often confused for one another.

First, consider Tables 5.6 and 5.7 which contain the errors distributions for TDT and FTB, respectively. For both data sets, the majority of errors take place in the noun and verb categories. This is expected as these categories are most frequent in the test sets and, as shown in Tables 5.2 and 5.3, and contain the most OOV word forms.

Second, consider Tables 5.8 and 5.9 which contain confusion matrices of errors for TDT and FTB, respectively. Due to space constraints, the matrices include 25 most prominent confusion pairs. For both data sets, the majority of errors take place when a noun is confused with a noun or a verb is confused with a verb, that is, the tagger yields the correct main POS class but an incorrect detailed morphological label. For example, consider the noun phrase *kiveä ja terästä oleva monumentti* (*a monument made of stone and steel*).[5] The word form *terästä* could be the partitive form of the noun *teräs* (steel) or the elative form of the noun *terä* (a blade). From a syntactical point of view, both interpretations are possible. From a semantical point of view, however, only

[5]The example is taken from FinnTreeBank.

	all words		OOV words	
main POS	absolute frequency	relative frequency (%)	absolute frequency	relative frequency (%)
Noun	271	38.1	182	53.5
Verb	205	28.8	61	17.9
Adjective	65	9.1	33	9.7
Pronoun	48	6.8	17	5.0
Adverb	45	6.3	11	3.2
Foreign	23	3.2	21	6.2
Numeral	15	2.1	4	1.2
Adposition	15	2.1	2	0.6
Conjunction	13	1.8	1	0.3
Symbol	7	1.0	7	2.1
Interjection	4	0.6	1	0.3

Table 5.6. Error distribution over main POS classes for Turku Dependency Treebank.

	all words		OOV words	
main POS	absolute frequency	relative frequency (%)	absolute frequency	relative frequency (%)
Noun	184	30.4	127	49.0
Verb	155	25.6	54	20.8
Adverb	71	11.7	11	4.2
Particle	47	7.8	4	1.5
Pronoun	46	7.6	3	1.2
Adjective	40	6.6	27	10.4
Numeral	24	4.0	11	4.2
Adposition	17	2.8	3	1.2
Unknown	12	2.0	11	4.2
Truncated	8	1.3	8	3.1
Punctuation	2	0.3	-	-

Table 5.7. Error distribution over main POS classes for FinnTreeBank.

the partitive interpretation is valid.

5.3.7 Discussion

Compared to the reference toolkits, the FinnPos system provides the highest accuracies with respect to tagging and lemmatization accuracy. In addition, the system is computationally more efficient to train and apply compared to the MarMot and Morfette systems which also utilize discriminative learning.

	Noun	Verb	Adjective	Pronoun	Adverb	OTHER
Noun	26.0	2.7	3.1	0.7	1.5	4.1
Verb	5.1	20.5	2.3	0.4	0.1	0.4
Adjective	2.3	2.4	3.0	0.0	1.4	0.1
Pronoun	1.5	0.1	0.0	2.4	1.3	1.4
Adverb	1.0	0.1	0.6	1.1	0.3	3.2
OTHER	4.2	1.0	0.1	0.1	2.1	3.2

Table 5.8. The confusion matrix of errors for Turku Dependency Treebank with relative error frequencies. For example, labeling a noun as a verb comprises 2.7 percentages of all errors, whereas labeling a verb as a noun comprises 5.1 percentages of all errors. For the five main POS classes with most labeling errors, results are shown separately. The class OTHER comprises all remaining main POS classes.

	Noun	Verb	Adverb	Particle	Pronoun	OTHER
Noun	17.7	1.7	3.0	1.5	0.0	6.6
Verb	2.5	17.2	0.3	1.2	0.5	4.0
Adverb	1.2	0.0	1.7	3.5	2.0	3.5
Particle	1.0	0.5	1.3	4.1	0.7	0.2
Pronoun	0.2	0.0	2.5	0.3	3.6	1.0
OTHER	5.6	2.8	3.0	0.3	0.7	4.6

Table 5.9. The confusion matrix of errors for FinnTreeBank with relative error frequencies.

