Statistical Models for Unsupervised Learning of Morphology and POS Tagging

Burcu Can

PhD

The University of York
Computer Science

September 2011

To my parents

Abstract

This thesis concentrates on two fields in natural language processing. The main contribution of the thesis is in the field of morphology learning. Morphology is the study of how words are formed combining different language constituents (called *morphemes*) and morphology learning is the process of analysing words, by splitting into these constituents. In the scope of this thesis, morphology is learned mainly by paradigmatic approaches, in which words are analysed in groups, called *paradigms*. Paradigms are morphological structures having the capability of generating various word forms. We propose approaches for capturing paradigms to perform morphological segmentation. One of the approaches proposed captures paradigms within a hierarchical tree structure. Using a hierarchical structure covers a wide range of paradigms by spotting morphological similarities.

The second scope of the thesis is part-of-speech (POS) tagging. Parts-of-speech are linguistic categories, which group words having similar syntactic features, i.e. noun, adjective, verb etc. In the thesis, we investigate how to exploit POS tags to learn morphology. We propose a model to capture paradigms through syntactic categories. When syntactic categories are provided, the proposed system can capture paradigms well. Following this approach, we extend it for the case of having no syntactic categories provided. To this end, we propose a joint model, in which POS tags and morphology are learned simultaneously.

Our results show that a joint model is possible for learning morphology and POS tagging.

We also study morpheme labelling, for which we propose a clustering algorithm that groups morphemes showing similar features. The algorithm can capture morphemes having similar meanings.

Contents

List of Tables 14
List of Figures 17
Acknowledgements 19
Declaration 23

1 Introduction and Motivation **25**
1.1 Introduction 25
1.2 Morphological Segmentation 26
1.2.1 Application Areas of Morphological Segmentation . . . 26
1.3 POS Tagging 28
1.3.1 Application Areas of POS Tagging 29
1.4 Learning Morphology and POS 30
1.5 The Interaction Between Morphology and POS 31
1.6 Research Objectives & Questions 32
1.7 Thesis Structure 34

2 Background **36**
2.1 Introduction 36
2.2 Linguistic Background 37
2.2.1 Morphology 37
2.2.1.1 Morphemes, Affixes, Roots, Stems etc. 39
2.2.1.2 Allomorphs 40
2.2.1.3 Inflectional and Derivational Morphology . . 41
2.2.2 Syntax 42
2.2.2.1 Syntactic Categories 42

2.2.2.2 Open-Class vs Closed-Class Syntactic Categories . . . 43
2.2.3 Interaction Between Linguistic Levels . . . 43
2.2.3.1 The Morphology-Phonology Interaction . . . 43
2.2.3.2 The Morphology-Syntax Interaction . . . 44
2.2.3.3 The Morphology-Semantics Interaction . . . 44
2.3 Machine Learning Background . . . 45
2.3.1 Maximum Likelihood Estimate . . . 46
2.3.2 Maximum A Posteriori Estimate . . . 48
2.3.3 Bayesian Modelling . . . 49
2.3.4 Minimum Description Length Principle . . . 50
2.3.5 Integration over Parameters . . . 51
2.3.6 Bayesian Non-Parametric Modelling . . . 55
2.3.7 Mixture Models . . . 58
2.3.8 Inference . . . 61
2.3.8.1 Markov Chain Monte Carlo (MCMC) . . . 61
2.4 Conclusion . . . 64

3 A Literature Review of Unsupervised Morphology Learning and POS Tagging 65
3.1 Introduction . . . 65
3.2 Unsupervised Morphology Learning . . . 66
3.2.1 Deterministic Models . . . 66
3.2.1.1 Letter Successor Variety (LSV) Models . . . 66
3.2.1.2 Information Theoretic Models . . . 68
3.2.1.3 Other Approaches . . . 73
3.2.2 Stochastic Models . . . 74
3.2.2.1 Probabilistic Frameworks . . . 75
3.2.2.2 Generative Probabilistic Models . . . 77
3.2.2.3 Log-Linear Models . . . 78
3.2.3 Evaluation of Morphology Segmentation Algorithms . . . 79
3.2.3.1 Evaluation Using a Gold Standard Segmentation 80
3.2.3.2 Evaluation through Other Tasks . . . 81
3.3 Unsupervised POS Tagging . . . 82
3.3.1 Clustering . . . 82
3.3.2 Hidden Markov Models . . . 85
3.3.3 Other Approaches . . . 89
3.3.4 Evaluation of the POS Tagging Algorithms . . . 93
3.3.5 A Literature Review to Cooperative Learning of Morphology and Syntax . . . 95
3.3.5.1 Morphology Learning by Using Syntax . . . 95

3.3.5.2 Learning Syntactic Categories Using Morphology . . . 96
3.3.5.3 Cooperation with Other Types of Linguistic Features . . . 97
3.4 Conclusion . . . 99

4 Morphological Segmentation Using Syntactic Categories 100
4.1 Introduction . . . 100
4.2 Motivation . . . 101
4.3 Learning Morphological Paradigms Using Syntactic Categories . . . 101
4.3.1 Learning Syntactic Categories . . . 102
4.3.2 Identifying Potential Morphemes . . . 104
4.3.3 Inducing Mophological Paradigms . . . 106
4.3.4 Merging Paradigms . . . 107
4.4 Morphological Segmentation . . . 110
4.4.1 Handling known words . . . 111
4.4.2 Handling unknown words . . . 112
4.4.3 Handling compounds . . . 112
4.5 Experiments & Evaluation . . . 112
4.6 Conclusion . . . 114

5 Probabilistic Hierarchical Clustering for Morphology Learning 116
5.1 Introduction . . . 116
5.2 Motivation . . . 117
5.3 Probabilistic Hierarchical Model . . . 118
5.3.1 Mathematical Definition . . . 118
5.4 Morphological Segmentation . . . 120
5.4.1 Model Definition . . . 120
5.4.2 Embedding Morphology into the Hierarchical Model . . . 125
5.4.3 Inference . . . 127
5.4.4 Morphology Segmentation . . . 130
5.5 Experiments . . . 132
5.5.1 Experiments with Single Split Points . . . 133
5.5.2 Experiments with Multiple Split Points . . . 137
5.5.3 Comparison with Other Systems . . . 140
5.5.4 Experiments in Various Languages . . . 141
5.6 Conclusion . . . 143

6 Joint Learning of Morphology and POS Tagging 145
6.1 Introduction . . . 145
6.2 Motivation . . . 146
6.3 Model Definition . . . 147
6.3.1 POS Tagging . . . 147
6.3.2 Morphology Learning . . . 151
6.4 Inference . . . 153
6.4.1 Inferring POS . . . 154
6.4.2 Inferring Morphology . . . 156
6.5 Algorithm . . . 157
6.6 Experiments & Evaluation . . . 158
6.6.1 POS Tagging Results . . . 159
6.6.2 Morphological Segmentation Results . . . 164
6.7 Discussion . . . 167
6.8 Conclusion . . . 168

7 Morpheme Labelling 170
7.1 Introduction . . . 170
7.2 Previous Work . . . 170
7.3 Intuition . . . 171
7.3.1 Allomorphs . . . 171
7.3.2 Homophonous morphemes . . . 172
7.4 Background . . . 173
7.4.1 Hierarchical Clustering . . . 173
7.4.1.1 Single-Linkage Clustering . . . 173
7.4.1.2 Complete-Linkage Clustering . . . 175
7.4.1.3 Average-Linkage Clustering . . . 175
7.5 The Algorithm for Clustering Morphemes . . . 176
7.6 Experiments & Results . . . 179
7.7 Discussion . . . 184
7.8 Conclusion . . . 185

8 Conclusion and Future Work 186
8.1 Introduction . . . 186
8.2 Thesis Summary . . . 186
8.3 Contributions . . . 187
8.4 Future Work . . . 188
8.5 Final Words . . . 190

Appendix 190

A Penn Treebank Tags **191**

References **193**

Index **212**

List of Tables

4.1 Some sample syntactic categories obtained from the English dataset. 104
4.2 Some high ranked potential morphemes in PoS clusters for English and German. 105
4.3 Some high ranked potential morphemes in PoS clusters for Turkish. 106
4.4 Sample paradigms in English 108
4.5 Sample paradigms in Turkish 109
4.6 Sample paradigms in German 109
4.7 Merged paradigms in English 110
4.8 Merged paradigms in Turkish 111
4.9 Merged paradigms in German 111
4.10 Obtained evaluation scores in Morpho Challenge 2009 Competition 1 with the winner participant's F-score. 114
4.11 Obtained average precisions (AP) for the Morpho Challenge 2009 Competition 2 114

5.1 Evaluation scores of single split point experiments obtained from the trees with 10K words. 134
5.2 Some tree nodes obtained from the trees with 10K words. 135
5.3 Evaluation scores of single split point experiments obtained from the trees with 16K words. 135
5.4 Some tree nodes obtained from the trees with 16K words. 136
5.5 Evaluation scores of single split point experiments obtained from the trees with 22K words. 137
5.6 Some tree nodes obtained from the trees with 22K words. 137

5.7 Evaluation scores of multiple split point experiments obtained from the trees with 10K words. 139
5.8 Evaluation scores of single split point experiments obtained from a tree with 16K words. 140
5.9 Evaluation scores of single split point experiments obtained from a tree with 22K words. 140
5.10 Comparison of our model with other unsupervised systems participated in Morpho Challenge 2010 for English. 141
5.11 Comparison with other unsupervised systems participated in Morpho Challenge 2009 for English. 142
5.12 Experiment results with concentration parameters: $\beta_s = 0.01$ and $\beta_m = 0.005$ for German with other participants in Morpho Challenge 2010. 143
5.13 Experiment results with concentration parameters: $\beta_s = 0.01$ and $\beta_m = 0.005$ for Turkish with other participants in Morpho Challenge 2010. 143

6.1 Comparison with other systems. 164
6.2 Additional morphemes captured by the system. 167
6.3 Comparison with Morfessor Baseline. 167

7.1 Clustering features 177
7.2 Clustering features (cont') 177
7.3 Some clustered morphemes in English. 180
7.4 Some clustered morphemes in Turkish. 181
7.5 Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem and morpheme position as features. 182
7.6 Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem, morpheme position and morpheme length. 183
7.7 Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem, morpheme position, last morphemes of the previous and following word. 183
7.8 Evaluation results, by weighting features, previous morpheme, following morpheme, current morpheme, stem and morpheme position in Turkish 184
7.9 Evaluation results by weighting features previous morpheme, following morpheme, current morpheme, stem, morpheme position an morpheme length. The experiment is performed on English and observed through 100 clusters. 184

A.1 45 tags in full Penn Treebank tag set. 192

List of Figures

2.1 Linguistic levels (Katamba & Stonham 2006) 37
2.2 Inflectional and derivational affixes (The figure is taken from Bubenik (1999)). 41
2.3 Plate diagram of a Dirichlet process: $DP(\alpha, H)$ 56
2.4 An illustration of the Chinese Restaurant Process. The new customer x_{N+1} sits at a table which is already occupied with a probability proportional to the number of customers sitting at the table; which is $\frac{3}{11+\alpha}$, $\frac{2}{11+\alpha}$, $\frac{1}{11+\alpha}$, and $\frac{5}{11+\alpha}$ respectively. The customer sits at a table which is empty with a probability proportional with the concentration parameter; which is $\frac{\alpha H(x_{N+1})}{11+\alpha}$. . 58

3.1 Word split points in a LSV model 67
3.2 A sample morphology from Linguistica, that can generate the words: *the, pen, pens, paper, papers, walk, walked walking, walks, work, worked, working, works, talk, talked, talking, talks, approve, approves, approved, organise, organises, organised, imagine, imagines, imagined.* 70
3.3 An HMM 86
3.4 Trigram HMM tagger 87
3.5 Contextualised HMM tagger 88
3.6 The derived model from Linguistica that employs the parts-of-speech of words as a separate list (Hu et al. 2005). 96

4.1 An illustration of the distributional clustering algorithm. 105
4.2 A sample set of syntactic clusters, and the potential morphemes in each cluster 106

4.3 An illustration of paradigm merging. $P1$ and $P2$ are merged with an accuracy measure of $Acc = 0.76$. 110

5.1 A sample tree structure. 117
5.2 A segment of a tree with internal nodes D_i, D_j, D_k having data points $\{x_1, x_2, x_3, x_4\}$. The subtree below the internal node D_i is called T_i, the subtree below the internal node D_j is called T_j, and the subtree below the internal node D_k is called T_k. 119
5.3 The plate diagram of the model, representing the generation of a word w_i from the stem s_i and the suffix m_i that are generated from Dirichlet processes. In the representation, solid-boxes denote that the process is repeated with the number given on the corner of each box. 123
5.4 A portion of a tree where leaf nodes keep the words, and the internal nodes correspond to paradigms. 127
5.5 Sampling new tree structures. a) Before sampling a new position for the node D_0. b) After inserting the node D_0 as a sibling node to D_8 130
5.6 Marginal likelihood convergence in time for datasets of size 16K and 22K. 133
5.7 An example that depicts how the word *housekeeper* can be analysed further to find more split points. 139

6.1 Plate diagram of POS tagging part of the model. 149
6.2 Plate diagram of morphology part of the model. 151
6.3 The complete joint model. 154
6.4 Many-to-1 accuracy scores obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K, and 250K. 159
6.5 One-to-1 accuracy scores obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K and 250K. 161
6.6 Variation of Information (VI) obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K and 250K. 162
6.7 One-to-one scores that vary in different iterations of Gibbs sampler. 163
6.8 Confusion matrices obtained from different corpora show how correlated found morphemes and true morphemes are. 165
6.9 Confusion matrices obtained from different corpora show how correlated found morphemes and true morphemes are. 166

7.1 Agglomerative vs divisive clustering. 174
7.2 Distance measuring in single-linkage agglomerative clustering. . 174
7.3 Distance measuring in complete-linkage agglomerative clustering. 175

7.4 Distance measuring in average-linkage agglomerative clustering 176

List of Algorithms

1 Algorithm for paradigm-capturing using syntactic categories . . 107
2 Morphological Segmentation 113
3 Creating initial tree. 128
4 Inference algorithm 131
5 The inference algorithm to infer POS tags and morphology interchangeably. 158

Acknowledgements

It is a pleasure to thank many people who made this thesis possible:

I want first to express my thanks to my supervisor, Suresh Manandhar, who shared his immense knowledge and precious time with me. I want to thank for his guidance, advice, his enthusiasm, patience, and company throughout my PhD. Without him, I would have been lost and it would have been next to impossible to write this thesis. He taught me to not be an engineer but to be a scientist, which made a radical change on my perspective towards research. He was more than a supervisor.

I am grateful to my assessor, James Cussens, who made available his support in a number of ways. I wish to thank for the helpful meetings, for his continuous feedbacks throughout the thesis, and for his remarkable suggestions on the thesis. I also would like to thank my external examiner, Mikko Kurimo, for his time to come to England. I am grateful for his useful comments on the thesis. More importantly, I would like to thank him for organising Morpho Challenge, which made me so enthusiastic about morphology.

I am grateful to a number of people who contributed this thesis technically. It is a pleasure to thank them for sharing their precious time with me. I would like to thank Ioannis Klapaftis for the helpful discussions, for inspiring ideas, and for his encouraging chats. I would like to thank Sami Virpioja for evaluating my

Morpho Challenge results, and for his effort in organising Morpho Challenge. I would like to thank Sharon Goldwater for her time to answer my questions and for sharing her evaluation script with me. I am grateful to Abdul Malik for the discussions on how to optimise the programs and I would like to thank him for running my experiments on the group server. I am indebted to Sam Simpson for correcting the English of my thesis.

I would like to express my special thanks to Filomena Ottoway for her everlasting energy to make any kind of paper work swiftly. Thanks to her, we never faced the bureaucracy. More importantly, I am grateful for her long-lasting friendship. She has been a very close friend during my PhD.

During this period, my officemates were always there. I wish to thank Ahmad Shahid; he was not only an officemate but also a gracious colleague who always showed a friendly face to me. In addition to his company, he was always there whenever I need a hand. I want to thank my other officemate, Matthew Buttler, for his friendship and nice chats everyday, which made the hard days less harder. I also would like to thank their wives, Gulmina Rextina and Vanessa Buttler, for providing their warm friendship.

I am very happy to meet Azniah Ismail, Shailesh Pandey and Shuguang Li, with whom we started our PhDs. I would like to thank them for their valuable support throughout my PhD.

I am grateful to Frank Zeyda for his hospitality during my stay in England for my viva. Not only is he a nice host, but also he has always been an enjoyable friend with his company. I would like to thank İpek Çalışkanelli for her sincere friendship; she has been always the one that I can rely on her help for anything. She is also the one who took care of printing and binding my thesis, which deserves sincere thanks for her time. I would like to thank Loukia Drosopoulou and Amy Bidgood for their warmest and sincere friendship. It is not only their company, it is more of their warmest friendship which made me feel like I am home. I would like to thank Thomas Lampert for fun conversations, entertaining mathematical discussions, and for his useful photography chats. He has been a good company during this period. I would like to thank Pierre Andrews for his valuable friendship. Not only will I remember him with his delicious muffins, but also I will remember with good memories sharing the same house for a while.

I am very pleased to meet Antonios Pentidis and I would like to thank him for his sincere friendship. I am grateful to Matthew Naylor for his warmest friendship; I always felt more positive after talking to him. I would like to thank my housemate, Malihe Tabatabaie. She has been an easy going housemate for 3 years. I would like to thank Richard Ribeiro for his devoted friendship. I wish to thank Marek-Marta Grzes for their valuable friendship, and company during our trip to Turkey. It is an unforgettable summer holiday in my memories. I would like to thank many other colleagues who made me not feel homesick and alone in Englad; Weiping (Eliza) Xu for her warm friendship and delicious Chinese food, Rasha Ibrahim and Dina Salah for not leaving me alone with our struggle, and many other colleagues who made their friendly faces available everytime, Juan Perna, Teodor Ghetiu, Michael Banks, Marco Perez Cervantes, Waleed Alsanie, Amer Alzaidi, Maria Arinbjarnar, Ioannis Korkontzelos, Andre Freire, Gul-E-Saman, Lishan Harbird, Jan Tobias Muehlberg, Alvaro Miyazawa, Marcelo Romero, Suraj Pandey, and Santa Basnet.

Although we hardly spent one year together in England, there is one person with whom we started a long-lasting friendship. Despite the long distance between us, Léa Tosold has and always will be my dear friend. I would like to send her my warmest thanks for making her friendship available throughout the thesis.

I an indebted to Hacettepe University for the financial support during the 4 years.

Last but most importantly, I would like to express my thanks to my parents for their invaluable love and support during my PhD. My special thanks are due to Eren Buğlalılar who has always been there with his love, patience, and support during these 4 years. Despite the distance, he was the one who has always stood by my side.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Some of the material contained in this thesis has appeared in the following published conference and workshop papers:

- Burcu Can, Suresh Manandhar. Unsupervised Learning of Morphology by Using Syntactic Categories. In Working Notes for the Cross Language Evaluation Forum (CLEF) 2009 Workshop, Corfu, Greece.
- Burcu Can, Suresh Manandhar. Clustering Morphological Paradigms Using Syntactic Categories. In Multilingual Information Access Evaluation Vol. I, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers Lecture Notes in Computer Science. Springer, 2010.
- Burcu Can, Suresh Manandhar. Probabilistic Hierarchical Clustering of Morphological Paradigms. In Proceedings of the European Chapter of the ACL (EACL). April 2012. (to appear)

CHAPTER 1

Introduction and Motivation

"Every beginning is difficult, holds in all sciences."
Karl Marx, preface to *Das Kapital*

1.1 Introduction

This thesis presents unsupervised statistical methods for morphological segmentation and part-of-speech (POS) tagging in natural languages. In this chapter, we provide an introduction to both fields by answering two main questions:

- Why are morphological segmentation and POS tagging needed?
- Why is unsupervised learning important for morphology learning and POS tagging?

To answer the first question, we motivate the research in this thesis by demonstrating the prominent application areas that have benefitted from both fields. As a response to the second question, we discuss why the research in this thesis focuses on unsupervised learning methods. The discussion is supported with a comparison between different learning methods that have been adopted for

morphological segmentation and POS tagging. The research questions are also presented, along with the research directions followed to answer each research question.

1.2 Morphological Segmentation

Morphology is the study of the internal structure of words. The term *'morphology'* was first introduced by the German linguist August Schleicher in 1859 (Schleicher 1859), and refers to the study of how various sub-word units combine together to form new words, through a sequence of rules. These units, called *morphemes*, are the smallest meaning bearing units in a word.

Example 1.2.1. The word *interestingly* is made up the morphemes *interest*, *ing*, and *ly*.

Morphological segmentation is the process of analysing a word by identifying its constituent morphemes. In earlier research, the process was performed manually. However, since the manual process is expensive, different ways to automate the process have been explored. Automating a process requires replacing the manual process with a system that fulfils the same function.

Which applications of morphological segmentation will benefit from the automation of morphological segmentation? The answer lies in the relation between morphology and other fields. Morphological segmentation serves a number of fields, which will be illustrated below.

1.2.1 Application Areas of Morphological Segmentation

Speech Recognition is one of the fields that benefits from morphological segmentation extensively. Speech recognition systems are mostly based on a word dictionary, along with a language model. Language models explore the characteristics of a language by investigating the sequences of constituents in the language, such as morphemes, words, utterances etc. Forming a word dictionary is troublesome, especially for morphologically rich languages (Finnish, Turkish, Arabic, etc.); at this point, morphology is adopted to cope with the infinite

number of word forms in the language. Language models consider morpheme sequences rather than word sequences. Using morphemes instead of words helps to deal with the out-of-vocabulary (OOV) words and also with data sparsity (Creutz et al. 2007). Creutz et al. (2007) propose morpheme modelling for speech recognition for Finnish, Estonian, Turkish and a dialect of Arabic spoken in Egypt (Egyptian Colloquial Arabic) which are all considered to be morphologically rich languages. Arısoy et al. (2006) model sub-word units such as stems, endings and syllables, instead of words, as recognition units for the Turkish language. Kirchhoff et al. (2006) present several approaches where morphemes are modelled, to reduce the data sparsity for Arabic. Additionally, Berton et al. (1996), Larson et al. (2000), and Roeland Ordelman & Jong (2003) exploit segmentation of compound words into their sub-words for compoundic languages, for example the Germanic family of languages (German, Swedish, Dutch etc.).

Machine Translation is another field that uses morphological segmentation. Machine Translation systems benefit from morphological information either in the pre-processing step, post-processing step, or integrating the morphological information along with the translation process. Some machine translation approaches using morphological knowledge within the pre-procesing step are Brown et al. (1993), Goldwater & McClosky (2005), and de Gispert & Mariño (2008). In these approaches, the translation process exploits morphological knowledge. Some other systems use morphological segmentation to integrate additional knowledge about words (called factored models) (Yang & Kirchoff 2006; Koehn & Hoang 2007; Avramidis & Koehn 2008). In factored models, different types of knowledge such as the lemma and part-of-speech can be used, as well as morphological information. In contrast to the approaches mentioned so far, other translation systems use morphological segmentation within the post-processing step, once the translation is completed (Minkov et al. 2007; Kristina Toutanova 2008). These systems generally use a stemmed text to perform the translation and morphological forms of words are generated within the post-processing step.

Information Retrieval researchers also benefit from morphological segment-

ation due to ambiguity and OOV words. Truncation[1] and stemming, two simple approaches, can be adopted in information retrieval to match query words with document words (Harman 1991; Krovetz 1993; Järvelin & Pirkola 2005). However, these approaches are too simple to handle morphologically rich languages, and therefore, cannot handle the ambiguity and OOV words. Stem generation is also adopted in information retrieval by generating various word forms, before matching stems with document words (Kettunen et al. 2005). Yet, stem generation does not give reliable results either, especially in morphologically rich languages. Lemmatisation is a prominent approach that solves the ambiguity problem by extracting the base forms of words, considering the context. It is similar to stemming in the sense that both methods extract the base forms of words. However, lemmatisation is more reliable than stem generation, because it considers the context. The two-level morphology by Koskenniemi (1983) is a well-known approach in morphological analysis, which is adopted for lemmatisation (Järvelin & Pirkola 2005).

Question Answering is another field that extensively uses morphological segmentation. In a question answering system, a morphological analysis is usually required for extracting questions, as well as answers retrieved. Approaches that are used for question answering are similar to information retrieval, i.e. stemming (Bilotti et al. 2004), lemmatisation (Aunimo et al. 2003). Query expansion is also adopted in question answering, where all word forms are indexed, then words are expanded with their morphological variants during retrieval (Bilotti et al. 2004).

1.3 POS Tagging

Syntax is another influential field in linguistics. While morphology is the study of the rules of how morphemes are organised in a word; syntax is the study of the rules of how words can be organised in a sentence. Each language has its own syntactic rules, in the same way that each language has its own morphological

[1] Truncation is trimming a word to match words having the same initial characters.

rules. Some languages have a free word order, whereas some have restrictions on the word order. The smallest units considered in syntax are words. Each word functions to fulfil a role in a sentence. Some words fulfil the function of expressing actions. These are typically called *verbs*. Some words fulfil the function of expressing objects that are affected by these actions. These are typically called *nouns*. Some words fulfil the function of defining some property of the nouns. These are typically called *adjectives*, etc.

Words are classified into categories according to the functions they fulfil in a sentence. These categories are called syntactic categories (or parts-of-speech - POS). Numerous natural language processing areas benefit from syntactic categorial information.