As discussed in Section 2.3.2, the TDT and FTB corpora differ somewhat in the included text domains as well as the labeling schemes. However, these differences appear to have a minor effect on the tagging and lemmatization accuracy of the FinnPos system.

According to the error analysis in Section 5.3.6, while the main POS label is often correct, the detailed morphological information is more difficult to infer. The analysis shows that substantial improvement in tagging accuracy would require improved inference of the detailed morphological information for nouns and verbs specifically. This, however, is a difficult task because the immediate syntactical context often does not provide adequate clues for disambiguation. The choice between different detailed labels is often lexically and semantically conditioned which makes it particularly difficult for OOV words.

5.4 Summary

In this chapter, we discussed the current author's contributions to statistical morphological tagging with a particular focus on CRF-based morphological tag-

ging of Finnish. To this end, the section described methodological issues related to improved accuracy and accelerated learning and decoding in presence of large label sets with rich inner structures. The methodological insights were then utilized in the design of the FinnPos system, the first statistical morphological tagging and lemmatization toolkit implemented specifically for Finnish.

6. Conclusions

This thesis work examined natural language processing methodology related to morphology, that is, the study of internal structure of words. In particular, we focused on two widely applied morphological analysis tasks, namely, morphological tagging and segmentation. Following the modern artificial intelligence discipline, these problems were approached using data-driven (statistical) machine learning methodology with a specific emphasis on the influential conditional random field (CRF) modeling framework. In what follows, we summarize the main contributions of the presented thesis.

Morphological Segmentation As the first contribution, this thesis discussed data-driven morphological segmentation employing the CRF model. In particular, we focused on the semi-supervised learning setting, in which the available data consists of a small number of annotated segmentation examples and a large amount of unannotated raw word forms. The work on this CRF-based approach originally stemmed from the observation that the previously proposed systems for the considered learning setting were based on generative semi-supervised modeling frameworks. In contrast, the key idea of the CRFs is to directly estimate a conditional distribution of segmentation given a word form in a discriminative manner from the annotated data. Subsequently, it was shown that the CRF-based approach can be extended from supervised to semi-supervised learning in a straightforward manner using a simple feature expansion approach utilizing predictions of unsupervised segmentation algorithms.

The provided empirical evaluation on four languages showed that the semi-supervised CRF-based approach is highly successful in the considered morphological segmentation task compared to competing generative methods. In general, discriminative learning approaches should generalize well under the condition that sufficient amount of annotated data is available during learning. Given the empirical results, this condition appears to be fulfilled for morpho-

logical segmentation in the considered learning setting.

In literature, the morphological segmentation task is often motivated as an inexpensive means of obtaining a type of morphological analysis for agglutinative languages. Given the results presented in this thesis, it appears that the CRF-based approach is currently the most suitable tool for learning the segmentation given a small amount of manually annotated data.

Morphological Tagging As the second contribution, this thesis discussed statistical morphological tagging in presence of large label sets with fine-grained inner structures. These types of morphological labels occur frequently in descriptions of morphologically rich languages, such as Finnish and other languages of the Finno-Ugric family. In particular, we discussed how to improve the tagging accuracy of a CRF-based tagger by exploiting the rich inner structure of the labels. In addition, we discussed heuristics for accelerated learning of the CRF model. Lastly, these methodological findings were utilized in the design of FinnPos, the first open-source data-driven morphological tagging and lemmatization toolkit designed specifically for Finnish. The FinnPos system is readily applicable for tagging and lemmatization of running text with models learned from the recently published Finnish Turku Dependency Treebank and FinnTreeBank.

Bibliography

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2004.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, pages 179–195, 1975.