1.3.1 Application Areas of POS Tagging

Information Retrieval is one of the areas that benefits from syntactic categorial information. Information retrieval applies syntactic categorial information in several ways: during information retrieval (Croft et al. 1991), during the filtering out of irrelevant documents (Chandrasekar & Srinivas 1997), for indexing documents to reduce the index size (Chowdhury & McCabe 1998), or for weighting terms according to their syntactic contexts (Lioma & Blanco 2009).

Word Sense Disambiguation (WSD) also uses syntactic categorial information. Although POS tagging is considered by researchers as a separate problem in natural language processing today, researchers argued that POS tagging should be considered as part of WSD (Wilks & Stevenson 1998). However, Wilks (2000) argues that POS tagging should be considered as a different problem since the process does not induce any semantic information about words. In other words, POS tagging does not perform any disambiguation on sense. Numerous works exploit POS tags for the sense disambiguation (Wilks & Stevenson 1998; Yoon et al. 2006; Cai et al. 2007).

Machine Translation (MT) is one of the fields that substantially requires

syntactic information as well as morphological information. Syntactic information is adopted in MT systems either within the pre-processing step (Habash & Sadat 2006) or within a language model as a separate component in the MT system (Kirchhoff & Yang 2005; Monz 2011; Youssef et al. 2009),

Last but not least, in **Parsing** (Watson 2006; Hänig et al. 2008), **Text to Speech** (Schlünz et al. 2010; Sun & Bellegarda 2011) and also in **Name Entity Recognition** (Stevenson & Gaizauskas 2000) syntactic information is substantially re-coursed as well.

1.4 Learning Morphology and POS

As mentioned earlier, automating a process requires a system that fulfils the same function as the manual process. The literature comprises different learning model: supervised learning, unsupervised learning and semi-supervised learning.

In **supervised learning**, a system learns the process on condition that target outputs that the system should produce for a set of input data are provided to the system. Using the target outputs, the system learns how to produce similar outputs for a set of unseen input data. In other words, the system is supervised by a set of tagged data, where tags refer to the outputs of a set of inputs. In **semi-supervised learning**, a small set of tagged data is provided in addition to a large set of untagged data. Therefore, system learns from both tagged and untagged data. **Unsupervised learning** does not require any tagged data, only plain input data. Learning is led by sophisticated methods and the system learns the structures in data through these methods[2].

It is possible to see the examples of the mentioned learning mechanisms for both morphology learning and POS tagging in the literature:

The PC-KIMMO (Koskenniemi 1984), a two-level morphology[3], is a prominent example in the context of supervised morphology learning. The system

[2]Various algorithms for unsupervised learning shall be presented in Chapter 2.

[3]Morphology is defined on two levels: surface form and lexical string. Surface form expresses the word in terms of the characters that the word is made from and the lexical string consists of different forms of morphemes (Ritchie 1992).

requires a set of manually defined rules and tagged data to learn the morphology of the given data. In the recent years, it has also been possible to see the examples of semi-supervised morphology learning (Kohonen et al. 2010). Further examples of unsupervised morphology learning are reviewed in Chapter 3.

Brill's tagger (Brill 1992) is one of the prominent examples in the context of supervised POS tagging that adopts rules to learn the parts-of-speech in the text. The tagger requires a set of rules along with tagged data. The system learns the tags by adding, removing or modifying existing rules. There are also numerous examples in semi-supervised POS tagging (Clark et al. 2003; Wang et al. 2007). Further examples of unsupervised POS tagging are reviewed in Chapter 3.

In the case of any supervision being adopted, both morphology learning and POS tagging require human taggers to constitute tagged data. Especially, the manual morphological segmentation becomes very arduous, when the evolution of languages is considered. Languages are not stable and evolve every day with the adoption of new words. Therefore, discovering morphology in an unsupervised manner gives flexibility with the changes in the language. Following the motivation to avoid any tagging process, the concept of this thesis is directed towards only unsupervised learning methods.

1.5 The Interaction Between Morphology and POS

There is a strong interaction between morphology and syntax at two distinct linguistic levels. The interaction exhibits mutual effects on both levels. Morphology in any language is constrained by syntactic rules, while syntax is shaped by the morphological rules of that language. In the sentence *"She walked quickly."*. The word *quickly* is an adverb, which is enforced by the ending *-ly*. If we investigate it at the morphological level, the morphology of the word is adapted according to the syntactic rules, which requires an adverb to follow a verb in English.

This type of interaction challenges the research in both morphology learning and POS tagging. Learning mechanisms for morphology and POS can be partially or fully incorporated. Learning can be partially incorporated in the sense that the system exploits information from one of these linguistic levels to extract information from the other linguistic level. Thus, the extraction can be conduc-

ted to learn morphology by using POS information. Alternatively, the extraction can be conducted to learn POS by using morphological information.

Moreover, a full incorporation of the two learning mechanisms enables a joint learning, where morphology and POS can be learned simultaneously. The joint learning of morphology and POS is one of the research objectives in the thesis (see Section 1.6).

1.6 Research Objectives & Questions

This thesis focuses on machine learning methods for morphological segmentation and POS tagging in an unsupervised setting. The research in the thesis either tackles morphology as a separate research problem, or incorporates syntactic and morphological information within a joint learning problem. The cooperation of syntax and morphology is motivated by the high correlation between two linguistic levels, as discussed in Section 1.5. This cooperation takes place in two directions in the thesis: first cooperation attempt tries to learn the morphology using the syntactic categories which follows in a pipeline process, where the syntactic categories are assumed to be learned in a separate process; the second cooperation attempt is conducted within the same learning process where morphology and syntactic categories are learned simultaneously which is called a joint learning model. Both attempts try to discover how the two linguistic levels can be incorporated.

In addition to these goals, there are also other research directions in the thesis. We demonstrate how morphology of words can be learned by making use of hierarchical structures (i.e. trees) without using any syntactic information. By adopting trees, we learn morphological structures that are called *paradigms*[4]. Paradigms are very influential in morphology learning, giving the flexibility of capturing many word forms which do not exist in the corpus.

We also look into how morphemes can be classified, by considering their

[4]A paradigm is a morphological structure that consists of various morphemes that have the potential of combining together to create new word forms. E.g. a simple paradigm can consist of {*quick, slow*} and {*-ness, -ly*}. Therefore, these word forms can be generated from this paradigm: *quickly, slowly, quickness, slowness*.

functions in sentences in an unsupervised setting.

Therefore, the research questions and proposed research directions to answer these questions can be summarised as follows:

- **Question 1:** How can syntactic categories be incorporated to learn morphological paradigms for unsupervised morphological segmentation?

  Direction 1: Development of a novel unsupervised algorithm that captures morphological paradigms through syntactic categories that are learned in a separate process by an unsupervised algorithm.

- **Question 2:** How can morphological paradigms be captured?

  Direction 2a: Development of a novel unsupervised probabilistic model that captures morphological paradigms in a hierarchical tree structure where words are organised in tree nodes in a way that morphologically similar words are placed close to each other in the tree structure.

  Direction 2b: Definition of an algorithm that discovers hidden structures in data while also discovering a tree structure that represents the data along with the hidden structures.

- **Question 3:** Is it possible to learn syntactic categories along with morphological segmentation of words in an unsupervised joint model?

  Direction 3: Development of a novel joint learning model that learns morphology and syntactic categories simultaneously.

- **Question 4:** How can morphemes be classified according to their functions in sentences?

  Direction 4: Construction of a clustering algorithm to capture morpheme classes by using morpheme sequences in sentences.

1.7 Thesis Structure

The thesis is structured as follows:

Chapter 2 presents a detailed description of the essential background knowledge that will be referred to throughout the thesis. The chapter is organised into two main sections, in which the linguistic and machine learning backgrounds are presented separately. The linguistic background describes the morphology and syntax in general terms, along with a discussion of how the linguistic levels are related to each other. The machine learning background describes general concepts and explains a variety of estimation algorithms in machine learning that are frequently used for unsupervised learning algorithms. In addition, the section presents the most common inference algorithms which are used in this research.

Chapter 3 presents previous work on unsupervised learning of morphology and POS tagging. The chapter is organised into two main sections. In the first section, previous work on unsupervised morphology learning is demonstrated, showing the most prominent works in the field. The research is presented in two categories: deterministic and stochastic models. The section is finalised with a discussion of evaluation algorithms for morphological segmentation, and a discussion about the benefits of unsupervised learning. In the second section, previous work on unsupervised POS tagging is presented. The previous research is investigated in two categories, where the research in the first category considers POS tagging as a clustering problem and the research in the second category considers POS tagging as a sequence labelling problem (using hidden Markov models - HMMs). The chapter also presents collaborative work that combines morphological and syntactic knowledge.

Chapter 4 presents a novel clustering algorithm that captures morphological paradigms through syntactic categories. First, the chapter presents previous work that makes use of syntactic information for morphological segmentation. Then, following the motivation behind the contribution presented here, the proposed algorithm is described in detail. Finally, the experiments and evaluation scores are demonstrated for different languages.

Chapter 5 demonstrates a novel probabilistic approach that adopts a hierarchical structure to capture paradigms, along with a tree structure. The inference algorithm is described, and an explanation of how a novel word is segmented, using the learned tree structure. Finally, the chapter presents experiments and evaluation scores along with a discussion about the results.

Chapter 6 demonstrates a novel joint model in which morphology and syntax are learned cooperatively and simultaneously. Presenting previous work that attempts to combine morphology and syntax in the same learning mechanism, the description of the novel model is given in two sections, where the morphology component and POS tagging component of the model are explained separately for a clearer presentation. Finally, the inference algorithm is described, and the chapter ends with a presentation of experiments and evaluation scores for different settings that adopt various corpus sizes.

Chapter 7 presents a clustering algorithm for morpheme labelling, which labels morphemes according to their functionalities. Describing allomorphs and homophonous morphemes, two different types of morphemes, the chapter explains the hierarchical clustering algorithm that aims to capture allomorphs and homophonous morphemes. Finally, the chapter presents experiments with different settings (i.e. various features) and the evaluation scores.

Finally, **Chapter 8** concludes the thesis with a brief summary of the contributions made to the fields of morphological segmentation and POS tagging. In addition, the chapter presents a discussion about future research directions in both fields.

CHAPTER 2

Background

"He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast ."

Leonardo Da Vinci

2.1 Introduction

In this chapter, the linguistic background and the machine learning background will be presented to facilitate an understanding of the remainder of the thesis for the reader. To this end, the chapter is organised into two sections: Section 2.2 presents the linguistic background that involves the general concepts in morphology and syntax, also giving examples regarding the interaction between different linguistic levels (such as morpholoy-phonology, morphology-syntax etc.), and Section 2.3 focuses on the machine learning background with a presentation of widely-used machine learning methods that have been employed for learning morphology and syntactic categories.

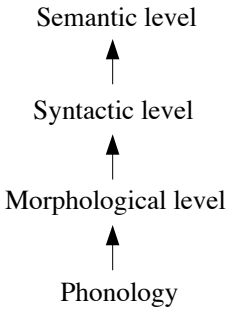


Figure 2.1: Linguistic levels (Katamba & Stonham 2006)

2.2 Linguistic Background

This section focuses on the linguistic levels, of morphology and syntax, from the linguistic perspective, by giving fundamental definitions in morphology and syntax to ease understanding of the remainder of the thesis.

2.2.1 Morphology

Components of a language, such as syntax and semantics are established on the constituents of morphology, which makes automatic acquisition of morphology an area that has drawn a lot of attention in natural language processing. Katamba & Stonham (2006) define the linguistic levels in a language as phonology, morphology, syntax, and semantics; so that each word is first constructed by the sounds, then the word structure is formed, words come together to form sentences, and finally the meanings are established. Therefore, each level bears information about the levels below it (see Fig. 2.1). Therefore, according to this definition, to analyse the morphology in a language, for example syntactic and semantic information could be useful.

Bauer (2003) depicts the history of the study of morphology from a linguist's point of view as follows:

> *The study of morphology has been influenced by all major groups of linguists: by the philologists of the nineteenth century, by the struc-*

turalists in the twentieth century, by the transformational grammarians in the second half of the twentieth century and by linguists with other theoretical orientations as well.

This indicates that morphology has a profound position in the structure of a language which has drawn the attention of researchers from different areas of linguistics. The history of morphology acquisition starts with the theory of distributional characteristics of letters by Harris (1955) (see Chapter 3 for a deeper discussion). A definition of morphology, which occupies a broad place in the literature is needed:

Morphology studies words and their inner structures (Bauer 2003). It studies the organisation of subwords to constitute different word forms which are semantically distinct from the original word. This distinction can also be syntactically true.

Example 2.2.1. The word *organise* can be derived into different morphological forms such as *organisation*, *organising*, *organised*, *organisations*, *organisational*, *organisation*, *disorganise*, *reorganise* etc. All word forms bear a different meaning whereas they can be either a noun, a verb, or an adjective.

In that sense, we can say that morphology enables the creation of new words in a language. Each language has its own rules for forming words. Morphology also studies these rules. Some languages have little or no morphology. **Isolating languages** fall into this category. For instance, Chinese is an isolating language in which words are mostly formed from syllables. In some languages, words have complex internal structures, in which different units (see 2.2.1.1 for the definitions of these units) come together and create words. These languages are called **agglutinative languages**. Finnish and Turkish are instances of agglutinative languages. Another group of languages fall into the **fusional languages** group where different units with different meanings combine to create a word form. Russian and Polish fall into this category. The following section will describe these units.

2.2.1.1 Morphemes, Affixes, Roots, Stems etc.

The smallest meaning-bearing unit of a word is called a **morpheme**. Agglutinative languages have a high number of morphemes in a word, while isolating languages have few morphemes or even none.

Example 2.2.2. The word form "*Türkçeleştiremediklerimizden mi?*" in Turkish which means "*Is it the one which we could not translate into Turkish?*" can be divided into its morphemes as: *Türk-çe-leş-tir-e-me-dik-ler-imiz-den mi?*. As it can be observed from the translation, each morpheme has a different meaning such as past tense, the third person plural, the stem, negation etc.

Remark. The term **morph** is also used alternatively in the literature referring to the physical form (a set of sounds - *phoneme*s) of a morpheme (Katamba & Stonham 2006). For example, the morph of *car* is */ka/*.

Morphemes can be classified into two groups: **free morphemes** and **bound morphemes**. Morphemes which can freely occur, without combining with other morphemes are called free morphemes. Bound morphemes can only be part of a word, and can only exist by combining with other morphemes.

Example 2.2.3. *Pen, effect, sleep* are free morphemes which can occur without combining with other morphemes, whereas *un-, de-, -ism* are bound morphemes which can only be a part of a word such as *uninteresting, deactivate, determinism*. In the word *houses*, there are two morphemes, one of which is a bound morpheme, *-s* and the other one is a free morpheme, *house*.

Free morphemes which cannot be analysed further, and to which the bound morphemes are attached, are called **root**s (Bauer 2003). The difference between a **stem** and a root is that roots cannot be divided further, whereas a stem can be divided into more morphemes.

Example 2.2.4. The word *blackboards* is segmented into the stem *blackboard* and bound morpheme *-s*, where the stem *blackboard* can be divided more to induce the roots *black* and *board*.

Morphemes can be further subdivided into **affixes** with a distinct abstract meaning (Haspelmath 2002). For example, in Turkish, there are 5 different case

affixes (*-i* for accusative, *-de* for prepositional, etc.). Affixes have different names according to the position they attach in the word. **Suffixes** attach to the end of a stem, **prefixes** attach to the front of a stem. There also other affix types such as **infixes** which are added within the stem, and **circumfixes** that are added as two parts in different positions in a word (Bubenik 1999).

Example 2.2.5. In the word *unintentionally*, *un-* is a prefix, *-ion* , *-al* and *-ly* are suffixes. In Bontoc, a language of Philippines, *fikas* (strong) becomes *fumikas* (to be strong) with the infixation.

Base is a part of a word to which any affix is attached (Bauer 2004). Both stems and roots are a special case of a base.

Example 2.2.6. The base of the word *successful* is *success* to which the affix *-ful* is attached, another affixation[1] is applied with the affix *-ly* which is added to the base *successful*.

2.2.1.2 Allomorphs

Allomorphs are morpheme variants (or morpheme alternants) that differ in shape from each other (Haspelmath 2002; Bauer 2004). The shape refers to the phonetic representation of a morpheme, this may only occur in pronunciation or in written form, that yield a change in the phonetic representation.

Example 2.2.7. The English plural may be pronounced differently in different words such as in *cats* as [s], in *dogs* as [z], and in *faces* as [az] although the morpheme forms are written exactly the same[2].

In some languages allomorphs are very common due to vowel harmony. Vowel harmony is the adaptation of morphemes phonologically according to the adjacent morphemes.

Example 2.2.8. Turkish is a language which has intensive vowel harmony, making allomorphs very common. For example, the accusative case of a word might

[1] Affixation is to attach affixes to a base.
[2] The example is taken from Haspelmath (2002)

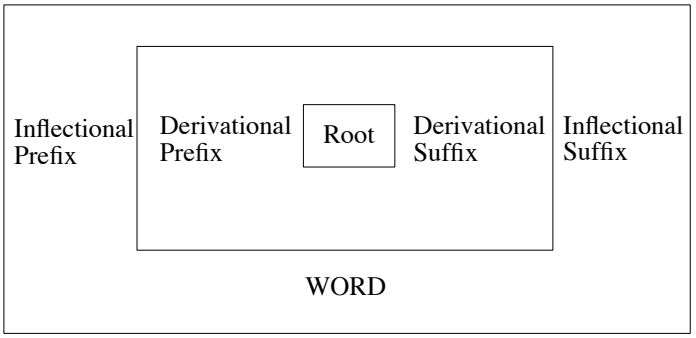


Figure 2.2: Inflectional and derivational affixes (The figure is taken from Bubenik (1999)).

have several different forms due to vowel harmony: [*ı*], [*i*], [*u*], [*ü*].

ı	*baraj*	dam	*baraj-ı*	the dam
i	*kent*	town	*kent-i*	the town
u	*koyun*	sheep	*koyun-u*	the sheep
ü	*üzüm*	grape	*üzüm-ü*	the grape

2.2.1.3 Inflectional and Derivational Morphology

Affixes are further analysed into two categories: **inflectional** and **derivational** affixes. Inflectional affixes construct new word forms, whereas derivational affixes create new words. Here another term arises to distinguish words from word forms. Each word form that is constructed with an inflection belongs to the same **lexeme**, whereas each word which is created by a derivation is a distinct word.

Example 2.2.9. The word forms *buying*, *buys*, *bought* belong to the same lexeme *buy*. In other words, these word forms are generated by adding inflectional affixes (in the example, tenses) to the same root. However, the word *buyer* is a different lexeme that is generated by adding the derivational affix *-er* to the root.

Fig. 2.2 depicts how a grammatically correct word is generated through inflection and derivation. New lexemes are first constructed through the derivational process, and an inflectional affixation follows a derivation.

2.2.2 Syntax

Syntax studies how sentences are formed through syntactic rules of a language. The rules govern how words are ordered in a sentence. As Tallerman (1998) remarks, every language has a syntax.

There have been several approaches to syntax which have different aspects: relational grammar, categorial grammar, universal grammar, cognitive grammar etc. One of the prominent theories in the literature has been universal grammar which originates from the work of Chomsky (1965). According to the theory of universal grammar, every language has its own syntax, however languages share a common set of properties which are limited in the human brain, and that makes them universal.

2.2.2.1 Syntactic Categories

Under syntactic rules, the main criteria that restricts the sequence of words are their classes. These classes or syntactic categories (grammatical categories, parts-of-speech etc.) consist of nouns, verbs and adjectives. These are the fundamental syntactic categories that exist in almost every language. What changes in different languages is the subcategories such as adverbs, determiners, conjunctions, and so on.

According to the typical definition of these terms, a **noun** is either the subject or the object in a given sentence, which is affected by an action; a **verb** is the action or state in a sentence; an **adjective** is the word that describes a noun, and a **preposition** links the words in a sentence by describing their relationship. Although these definitions do not change from one language to another, the order in the sentence may differ.

Example 2.2.10. In English, adjectives precede the nouns that they define; however, in Spanish, it is vice versa. For example, "a young boy" becomes "*un chico joven*" in Spanish, where *chico* means *boy* and *joven* means *young*.

Therefore, even if a word is unknown in a sentence, it is possible to tell its category just by looking at the sentence (even a small portion of the sentence that surrounds the word, i.e. the context of the word).

Example 2.2.11. : In the sentence, "The blue *skafer* is on the table.", although the reader might not know what *skafer* means, she will be able to tell that it is a noun.

It is also possible to tell whether a word belongs to a category through a **substitution test** (Manning & Schütze 1999). The substitution test shows that if the words belong to the same syntactic category, they should be able to be substituted for each other.

Remark. The context of a word is a vital clue to its syntactic category. In Chapter 3, we present the prominent work in syntactic acquisition, where it will be noticed that all the research makes use of the context of a word, to induce its syntactic category.

2.2.2.2 Open-Class vs Closed-Class Syntactic Categories

Syntactic categories are traditionally divided into two classes: open class categories (or lexical categories) and closed class categories (or functional categories). Open categories are classes which accept many members or where the number of members is indefinite, however, the number of members of a closed category is definite and do not receive any new members[3]. Open categories are normally universal categories such as nouns, adjectives, and verbs; whereas, closed categories consist of conjunctions, determiners, prepositions, and so on.

2.2.3 Interaction Between Linguistic Levels

2.2.3.1 The Morphology-Phonology Interaction

Morphology is influenced by phonology, which determines the use of sounds within morphemes. Allomorphs and vowel harmonisation are two samples of this interaction between morphology and phonology.

Example 2.2.12. Another example is the indefinite article *a/an* in English. When words begin with a consonant, *a* is used, whereas the article becomes *an* with words that begin with a vowel (Katamba & Stonham 2006).

[3]The number of members of a closed category is said to be twenty to thirty at most (Emonds 1985)

2.2.3.2 The Morphology-Syntax Interaction

The interaction between morphology and syntax is shaped by the inflection and derivation of words. Inflection of a word is determined by the syntactic rules in a sentence. In other words, the appropriate word form is chosen according to the syntactic structure of the sentence.

Example 2.2.13. The word *talk* is exposed to affixation to form a grammatically correct sentence. For example, it can be in the past tense form *talked*, progressive form *talking*, present tense form *talks* etc. It should be noted by the reader that all these forms belong to the same syntactic category, which is 'verb'.

In contrast to inflection, derivation may change the syntactic category of a word.

Example 2.2.14. The words *personal*, *personally*, *personalise* belong to different syntactic categories (adjective, adverb, and verb respectively) which are generated from the root *person*.

2.2.3.3 The Morphology-Semantics Interaction

Obviously each word/word form bears a distinct meaning, therefore, yielding a tight interaction between morphology and semantics. Independently from the affixation type (either inflection or derivation) each word form has a semantic analysis.

Example 2.2.15. Different affixations result in different meanings; such as in Turkish, with the inflection, it is possible to embed the person, tense, passive voice, obligation, possibility, conditionality etc. within the word. For example, *gid-iyor-um* ('I am going') has the meaning of a first person singular and present continuous tense, whereas *git-meli-y-di-ler* ('they must have gone') has the meaning of an obligation, a past tense, and a third person plural, and more than that the word *git-me-meli-y-se-m* ('if I should not go') has the meaning of a negation, an obligation, a conditionality, and a first person singular.

2.3 Machine Learning Background

After giving the linguistic preliminaries, this section presents the basic computational preliminaries, which include the learning approaches that have been used for NLP tasks, especially for morphology learning and POS tagging. All the approaches presented can be considered as a model learning where the actual data needs to be represented by a model with a set of parameters. Learning refers to the estimation of these parameters. The approaches diverge according to the parameter estimation techniques employed.

Here, we mainly refer to the mathematical models. A **mathematical model** typically describes real world phenomena in terms of mathematical terms. In other words, real world phenomena, e.g. the behaviour of a system, are described by using mathematical language.

Example 2.3.1. A sample model could be one which measures the total energy of a body at rest, which is a well known equation in natural sciences:

$$E = mc^2 \tag{2.1}$$

which describes energy in terms of its mass and the speed of the light. Here the equation consists of a variable and a constant to infer a quantitative measure about the observed data. A **variable** is an attribute which may change for different types of problems; whereas a **constant** has a stable value which does not change from one problem to another.

Moreover, we will refer to statistical models specifically. A **statistical model** is a mathematical model in which everything is defined in terms of probability distributions. The aim of modelling is either summarising the data, or making predictions about future observations, which are defined as probability distributions.

Example 2.3.2. In a probabilistic model, a set of data that is known to be in the form of a Gaussian distribution can have parameters, a mean and a variance which need to be estimated through a learning procedure, where the aim is to define the probability distribution that will summarise the data in a Gaussian

form. The definition of a **parameter** is a quantity which describes the relation between variables in the model.

A well-defined statistical model can be considered as a black box which outputs predictions based on the input. These predictions can be referred to as **hypotheses**.

Example 2.3.3. For a morphological analysis system, a hypothesis H_1 refers to any morphological segmentation of the data D. Therefore, each hypothesis (each distinct analysis of the data) defines a probability that yields to the probability distribution $p(H|D)$ where H includes all the hypotheses.

Various estimation methods with distinct principles have been used in the literature when looking for the hypothesis that explains the data best. We will describe the most prominent parameter estimation methods that have been used for morphological segmentation and syntax acquisition.