Ella Bingham and Heikki Mannila. 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 (ACM SIGKDD 2001)*, pages 245–250, San Fransisco, California, USA, 2001.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, Wisconsin, USA, 1998.

Thorsten Brants. TnT: A statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP 2000)*, pages 224–231, Seattle, Washington, USA, 2000.

Michael R Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1-3):71–105, 1999.

Peter Brown, Peter Desouza, Robert Mercer, Vincent Della Pietra, and Jenifer Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4): 467–479, 1992.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 173–180, Ann Arbor, Michigan, USA, 2005.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the 2010*

Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pages 575–584, Massachusetts, USA, 2010.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. A Bayesian mixture model for part-of-speech induction using multiple features. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 638–647, Edinburgh, UK, 2011.

Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. Learning morphology with Morfette. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2362–2367, Marrakech, Morocco, 2008.

Alexander Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 59–66, Budapest, Hungary, 2003.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, volume 10, pages 1–8, Philadelphia, Pennsylvania, USA, 2002.

Michael Collins and Brian Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL 2004)*, pages 111–118, Barcelona, Spain, 2004.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine learning (ICML 2008)*, pages 160–167, Helsinki, Finland, 2008.

Çağrı Çöltekin. Improving successor variety for morphological segmentation. *LOT Occasional Series*, 16:13–28, 2010.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL 2002*, pages 21–30, Philadelphia, Pennsylvania, USA, 2002.

Mathias Creutz and Krista Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005)*, pages 106–113, Espoo, Finland, 2005.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1):3:1–3:29, 2007.

Hal Daumé III. *Practical structured learning techniques for natural language processing*. PhD thesis, University of Southern California, 2006.

Adrià de Gispert, Sami Virpioja, Mikko Kurimo, and William Byrne. Minimum Bayes risk combination of translation hypotheses from alternative morphological decompositions. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 73–76, Boulder, Colorado, USA, 2009.

Thomas Emerson. The 2nd international Chinese word segmentation bakeoff. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, volume 133. Jeju Island, Korea, 2005.

Winthrop Nelson Francis. *A standard sample of present-day English for use with digital computers*. Brown University, 1964.

Yoav Freund and Robert Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

Sharon Goldwater. *Nonparametric Bayesian Models of Lexical Acquisition*. PhD thesis, Brown University, 2006.

Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

Spence Green and John DeNero. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 146–155, Jeju Island, Korea, 2012.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1177–1185, Dublin, Ireland, 2014.

Auli Hakulinen, Riitta Korhonen, Maria Vilkuna, and Vesa Koivisto. *Iso suomen kielioppi*. Suomalaisen kirjallisuuden seura, 2004. ISBN ISBN:978-952-5446-35-7. URL `http://scripta.kotus.fi/visk`.

Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 209–212, Prague, Czech Republic, 2007.

John M Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. 1971.

Zellig S Harris. From phoneme to morpheme. *Language*, 31(2):190–222, 1955.

Katri Haverinen, Filip Ginter, Veronika Laippala, Timo Viljanen, and Tapio Salakoski. Dependency annotation of Wikipedia: First steps towards a Finnish treebank. In *The 8th International Workshop on Treebanks and Linguistic Theories (TLT 2009)*, pages 95–105, Milan, Italy, 2009.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48(3):493–531, 2014.

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*, 20(4):515–541, 2006.

Timo Honkela, Ville Pulkki, and Teuvo Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing map. In *Proceedings of the 5th International Conference on Artificial Neural Networks (ICANN 1995)*, pages 3–7, Paris, France, 1995.

Liang Huang, Suphan Fayong, and Yang Guo. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2012)*, pages 142–151, Montreal, Canada, 2012.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106 (4):620, 1957.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (COLING ACL 2006)*, pages 209–216, Sidney, Australia, 2006.

Mark Johnson. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the 10th Meeting of ACL Special Interest Group on Computational Morphology and Phonology (SIGMORPHON 2008)*, pages 20–27, Columbus, Ohio, USA, 2008.