2.3.1 Maximum Likelihood Estimate

Maximum Likelihood Estimate (MLE) outputs the hypothesis that leads to the highest probability of data under this hypothesis. The probability of data under a hypothesis is called the **likelihood**. Let the likelihood of the data under a model with parameters θ be $L(\theta|D)$ where $D = \{x_1, x_2, \ldots, x_k\}$, that is:

$$L(\theta|D) = p(D|\theta) \tag{2.2}$$

The MLE estimate $\hat{\theta}$ is given as:

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} p(D|\theta) \\ &= \arg\max_{\theta} p(x_1, x_2, \ldots, x_k|\theta) \\ &= \arg\max_{\theta} \prod_{i=1}^{k} p(x_i|\theta) \end{aligned} \tag{2.3}$$

if we assume that data is distributed independently and identically (iid).

The MLE employs only the observed data to find a good estimate of parameters. Therefore, model and empirical distributions are identical and have a minimum divergence between. For this reason, maximum likelihood estimation is also regarded as the minimisation of the Kullback-Leibler (KL) divergence. The KL divergence is a non-symmetric measure that defines how different two distributions are. It is computed as follows:

$$D_{KL}(P \parallel Q) = \sum_{x \in D} P(x) \log \frac{P(x)}{Q(x)} \tag{2.4}$$

Hence, the MLE can be redefined by using KL divergence between the model and the empirical distribution:

$$\hat{\theta} = \arg\min_{\theta} D_{KL}(P_\theta \parallel P_D) \tag{2.5}$$

where P_θ denotes the model distribution and P_D denotes the empirical distribution.

With a small set of data, the MLE may mislead us to an estimation which is far from estimating the data generating probability distribution accurately. Therefore, it is crucial to have sufficient data to have a decent estimate.

Concerning morphological segmentation, the hypothesis that outputs the analysis such that the stem is the full word and the suffix is an empty character leads to a MLE. The reason is that the hypothesis that does not split words is exactly the same as the empirical data, which makes the KL divergence between the hypothesis and the empirical data zero[4].

Example 2.3.4. Let the empirical data consist of words *walked, talked, walking, washed, washing*. If we consider the hypothesis that outputs the words without splitting, such that *walked*+$\emptyset$, *talked*+$\emptyset$, *walking*+$\emptyset$, *washed*+$\emptyset$, washing+$\emptyset$ yields a probability of $(\frac{1}{5})^5$ for stems and 1^5 for suffixes, therefore a probability of 0.00032 for the hypothesis. However, if we consider another hypothesis that splits the words such that *walk*+*ed, talk*+*ed, walk*+*ing, wash*+*ed, wash*+*ing*, the

[4]See Goldwater (2007) for more discussion about the same argument.

probability of stems becomes ($\frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} \times \frac{2}{5} \times \frac{2}{5}$) and the probability of suffixes becomes ($\frac{3}{5} \times \frac{3}{5} \times \frac{2}{5} \times \frac{3}{5} \times \frac{2}{5}$); therefore the entire probability of the hypothesis becomes 0.00018. The hypothesis that does not suggest any split points has got a higher probability than the one which suggests split points. The difference becomes more explicit with larger corpora.

2.3.2 Maximum A Posteriori Estimate

MLE does not involve a prior distribution over hypotheses. However, in some cases, it is useful to define a prior probability distribution over hypotheses. The prior probability distribution is defined before the data is considered. The probability which is defined before seeing the data is called the **prior probability**. The likelihood and the prior probability determine the **posterior probability**, which will lead us to a Maximum a Posteriori Estimate (MAP). The same formulation that is given in Equation 2.3, but now including prior information about the hypothesis is given as follows for the MAP estimate:

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} p(\theta)p(D|\theta) \\ &= \arg\max_{\theta} p(\theta)p(x_1, x_2, \ldots, x_k|\theta) \\ &= \arg\max_{\theta} p(\theta)\prod_{i=1}^{k} p(x_i|\theta) \end{aligned} \tag{2.6}$$

Prior information could be either informative or non-informative. **Informative prior** gives expressive information about each hypothesis; whereas **non-informative** does not give any significant information about the hypotheses. A typical non-informative prior is a uniform distribution which assigns an equal probability to each hypothesis.

Both ML and MAP estimates serve as an optimisation problem where a point estimate is searched which will result in one solution. Using these two estimates it is not possible to estimate a probability distribution over hypotheses or parameters. Bayesian modelling introduces a different perspective from these

two estimation methods by discovering the estimate in the form of a probability distribution.

2.3.3 Bayesian Modelling

Bayesian modelling originates with the idea of having a probability distribution over the target instances (either parameter values, latent variables, or hypotheses). A Bayesian model can be either a parametric or a nonparametric statistical model.

Bayesian modelling derives from **Bayes' theorem**[5]:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{2.7}$$

Bayes' theorem defines an inverse probability distribution over the parameters using the likelihood and the prior probability. The inverse probability distribution over parameters given the data is called the **posterior probability**. Posterior probability defines how probable the parameter values for the observed data are, by taking into account the likelihood and the prior probability. Here the probability of the data which is computed through all possible values of the parameters is used for normalisation. It is the **marginal probability** of data. According to the type of the parameter values, either discrete or continuous, it can be calculated in two different ways. If the parameters are discrete:

$$p(D) = \sum_i p(D|\theta_i)p(\theta_i) \tag{2.8}$$

whereas if the parameters are continuous:

$$p(D) = \int p(D|\theta)p(\theta)d\theta \tag{2.9}$$

In addition to defining a posterior probability over the parameter values, Bayes' theorem, can also be used to update the parameter values, whenever new data is seen. This way of learning the parameter values is called **online learn-**

[5]Bayes' theorem was proposed by Thomas Bayes (c. 1702 –17 April 1761), an English mathematician.

ing where the data is continuously employed to improve the current parameter values during learning. Another type of learning is **batch learning**, which takes the entire data to perform the learning by fixing the parameter values at the end. Whatever learning method is followed, the final aim is to estimate the parameters.

2.3.4 Minimum Description Length Principle

Another estimation method is Minimum Description Length (MDL) principle. Differently from other estimation methods described earlier, the MDL is based on information theory. The method was proposed by Rissanen (Rissanen 1978, 1989). The MDL performs reasoning based on information theory, by searching for the hypothesis which will lead to the best compression of the data. Compression of the data is performed over the regularities in the data (Grünwald 2005). Grünwald (2005) states that, the more compression is performed, the more is learned from the data.

Example 2.3.5. If the MDL is applied to morphological segmentation, the hypothesis which leads to the corpus that occupies the minimum space is chosen as the output analysis of the data. While compressing the corpus, regularities (which are the common morphemes) are determined, to minimise the data length. Let a sample data set be $D = \{walked, talked, walking, talking\}$. If the data is compressed as $D = \{walk, talk, ed, ing\}$, which has all the vital information about the actual data, the true morphological analysis of the corpus is considered to have been discovered.

It it essential to mention that the MDL principle can be defined using different frameworks, such as a frequentist approach, or a Bayesian approach. In either case, the main principle remains the same, which is to express the data in a compact state.

Remark. More discussion about MDL is given in Section 3.2.1.2 where also some examples from the literature are also presented.

2.3.5 Integration over Parameters

Some estimation approaches which aim to estimate the parameter values are discussed above. Another perspective lies in an integration over all possible values of the parameters without fixing them. Therefore, any inference of the latent variables is carried out through an integration over the parameters without estimating them. Imagine the latent variables are the segmentation points in each word where S represents the set of all segmentations in corpus D:

$$p(S|D) = \int p(S|D,\theta)p(\theta|D)d\theta \tag{2.10}$$

where θ denotes the set of parameters. As seen in the equation, the parameters θ are integrated out without being estimated. Therefore, the latent variables are inferred from all possible values of the parameters, which eliminates the probability of having a biased estimation of parameters, by leading to a more robust and correct inference.

If Bayesian modelling is adopted, prior probability over the parameters can be defined. In the integration over parameters, it is also very convenient to use **conjugate priors** which enable the integration procedure to become tractable. Conjugate priors have a way of neutralising the form of the likelihood probability distribution, in a way that the posterior probability distribution results in the same form as the prior probability distribution.

For example, a Multinomial distribution has a conjugate prior in the form of a Dirichlet distribution. Therefore any Multinomial-Dirichlet conjugation results in a posterior distribution, with a Dirichlet distribution form. Let a Multinomial distribution be defined on a set of possible outcomes $\{1,\ldots,K\}$ with parameters θ, and the prior probability distribution be defined for the parameters in a Dirichlet distribution form with **hyperparameters**[6] β:

[6]The parameters of a prior probability distribution are hyperparameters.

$$\begin{aligned} x_i &\sim Multinomial(\theta) \\ \theta &\sim Dirichlet(\beta) \end{aligned} \tag{2.11}$$

Here the first line states that the variable x_i is drawn from a Multinomial distribution with parameters θ and the second line declares that the parameters θ are drawn from a Dirichlet distribution with hyperparameters β. The definition of the Dirichlet distribution follows the form:

$$p(\theta|\beta) = \frac{1}{B(\beta)} \prod_{k=1}^{K} \theta_k^{\beta_k - 1} \tag{2.12}$$

where $B(\beta)$ is a normalising constant in a beta function form:

$$B(\beta) = \frac{\prod_{k=1}^{K} \Gamma(\beta_k)}{\Gamma(\sum_{k=1}^{K} \beta_k)} \tag{2.13}$$

where Γ is the gamma function. The gamma function is defined as $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$ for positive complex numbers, whereas it becomes $\Gamma(t) = (t-1)!$ for positive integers.

The Multinomial distribution is defined on the outcomes $x = \{x_1, \ldots, x_k\}$ as follows:

$$p(x|\theta) = \frac{N!}{\prod_{k=1}^{K} n_k!} \prod_{k=1}^{K} \theta_k^{x_k} \tag{2.14}$$

where N represents the total number of data points belonging to one of the

possible outcomes:

$$N = \sum_{k=1}^{K} n_k \tag{2.15}$$

Here, n_k denotes the number of data points belonging to the outcome x_k.

The first factor in Equation 2.14 provides the exchangeability over the variables, the second factor computes the probability of observing each outcome.

After giving the preliminaries about Multinomial and Dirichlet distributions, we can now apply Bayes' rule to define a posterior distribution over the parameters θ:

$$\begin{aligned} p(\theta|x,\beta) &\propto p(x|\theta)p(\theta|\beta) \\ &= \frac{N!}{\prod_{k=1}^{K} n_k!} \prod_{k=1}^{K} \theta_k^{n_k} \frac{\prod_{k=1}^{K} \Gamma(\beta_i)}{\Gamma(\sum_{k=1}^{K} \beta_k)} \prod_{k=1}^{K} \theta_k^{\beta_k - 1} \\ &= \frac{N!}{\prod_{k=1}^{K} n_k!} \frac{\prod_{k=1}^{K} \Gamma(\beta_i)}{\Gamma(\sum_{k=1}^{K} \beta_i)} \prod_{k=1}^{K} \theta_k^{n_k + \beta_k - 1} \\ &\propto Dirichlet(n_k + \beta_k - 1) \end{aligned} \tag{2.16}$$

As can be seen in Equation 2.16, the posterior probability distribution is in a Dirichlet form, like the prior probability distribution, but with a new set of parameters.

It makes it tractable to use conjugacy when making predictions from the data through an integration:

$$
\begin{aligned}
p(x_{N+1}=j|x,\beta) &= \int p(x_{N+1}=j|x,\theta)p(\theta|\beta)d\theta \\
&= \int \theta_j \frac{\Gamma(N+\sum_{k=1}^{K}\beta_k}{\prod_{k=1}^{K}\Gamma(n_k+\beta_k)} \prod_{k=1}^{K} \theta_k^{n_k+\beta_k-1} d\theta \\
&= \frac{\Gamma(N+\sum_{k=1}^{K}\beta_k}{\prod_{k=1}^{K}\Gamma(n_k+\beta_k)} \int \theta_j \theta_j^{n_j+\beta_j-1} \prod_{k\neq j}^{K} \theta_k^{n_k+\beta_k-1} d\theta \\
&= \frac{\Gamma(N+\sum_{k=1}^{K}\beta_k}{\prod_{k=1}^{K}\Gamma(n_k+\beta_k)} \int \theta_j^{n_j+\beta_j} \prod_{k\neq j}^{K} \theta_k^{n_k+\beta_k-1} d\theta \\
&= \frac{\Gamma(N+\sum_{k=1}^{K}\beta_k}{\prod_{k=1}^{K}\Gamma(n_k+\beta_k)} \frac{\Gamma(n_j+\beta_j+1)\prod_{k\neq j}^{K}\Gamma(n_k+\beta_k)}{\Gamma(N+\sum_{k=1}^{K}\beta_k+1)} \\
&= \frac{n_j+\beta_j}{N+\sum_{k=1}^{K}\beta_k}
\end{aligned}
\tag{2.17}
$$

As we can see in the second line, the posterior distribution of the parameters which is derived in Equation 2.16 is used as a prior probability distribution over the parameters. The fifth line is derived by considering the probability distribution must sum up to 1. Therefore, the inverse of the normalising constant in the Dirichlet probability distribution (given in Equation 2.13) is inserted as the result of the integration. The final result is interpreted by a rich-get-richer behaviour, where the observation x_{N+1} has a higher probability of belonging to a category if the number of previous observations in that category is greater in number. It is noteworthy to mention that Equation 2.17 has derived the posterior mean.

It is also possible to consider this as a mixture model with K components, where each component is made up of n_k data points. According to Equation 2.17, components attract more data points in proportion with the number of the data points they have (see Section 2.3.6 for a further discussion). If we take the infinite limit of the number of components, by considering the probability of creating a new mixture component, the final equation is modified accordingly:

$$p(x_{N+1} = j|x, \beta) = \begin{cases} \frac{n_j}{N+\sum_{k=1}^{K} \beta_k} & j \in K \\ \frac{\beta_j}{N+\sum_{k=1}^{K} \beta_k} & \text{otherwise} \end{cases} \tag{2.18}$$

With the new perspective for having a new category, either the new data point is assigned to an existing category, with a probability that is proportional to the number of data points in that category, or it is assigned to a new category with a probability in proportion to the hyperparameter defined for that category. The perspective gives a natural smoothing by leaving some of the probability mass for the unseen events. The perspective can be applied to any type of model selection problem, such as determining the number of hidden states in a hidden Markov model[7], determining the number of latent variables in a latent variable model, etc. (Orbanz & Teh 2010). This leads to non-parametric Bayesian modelling which will be explained thoroughly, below.

Remark. In Section 2.3.7 Dirichlet process mixture models shall be explained in depth by deriving the Equation 2.18 from a mixture model perspective.

2.3.6 Bayesian Non-Parametric Modelling

The term *non-parametric* causes serious confusion in the field by suggesting the idea that there are no parameters in the model. However, *non-parametric* means that there are an infinite number of parameters. In other words, in a non-parametric model the number of parameters can grow with the data. In real life, data is very complicated. Therefore, Bayesian non-parametric models present a more realistic and flexible framework to capture the irregularities in the data by permitting flexibility in the parameter space.

A well-known approach in Bayesian non-parametric modelling is the **Dirichlet Processes**. A Dirichlet process defines a probability distribution over an infinite number of objects (Orbanz & Teh 2010). Imagine we have a space of partitions $A = \{A_1, \ldots, A_K\}$; each consisting of a set of data points. Each draw from a Dirichlet Process is a distribution over a variable number of partitions

[7] See Section 3.3.2 in Chapter 3 for the definition of a hidden Markov model (HMM).

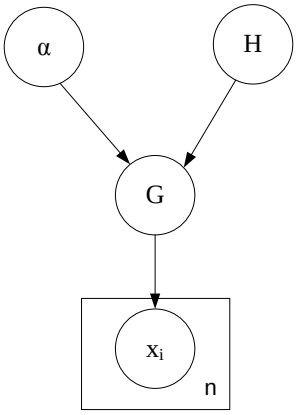


Figure 2.3: Plate diagram of a Dirichlet process: $DP(\alpha, H)$

which may also vary in size. In addition, probability distributions belonging to each partition are Dirichlet distributed:

$$(G(A_1), \ldots, G(A_K)) \sim Dirichlet(\alpha H(A_1), \ldots, \alpha H(A_k)) \quad (2.19)$$

Here α is a concentration parameter, which determines the variance between the probability distributions of each partition, whereas H is a base distribution and is the mean of the probability distributions. A summarised definition presented in support of an example, where the data points $x = \{x_1, \ldots, x_N\}$ are generated from a Dirichlet process $DP(\alpha, H)$ with a concentration parameter α and a base distribution H (see Figure 2.3):

$$\begin{aligned} x_i &\sim G \\ G &\sim DP(\alpha, H) \end{aligned} \quad (2.20)$$

We can apply the same integration procedure to integrate out the probability distribution G for the estimation of the latent variables or future observations, as

discussed in Section 2.3.5. Here we apply it for a future observation $x_{N+1} = j$ (Blackwell & MacQueen 1973):

$$\begin{aligned} p(x_{N+1} = j|x, \alpha, H) &= \frac{1}{N+\alpha}\sum_{i=1}^{N} I(x_i = j) + \frac{\alpha}{N+\alpha}H(j) \\ &= \frac{n_j + \alpha H(j)}{N+\alpha} \end{aligned} \tag{2.21}$$

Here I is an identity function that outputs 1, if $x_i = j$, otherwise the function outputs 0.

This leads us to a well-known perspective in Dirichlet process which is the Chinese Restuarant Process (CRP). Imagine a restaurant that consists of an infinite number of tables with an infinite number of seats at each table where each customer chooses a table and sits down(see Figure 2.4). At each table, a different type of meal is served. The customer chooses an occupied table with a probability which is proportional to the number of customers who are already sitting at the table, whereas she chooses an empty table with a probability proportional to a defined constant α. Therefore, tables which have a great number of customers attract more customers according to the rich-get-richer principle. A particular setting of a table with N customers has a joint probability of:

$$\begin{aligned} p(x_1, \ldots, x_N) &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)\ldots p(x_N|x_1, \ldots, x_{N-1} \\ &= \frac{\alpha H(x_1)}{\alpha}\frac{(\alpha H(x_2)||1)}{1+\alpha}\frac{(\alpha H(x_3)||1||2)}{2+\alpha}\cdots\frac{(\alpha H(x_N)||1||2|\ldots)}{N-1+\alpha} \\ &= \frac{\alpha^K}{\alpha(1+\alpha)(2+\alpha)\ldots(N-1+\alpha)}\prod_{i=1}^{K} H(x_i)\prod_{i=1}^{K}(n_{x_i}-1)! \\ &= \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}\alpha^K\prod_{i=1}^{K} H(x_i)\prod_{i=1}^{K}(n_{x_i}-1)! \end{aligned} \tag{2.22}$$

In Equation 2.22, the second line chooses one of the factors depending on

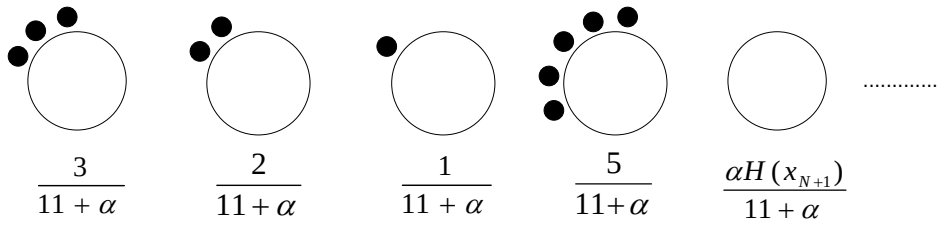


Figure 2.4: An illustration of the Chinese Restaurant Process. The new customer x_{N+1} sits at a table which is already occupied with a probability proportional to the number of customers sitting at the table; which is $\frac{3}{11+\alpha}$, $\frac{2}{11+\alpha}$, $\frac{1}{11+\alpha}$, and $\frac{5}{11+\alpha}$ respectively. The customer sits at a table which is empty with a probability proportional with the concentration parameter; which is $\frac{\alpha H(x_{N+1})}{11+\alpha}$.

whether the table is occupied or has just been created. Therefore, for each table, at least one creation is performed which forms the second factor and the third factor in the last equation. Once the table is created, factors that represent the number of customers sitting at each table are chosen accordingly, which forms the last factor in the last equation.

The Chinese Restaurant Process explains a Dirichlet process from a restaurant perspective. There are also other perspectives that explains a Dirichlet process, i.e. the stick-breaking process (Sethuraman 1994; Ishwaran & James 2001), and the Pitman-Yor process (Pitman 1995; Pitman & Yor 1997).

2.3.7 Mixture Models

In real life, data is not so simple that it can be explained as a group of data in which the members have similar properties, so that this can be considered as one class. In contrast, real life data is generally complex and made up of various subpopulations that have items sharing similar properties within the same subpopulation. The target of mixture models is to find out about the underlying populations. To this end, mixture models have been applied to various types of problems due to their capability to overcome the complexity of the data. The applied problems include density estimation, clustering, latent class analysis, etc.

A simple mixture model with k mixture components $\{c_1, \ldots, c_k\}$ where each

of them having a density of f_i, has the following form:

$$f_i(x) = \sum_{j=1}^{N} \pi_i f_i(x_j) \tag{2.23}$$

where π_i denotes the weight of the mixture component. This is a simple finite mixture model, where the number of components is fixed. It is also possible to apply it for an infinite number of components. Imagine that the weights of the clusters follow a Multinomial distribution with parameters θ. In addition, a prior distribution is defined for θ in a Dirichlet form with hyperparameters β (follows Rasmussen (2000)):

$$\begin{aligned} \pi_i &\sim Multinomial(\theta) \\ \theta &\sim Dirichlet(\beta) \end{aligned} \tag{2.24}$$

If Multinomial-Dirichlet conjugation is applied while integrating out the parameters θ (see 2.3.5, we get the following joint distribution over the weights:

$$\begin{aligned} p(\pi_1, \ldots, \pi_N|\beta) &= \int p(\pi_1, \ldots, \pi_N|\theta_1, \ldots, \theta_k) p(\theta_1, \ldots, \theta_k|\beta_1, \ldots, \beta_k) d\theta_1 \ldots \theta_k \\ &= \frac{\Gamma(\beta)}{\Gamma(N+\beta)} \prod_{i=1}^{k} \frac{\Gamma(n_{c_i} + \beta/k)}{\Gamma(\beta/k)} \end{aligned} \tag{2.25}$$

where n_{c_i} represents the number of elements in c_i. The conditional distribution of a component indicator, given the rest of the components, is defined as follows, which is to be used for sampling (following Equation 2.17):

$$p(c_i = j|c_{-i}, \beta) = \frac{n_{-i,j} + \beta/k}{N - 1 + \beta} \tag{2.26}$$

From here, it is possible to define the model as an infinite mixture model

where $k \leftarrow \infty$. If the infinite limit of Equation 2.26 is taken, the following equations are obtained depending on whether an existing mixture component is used, or a new component is created respectively:

$$p(c_i = j|c_{-i}, \beta) = \frac{n_{-i,j}}{N - 1 + \beta} \tag{2.27}$$

$$p(c_i = j|c_{-i}, \beta) = \frac{\beta/n}{N - 1 + \beta} \tag{2.28}$$

Remark. Infinite mixture models are briefly mentioned in Section 2.3.5 and the equation obtained here is also derived, to explain the Multinomial-Dirichlet conjugation from a mixture model perspective.

After giving the definition of a mixture model, the Dirichlet Process Mixture Models (DPMMs) will now be explained. DPMMs (Antoniak 1974; Ferguson 1983) are like infinite mixture models where a model is composed of an infinite number of mixture components. Each mixture component of a Dirichlet process mixture model is drawn from a Dirichlet process (see Section 2.3.6 for a detailed discussion on Dirichlet processes). Imagine we convert the model explained above into a DPMM:

$$\begin{aligned} G &\sim DP(\beta, G_0) \\ \pi_i &\sim G \\ y_i &\sim F(\pi_i) \end{aligned} \tag{2.29}$$

In the modified model, each mixture component's parameters π_i are drawn from a mixture distribution G which is generated by a Dirichlet process $DP(\beta, G_0)$ with base distribution G_0 and concentration parameter β. Members of each mixture component are drawn from the component's distribution $F(\pi_i)$.

2.3.8 Inference

Inference of the parameters is an essential part in a learning mechanism. Parameters of a model or if needed, latent variables in the data, are inferred using various approaches, such as MAP or ML (discussed in Section 2.3.2 and Section 2.3.1), which give a point estimate for the parameters as mentioned above. However, sometimes it is needed to guess the true nature of the parameters by estimating their posterior probabilities. A thorough Bayesian inference requires an estimation of the distributions over the possible values of the parameters instead of a point estimate.

One common way to estimate the parameters' posterior distributions is to draw random samples from their posterior distributions. Drawing random samples from a distribution is called **sampling**. Markov Chain Monte Carlo (MCMC) methods constitute a big portion of the sampling algorithms in machine learning, and will be presented shortly in the following section.

2.3.8.1 Markov Chain Monte Carlo (MCMC)

MCMC algorithms are designed to find out about complex probability distributions. They are usually used in Bayesian statistics where the underlying posterior probability distribution is unknown. These probability distributions are generally posterior distributions that need to be modelled. In an MCMC algorithm, samples are drawn from a sequence of probability distributions where the samples form a Markov chain. A Markov chain is made up of a sequence of states. Let the sequence of states be $X = \{X_1, X_2, \ldots, X_n\}$. With the Markov property, each state is dependent only on the previous state:

$$p(X_{n+1} = x | X_1 = x_1, \ldots, X_n = x_n) = p(X_{n+1} = x | X_n = x_n) \tag{2.30}$$

With a random sampling from the distribution that is being estimated, the Markov chain should converge to a distribution over states, which is called an equilibrium. Gibbs sampling and Metropolis-Hastings algorithm are two prominent examples of the MCMC algorithms.