Mark Johnson and Katherine Demuth. Unsupervised phonemic Chinese word segmentation using Adaptor Grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 528–536, Beijing, China, 2010.

Mark Johnson and Sharon Goldwater. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 317–325, Boulder, Colorado, USA, 2009.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: a framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pages 641–648, Vancouver, Canada, 2006.

Heiki-Jaan Kaalep, Kadri Muischnek, Kristel Uiboaed, and Kaarel Veskis. The Estonian reference corpus: Its composition and morphology-aware user interface. In *Proceedings of the 2010 Conference on Human Language Technologies – The Baltic Perspective: Proceedings of the 4th International Conference Baltic (HLT 2010)*, pages 143–146, Riga, Latvia, 2010.

Özkan Kılıç and Cem Bozsahin. Semi-supervised morpheme segmentation without morphological analysis. In *Proceedings of the LREC 2012 Workshop on Language Resources and Technologies for Turkic Languages*, pages 52–56, Istanbul, Turkey, 2012.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology (SIGMORPHON 2010)*, pages 78–86, Uppsala, Sweden, 2010.

Mikko Kurimo, Sami Virpioja, Ville Turunen, Graeme W. Blackwood, and William Byrne. Overview and results of Morpho Challenge 2009. In *Working Notes for the CLEF 2009 Workshop*, pages 578–597, Corfu, Greece, 2009.

Mikko Kurimo, Sami Virpioja, and Ville Turunen. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo, Finland, 2010.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williamstown, Massachusetts, USA, 2001.

Michael Lamar, Yariv Maron, Mark Johnson, and Elie Bienenstock. SVD and clustering for unsupervised POS tagging. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 215–219, Uppsala, Sweden, 2010.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. Simple type-level unsupervised POS tagging. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 853–861, Massachusetts, USA, 2010.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. Modeling syntactic context improves morphological segmentation. In *Proceedings of the 15th Conference on Computational Natural Language Learning (CoNLL 2011)*, pages 1–9, Portland, Oregon, USA, 2011.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013)*, pages 29–37, Sofia, Bulgaria, 2013.

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on natural language learning (CoNLL 2002)*, pages 49–55, Taipei, Taiwan, 2002.

Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of the 46th Annual Meeting of Association for Computational Linguistics: Human Language Technologies (ACL HLT 2008)*, pages 870–878, Columbus, Ohio, USA, 2008.

Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

Mitchell Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2):313–330, 1993.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL HLT 2003)*, pages 188–191, Edmonton, Canada, 2003.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 591–598, Stanford, CA, USA, 2000.

Christian Monson, Kristy Hollingshead, and Brian Roark. Simulating morphological analyzers with stochastic taggers for confidence estimation. In *Multilingual Information Access Evaluation I - Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*. 2010.

Alexander Mozeika, Onur Dikmen, and Joonas Piili. Consistent inference of a general model using the pseudolikelihood method. *Physical Review E*, 90:010101, 2014.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient higher-order CRFs for morphological tagging. In *Proceedings of 2013 Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 322–332, Seattle, Washington, USA, 2013.

Karthik Narasimhan, Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, and Regina Barzilay. Morphological segmentation for keyword spotting. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 880–885, Doha, Qatar, 2014.

Robert Östling. Stagger: A modern POS tagger for Swedish. *Northern European Journal of Language Technology*, 3(1):1–18, 2012.

Chris Pal, Charles Sutton, and Andrew McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Internation Conference on Acoustics, Speech and Signal Processing (ICASP 2006)*, volume 5, pages 581–584, Toulouse, France, 2006.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 562–568, Geneva, Switzerland, 2004.

Tommi Pirinen. Automatic finite state morphological analysis of Finnish language using open source resources (in Finnish). Master's thesis, University of Helsinki, 2008.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, pages 209–217, Boulder, Colorado, USA, 2009.