2.3.8.1.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm was first proposed by Metropolis et al. (1953) for the Boltzman distribution. The algorithm was enhanced for other types of distributions by Hastings (1970). The algorithm is based on random draws from a series of distributions, where after a number of iterations, the distribution from which samples are drawn becomes the target distribution. Each sample that is drawn is subjected to an acceptance-rejection rule, where the sample might be added to the Markov chain or might be rejected, and therefore another new sample is drawn. Let a Markov chain consist of states $\mathbf{X} = \{\ldots, \mathbf{X}^{(t-2)}, \mathbf{X}^{(t-1)}, \mathbf{X}^{(t)}\}$ at various time intervals. To determine the following state $X_1^{(t+1)}$ in the next time interval $(t+1)$, a new state is generated that only depends on the current state $X^{(t)}$. The transition is based on a proposal distribution $q(X|X_1^{(t)})$. The new state $X_1^{(t+1)}$ is accepted if:

$$\alpha < \frac{p(X_1^{(t+1)})}{p(X_1^{(t)})} \frac{q(X_1^{(t)}|X_1^{(t+1)})}{q(X_1^{(t+1)}|X_1^{(t)})} \tag{2.31}$$

where α is a random value drawn from $\alpha \sim Uniform(0, 1)$. Otherwise, the system stays in the same state $X_1^{(t)}$ and a new sample is drawn to be added to the Markov chain.

Accepting the new state, although its probability is lower than the previous state, makes the sampler mix well. If the new state is not accepted in either case, then it is rejected. New states are suggested incrementally, until the distribution from which the new values are sampled converges to the target distribution.

One advantage of using the Metropolis-Hastings algorithm is that any integration that comes within a normalisation constant disappears due to the proportion of the probabilities of the two states. Therefore, the algorithm is convenient in problems where it is computationally expensive to calculate a normalisation constant through an integration.

The proposal distribution should be chosen to ensure that the Metropolis-Hastings algorithm produces an ergodic Markov chain. The ergodicity of a Markov chain assures that it converges to a stationary distribution after a number of iterations. An ergodic chain must be aperiodic and irreducible. A state in a Markov chain is called aperiodic if the greatest common divisor of return times to

the state is 1. If all the states in a Markov chain are aperiodic, the chain is called aperiodic. Irreducibility of a chain means that in any state it must be possible to reach any other state within a limited number of moves.

2.3.8.1.2 Gibbs Sampling

Gibbs sampling is a special case of Metropolis-Hastings algorithm. In contrast to the Metropolis-Hastings algorithm, Gibbs sampling accepts every new state to reach an equilibrium state. Every new sample in the Gibbs sampling is drawn from the distribution of the sample conditioned on the rest of the parameters or random variables of interest. Let $X = \{X_1^{(t-1)}, X_2^{(t-1)}, X_3^{(t-1)}, \ldots, X_n^{(t-1)}\}$ be a set of parameters that needs to be estimated through Gibbs sampling. The new value of each parameter is drawn from the conditional distribution on the rest of the parameters, such that:

$$\begin{aligned} X_1^{(t)} &\sim p(X_1^{(t)}|X_2^{(t-1)}, X_3^{(t-1)}, X_4^{(t-1)}, \ldots, X_n^{(t-1)}) \\ X_2^{(t)} &\sim p(X_2^{(t)}|X_1^{(t)}, X_3^{(t-1)}, X_4^{(t-1)}, \ldots, X_n^{(t-1)}) \\ X_3^{(t)} &\sim p(X_3^{(t)}|X_1^{(t)}, X_2^{(t)}, X_4^{(t-1)}, \ldots, X_n^{(t-1)}) \\ X_n^{(t)} &\sim p(X_n^{(t)}|X_1^{(t)}, X_2^{(t)}, X_4^{(t)}, \ldots, X_{n-1}^{(t)}) \end{aligned} \tag{2.32}$$

Until the joint distribution of the parameters $p(X_1, X_2, \ldots, X_n)$ converges to an equilibrium distribution, Gibbs sampling continues to sample new values for the parameters. The reader should note that in Gibbs sampling only one change can be applied at a time. Therefore, in the example given above, only one parameter's value can be updated at a time. There are other types of sampling algorithms, such as block sampling, where a set of parameters can be sampled together.

A difference between the Metropolis-Hastings algorithm and Gibbs sampling is the need for a normalisation. As mentioned above, in the Metropolis-Hastings algorithm, normalisations can be ignored due to the division operation. However, Gibbs sampling requires a normalisation, to draw from a conditional distribution which must be normalised beforehand.

2.4 Conclusion

In this chapter, essential background knowledge is presented to be referred throughout the the thesis. The background knowledge is presented in two main sections.

The first section presents some fundamental linguistic knowledge. As the thesis mainly focuses on morphology and syntax, a general overview of the two fields is given from the linguistic perspective. The overview consists of the basic terms and their definitions. In addition, the relationship between morphology and syntax, as well as the connection between morphology and other linguistic fields is given.

The second section presents some statistical machine learning methods which are either exploited in the presented works in this thesis, or used for morphology learning and POS tagging in the field frequently. The machine learning background consists of most prominent parameter estimation methods. Following the descriptions of MLE and MAP estimation, the Bayesian modelling perspective is presented along with non-parametric Bayesian modelling, where most prominent examples of non-parametric Bayesian modelling are described, i.e. the Dirichlet Process, Dirichlet Process mixture models, and the Chinese restaurant process. In addition to the estimation methods, it is also discussed about how to perform inference from a model without estimating the parameters, but integrating out the parameters.

CHAPTER 3

A Literature Review of Unsupervised Morphology Learning and POS Tagging

"The only source of knowledge is experience."
Albert Einstein

3.1 Introduction

Morphology learning and POS tagging are two longstanding areas in natural language processing. This chapter presents previous research on unsupervised learning of morphology and POS tags. The literature review consists of only unsupervised approaches since the scope of this thesis consists of only unsupervised learning.

3.2 A Literature Review of Unsupervised Morphology Learning

In this section, previous research on unsupervised morphology learning is presented by classifying the approaches according to the type of the mathematical model used. Mathematical models are used to define beliefs about any particular problem mathematically. Defining the beliefs mathematically, we can test how a system will react and what outputs it will produce in terms of mathematical equations. In morphology learning, a mathematical model produces the morphological analysis of a word as an output of a series of mathematical calculations.

Mathematical models can be classified using different criteria. However, in this thesis, the classification will be made according to the predictability of the model, classifying models into: deterministic models or stochastic models.

3.2.1 Deterministic Models

Deterministic models define variables in a deterministic fashion, where no randomness is engaged. The deterministic models used for morphological segmentation in the literature, are categorised into two main approaches: Letter successor variety models and information theoretic models.

3.2.1.1 Letter Successor Variety (LSV) Models

Harris (1955) is the source of the inspiration for the contributions in this category. Harris introduces the distributional characteristics of letters within a word. He identifies morpheme boundaries using letter successor counts following each letter. If the number of the successor types increase significantly at a position within an utterance, then a new morpheme boundary is introduced.

Example 3.2.1. Within the utterance *talked*, the number of letter successors is to be low until the letter *k*, however, this number is to rise after the letter *k* because of the number of possible morphemes that could follow the word (i.e. *-ed*, *-s*, *-ing* etc).

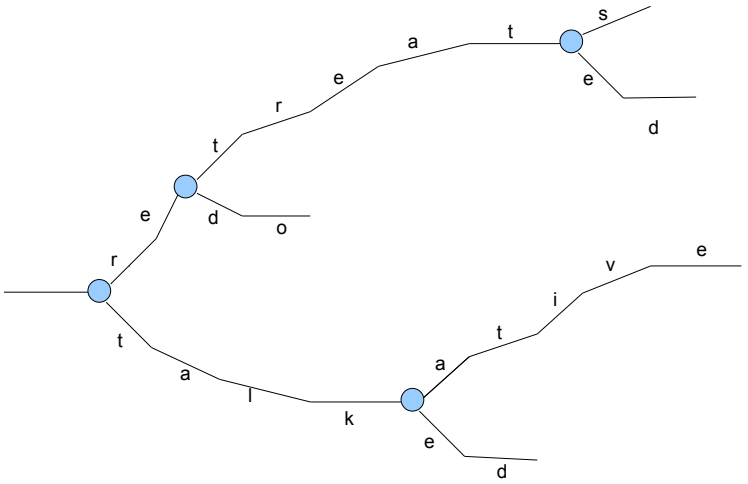


Figure 3.1: Word split points in a LSV model

If words are inserted into a tree, branches occur in the nodes that reserve potential split points. An example is given in Fig. 3.1. In the example, *re-* is a potential prefix whereas *-s*, *-ed* and *-ing* are potential suffixes in the tree.

Hafer & Weiss (1974) follow the same idea using statistical properties of successor and predecessor letter successor varieties to define a segmentation point in a word. As well as using a cutoff value to decide where to define a segmentation point like Harris, Hafer & Weiss (1974) improve the original idea by also using the entropy of the successors and predecessors analogously to the counts.

Déjean (1998) uses a morpheme dictionary that consists of the most frequent morphemes in a corpus, identified by letter successor properties. The segmentation process is performed by a morpheme dictionary.

Bordag (2005) uses the same approach by incorporating letter successor varieties with context information to eliminate noise. Bordag employs context similarity to analyse similar words together. This eliminates a significant amount of

noise.

Example 3.2.2. If the words *early* and *clearly* are analysed together, the number of different letters preceding the word *early* increases because of the word *clearly*. However, these two words are contextually distinct and need to be analysed independently.

Bordag (2006) places letter successor varieties on a *trie*[1] classifier to generalise the results for novel words. He inserts analysed words on a Patricia trie (Morrison 1968) with the frequencies of the morphemes. If a novel word is to be analysed, the *trie* is searched from the root until the correct branch in the trie is found which gives a split for the word. Using *tries* helps to handle exceptions as well. For example, a trie with the words *clear+ly*, *strong+ly* and *early* can classify hundreds of words ending with *-ly*, but still remembers one exception which is *early*.

Bordag (2008) improves the previous approach to analyse a concatenative morphology. To this end, the existing model is combined with a compound identifier that finds the compounds in a language such as German.

3.2.1.2 Information Theoretic Models

Among information theoretic models, only method used, to our knowledge, is the Minimum Description Length (MDL) principle for morphology learning. Although the principle can also be defined in a probabilistic framework, it will be introduced as an information theoretic model, due to its inspiration from information theory.

The model is extensively referred to in statistical models. The length of a probability measure $p(x)$ in bits is the negative binary logarithm $-\log_2 p(x)$ of the measure. The Bayesian rule can easily be applied for the MDL principle.

[1] A *trie* is a sophisticated tree structure where strings are usually stored. The word comes from another word $re(trie)val$. Searching tries is efficient since it allows searching through prefixes and suffixes.

To maximise the posterior probability of a model θ given data x is equivalent to minimising the description length of the model:

$$\begin{aligned} \arg\min_{\theta}[-\log_2 p(\theta|x)] &= \frac{-\log_2 [p(x|\theta)p(\theta)]}{-\log_2 [p(x)]} \\ &\propto -\log_2 [p(x|\theta)p(\theta)] \end{aligned}$$

Brent et al. (1995) encode a list of words using binary sequences as a combination of stems and suffixes, where the stems and suffixes are kept in tables. The morphological analysis of the lexicon which yields the minimum code is chosen to be the final segmentation. In another approach, in addition to stem and suffix codes, another set of codes using syntactic categories are employed. Similarly, the size of the coded lexicon is minimised.

Linguistica (Goldsmith 2001, 2006) is one of the state of the art systems in unsupervised morphology learning. Goldsmith introduces the morphological structures *signature*s to encode the data. A *signature* represents the inner structure of a list of words that have similar inflective morphology. The morphology of a corpus is represented in three lists: an affix list, a stem list, and a signature list (see Figure 3.2). The affix and the stem list contain the letters, whereas the signature list only contains pointers to stems and affixes[2]. The aim is to find the morphology that will analyse the corpus in its most compact state. This postulates finding the minimum description length of the corpus and the morphology:

$$\begin{aligned} DescriptionLength(Corpus, Model) &= length(Model) + \\ &\quad [-\log_2 (p(Corpus|Model))] \quad (3.1) \end{aligned}$$

Here, the same notation as Goldsmith is used, so as not to cause any conflict between two descriptions. Stems will be represented by t, affixes will be

[2]Goldsmith explains the reason for using the pointers (in the signatures) because of the less space requirement of the pointers in comparison to the letters. Another reason explained by Goldsmith is that the compactness of the signatures is the issue which will define a compact corpus (Goldsmith 2001).

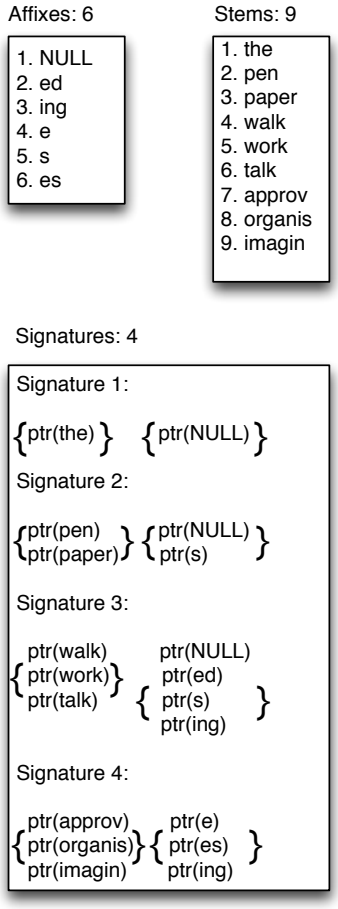


Figure 3.2: A sample morphology from Linguistica, that can generate the words: *the, pen, pens, paper, papers, walk, walked walking, walks, work, worked, working, works, talk, talked, talking, talks, approve, approves, approved, organise, organises, organised, imagine, imagines, imagined.*

represented by f, and signatures by σ. The corpus will be notated by W. As the generation of a word requires a signature, an affix, and a stem; Goldsmith defines the probability of each word using the probability of its signature $\times$ the probability of its stem given the signature $\times$ the probability of its affix given the signature:

$$p(w = t + f) = p(\sigma)p(t|\sigma)p(f|\sigma) \tag{3.2}$$

With this definition, the size of a word becomes the sum of the size of the pointer to its signature, stem, and affix. Goldsmith uses the information theoretic measure to calculate the size of a pointer, which is the inverse logarithm of a probabilistic value (Ming Li 1997). Goldsmith determines the probabilities with the relative frequencies in the corpus. Therefore, the length of a pointer to a stem becomes:

$$\langle ptr(s) \rangle = -\log_2 \frac{[s]}{[W]} \tag{3.3}$$

where $[s]$ denotes the number of occurrences of a stem s, and $[W]$ is the length of a corpus. This gives the length of a pointer to a stem in bits. The length of a pointer to an affix and a signature are calculated analogously. The length of a corpus is determined over all words in the corpus. The length of a morphology is the sum of the lengths of all lists, plus information about how many items each list consists of. The length of a stem in a list is determined by the letters that it comprises. The length of a letter is $\log_2 26$ for a language with 26 letters. Therefore, the length of a stem is estimated by:

$$\langle s \rangle = \lambda(\langle T \rangle) + \sum_{t \in Stems} len(t) \tag{3.4}$$

where $len(t)$ is the summation of the lengths of letters that the stem t consists of and $\lambda(\langle T \rangle)$ denotes how many items the stem list consists of. The length of a list of affixes and signatures are calculated analogously.

Goldsmith also defines a recursive morphology, in which complex stems are employed. To this end, a flag for each stem is placed in the stem list to determine whether the stem is a simple stem or a complex stem with a triple pointer to a signature, stem, and affix. This modification in the definition of a stem enables the analysis of words such as *[organis-ation]-s* where the stem *organis-ation* is decoded as a complex stem that consists of a pointer to a signature which includes the stem *organis* and the affix *-ation*.

Goldsmith's morphology definition employs some heuristics for the initial analysis of a corpus. Given the initial analysis of the corpus, the description length of the corpus and the model is minimised.

Morfessor (Creutz & Lagus 2002) is another state of the art model in the field. The model has several versions that adopt different methods. Morfessor Baseline (Creutz & Lagus 2002) engages the MDL principle to minimise the length of a *codebook*[3] and a corpus. Using a cost function, the compact representation of a corpus is inferred through:

$$\begin{aligned} Cost &= Cost(Corpus) + Cost(Codebook) \\ &= \sum_{i \in D} -\log p(m_i) + \sum_{j \in M} k \times l(m_j) \end{aligned}$$

where m_i denotes the morphemes. Here the corpus is generated by morphemes in the *codebook*. The length of a corpus is computed using the maximum likelihoods of the morphemes, whereas the length of a *codebook* is the summation over all morphemes' lengths $l(m_j)$ where k is the number of bits to encode each letter. Morfessor Baseline deploys a recursive segmentation where each discovered morpheme is analysed recursively as long as it improves the cost. Morfessor Baseline does not make use of signatures, instead a *codebook* is used for all morphemes that are used to generate the entire corpus.

Remark. The other members of the Morfessor family are not described here as they diverge from the type of the model adopted. Morfessor ML (Creutz & Lagus 2004) and Morfessor MAP (Creutz & Lagus 2005a) are presented in Section 3.2.2.1, and another member of the Morfessor family that employs the prior probabilities of the morpheme lengths and morpheme frequencies is presented in Section 3.2.2.2, as a statistical approach.

Argamon et al. (2004) introduce a novel recursive algorithm using the MDL principle to segment the words recursively by suggesting multiple split points in a word.

[3] A *codebook* consists of the morphemes that will generate a corpus.

Kazakov & Manandhar (Kazakov 1997; Kazakov & Manandhar 2001) develop a hybrid approach. The hybrid approach is a combination of genetic algorithms and inductive logic programming (ILP). A MDL bias is introduced to genetic algorithms, as a fitness function. First, an initial segmentation is performed by the genetic algorithms. Subsequently, by using the initial segmentation, rules are learned by a decision list to generalise the rules for novel words. In a later approach, the authors also engage semantic similarity.

3.2.1.3 Other Approaches

There have been other attempts in the field, which use techniques which do not belong in the categories reviewed earlier.

Neuvel & Fulop (2002) propose an algorithm based on the word-based theory of morphology (Ford et al. 1967). In this approach, instead of inducing the morphemes, morphological relations between the words are defined to learn new word forms.

Keshava & Pitler (2006) derives a new algorithm called RePortS which builds trees with the letters of the lexicon. A forward tree is used for the suffixes, whereas a backward tree is used for the prefixes. Using trees, Keshava & Pitler (2006) defines some criteria based on the strings' conditional probabilities on the tree, to identify the suffixes and prefixes by giving them scores. Finally, words are segmented using these scores.

Demberg (2007) improves the original RePortS algorithm for a complex morphology, by adding an extra step to the algorithm, to find a candidate stem list.

Lavallée & Langlais (2009) use formal analogies to find the relation between 4 word forms, such as $\{$*walking*, *speaker*, *walks*, *speaks*$\}$. However, due to the complexity of the searching analogies, it is considered to be impractical for large lexicons.

Monson et al. (2008) follow a paradigmatic approach that discovers candidate suffixes and stems to build paradigms. In their approach, candidate suffixes are any final substrings of words that are found iteratively through a corpus. Once partial paradigms are built, they are merged by clustering. Finally, words are segmented by stripping off suffixes that occur in paradigms.

Remark. Monson et al. (2008) derive the probabilistic version of the approach, which will be reviewed in the following section.

Last but not least, **Lignos et al. (2009)** employ Base and Transforms model (Chan 2008) that is based on the discovery of the base and derived forms of words. The discovery is performed through transforms, which are orthographic modifications that are applied on a word to derive another form of the same word. A transform given by (s_1, s_2) removes the suffix s_1 from the word and adds another suffix s_2 to derive another form of the word. **Lignos (2010)** extends the previous version by introducing base inference model which learns the base forms when the base form of a word does not exist in the corpus. The new model handles also compounding during learning by decomposing a word into its component words which yield the highest geometric mean of the component frequencies, whereas **Lignos et al. (2009)** handle compounding as a post-processing.

3.2.2 Stochastic Models

Stochastic modelling has randomness while defining the data, on the contrary to deterministic modelling. Statistical modelling also has randomness in the production of outputs. In contrast with deterministic approaches, models in this category suggest stochastic solutions to the problems. In the literature, significant attention is given to statistical modelling. Statistical models that are used for morphology learning will be classified into four groups: probabilistic frameworks, generative models, log-linear models and contrastive estimation. Although in some categories, there is only one attempt in terms of morphology learning, I find it noteworthy to classify them in a separate category.

3.2.2.1 Probabilistic Frameworks

There is a remarkable amount of work using a probabilistic framework for morphology learning.

Creutz & Lagus (2002) propose another baseline member of the Morfessor family that employs Maximum-Likelihood (ML) estimation rather than a MDL criterion. The model is optimised by Expectation Maximisation (EM). Initially words are split randomly. Iteratively, words are split by drawing a morpheme length from a Poisson distribution. Splits are either accepted or rejected according to the rejection criteria which has two conditions; rare morphemes and one letter morphemes are rejected. This model is similar to the MDL version of the baseline model (Creutz & Lagus 2002) that is mentioned in Section 3.2.1.2. The difference being the ML version does not use any prior probability for the models.

Creutz & Lagus (2004) propose Morfessor Categories ML. In this version of Morfessor, again ML estimation is adopted. Moreover, differently from the baseline ML model, a first order Markov chain is used to assign probabilities to each possible split of a word form. In the model, each segmented morph belongs to one of these categories: prefix, suffix or stem. Within a bigram model, the probability of a segmentation of a word w into the morphemes $m_1, m_2, \ldots, m_k$ is computed as follows:

$$p(m_1, m_2, \ldots, m_k|w) = [\prod_{i=1}^{k} p(C_i|C_{i-1})p(m_i|C_i)]p(C_{k+1}|C_k) \tag{3.5}$$

where $p(C_i|C_{i-1})$ is the transition probability from one category to another. Once the category is selected, a morpheme is emitted from the selected category with the probability $p(m_i|C_i)$. To learn the probabilities in the model, words are initially segmented by applying the Morfessor Baseline Creutz (2003) (see Section 3.2.2.2 for further description). Once the words are segmented initially, initial category membership probabilities $p(C_i|m_i)$ are estimated by using the

perplexity measure. The perplexity measure expresses the predictability of the preceding and following words of a given word. The EM algorithm is used to estimate the probabilities in each iteration after re-tagging the words, by using the Viterbi algorithm until the probabilities converge. This work proves that the dependencies between the morphemes are crucial in morphology learning.

Creutz & Lagus (2005a) develop Morfessor Categories MAP (maximum a posteriori) introducing prior information for the lexicon. In this approach, each morpheme is defined using two parameters: meaning and form. The form of a morpheme refers to the substructure of the morpheme (made of a string of letters or by two submorphemes) and the meaning of a morpheme consists of the length, frequency and perplexity of the morpheme as defined in the previous members of the Morfessor family (Creutz 2003; Creutz & Lagus 2004).

Christian Monson (2009) extend ParaMor (Monson et al. 2008) by assigning a likelihood for each morpheme boundary before applying the segmentation using paradigms. In order to assign a likelihood for each morpheme boundary, a tagger is trained through the segmentation results of the baseline ParaMor (Monson et al. 2008). The tagger tags each split point within a word as a morpheme boundary if the split point corresponds to a morpheme boundary or the split point is tagged as the continuation of a morpheme, which means that it is not a morpheme boundary. The probabilistic ParaMor has a higher accuracy compared to the baseline ParaMor. Moreover, the authors combine the results of the baseline ParaMor with Morfessor (Creutz 2006) to train the tagger in another experiment.

There are other contributions using a probabilistic framework. Some of these approaches use a nonparametric Bayesian framework:

Goldwater et al. (2006) introduce a two stage model in which initially words are generated by a generator component and then the frequencies of the words are estimated by an adaptor, to create a power-law distribution. The adaptor runs a Pitman-Yor process by locating the words in tables in a rich-get-richer fashion.

Snyder & Barzilay (2008) develop another non-parametric Bayesian model that makes use of bilingual parallel corpora to induce frequently occurring morphemes (*abstract* morphemes) within parallel short phrases, instead of inducing the morphemes in each language individually. The model is a hierarchical Bayesian model where the defined distributions are drawn from Dirichlet processes. Although it has only been tested on bilingual corpora, the model can also be extended to induce morphemes across multiple languages.

3.2.2.2 Generative Probabilistic Models

Differently from a discriminative model, a generative model defines a joint probability distribution between variables and data. It is based on intuitions about how the data is generated. However, a discriminative model turns to a conditional probability distribution of a group of target variables, conditioned on the observed data. For the problem of morphological segmentation, a typical generative model defines a joint probability distribution between the possible segmentations of the corpus and the corpus itself.

Creutz (2003) proposes another member of the Morfessor family adopting a generative probabilistic model. This version of the Morfessor is proposed to overcome the over-segmentation problem in the Baseline Morfessor. The proposed model uses prior information about morpheme lengths and morpheme frequencies, within a generative probabilistic model framework. The model is based on the probabilistic model by Brent (1999). The generative story begins by determining the number of morphemes in the lexicon according to a uniform distribution. Morpheme lengths are then drawn from a gamma distribution, and each morpheme is formed by letters. Letter probabilities are the maximum likelihoods of each letter in the corpus. Finally, morpheme frequencies are defined by Mandelbrot's correction of Zipf's formula (see Baayen (2001)). Once the lexicon has been generated, the corpus is generated with the morphemes. The optimal model is searched following a similar recursive search algorithm which is used in the Baseline Morfessor (Creutz & Lagus 2002). Results show that the usage of prior information increases the accuracy of the algorithm.

Chan (2006) suggests an algorithm based