Randolph Quirk, David Crystal, and Pearson Education. *A comprehensive grammar of the English language*. Cambridge Univ Press, 1985.

Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)*, volume 1, pages 133–142, New Brunswick, New Jersey, USA, 1996.

Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Series in Computer Science, 1989.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

Alexander M Rush and Slav Petrov. Vine pruning for efficient multi-pass dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2012)*, pages 498–507, Montreal, Canada, 2012.

Hinrich Schütze. Distributional part-of-speech tagging. In *Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics (EACL 1995)*, pages 141–148, Dublin, Ireland, 1995.

Miikka Silfverberg and Krister Linden. Combining statistical models for POS tagging using finite-state calculus. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 183–190, Riga, Latvia, 2011.

Kairit Sirts and Sharon Goldwater. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1(May):255–266, 2013.

Benjamin Snyder and Regina Barzilay. Cross-lingual propagation for morphological analysis. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008)*, pages 848–854, Chicago, Illinois, 2008.

Sebastian Spiegler and Peter A Flach. Enhanced word decomposition by calibrating the decision threshold of probabilistic models and using a model ensemble. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 375–383, Uppsala, Sweden, 2010.

Richard Sproat and Thomas Emerson. The first international Chinese word segmentation bakeoff. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing-Volume 17*, pages 133–143, Sapporo, Japan, 2003.

Weiwei Sun and Jia Xu. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 970–979, Edinburgh, UK, 2011.

Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine Learning*, 77(2):165–194, 2009.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

Erik Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning (CoNLL 2000)*, pages 127–132, Lisbon, Portugal, 2000.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (NAACL HLT 2003)*, pages 173–180, Edmonton, Canada, 2003.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 384–394, Uppsala, Sweden, 2010.

Ville Turunen and Mikko Kurimo. Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval. *ACM Transactions on Speech and Language Processing*, 8(1):1:1–1:25, 2011.

Douglas L Vail, John D Lafferty, and Manuela M Veloso. Feature selection in conditional random fields for activity recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2007 (IROS 2007*, pages 3379–3384, San Diego, California, USA, 2007.

Sami Virpioja, Oskar Kohonen, and Krista Lagus. Evaluating the effect of word frequencies in a probabilistic generative model of morphology. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 230–237, Riga, Latvia, 2011a.

Sami Virpioja, Ville Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90, 2011b.

S.V.N. Vishwanathan, Nicol Schraudolph, Mark Schmidt, and Kevin Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine learning (ICML 2006)*, pages 969–976, Pittsburgh, Pennsylvania, USA, 2006.

Atro Voutilainen. FinnTreeBank: Creating a research resource and service for language researchers with Constraint Grammar. In *Proceedings of the NODALIDA 2011 Workshop Constraint Grammar Applications*, pages 41–49, Riga, Latvia, 2011.

Yang Wang, Gholamreza Haffari, Shaojun Wang, and Greg Mori. A rate distortion approach for semi-supervised conditional random fields. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 2008–2016, Vancouver, Canada, 2009.

Yiou Wang, Yoshimasa Tsuruoka Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai, Thailand, 2011.

David Weiss and Ben Taskar. Structured prediction cascades. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, pages 916–923, Sardinia, Italy, 2010.

David Weiss, Benjamin Sapp, and Ben Taskar. Structured prediction cascades. 2012. URL `http://arxiv.org/abs/1208.3279`.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL 1995)*, pages 189–196, Cambridge, Massachusetts, USA, 1995.

Yue Zhang and Stephen Clark. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151, 2011.

Hai Zhao, Chang-Ning Huang, and Mu Li. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*, pages 162–165, Sydney, Australia, 2006.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.

Aalto-DD 67/2016

ISBN 978-952-60-6753-7 (printed)
ISBN 978-952-60-6754-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Signal Processing and Acoustics
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS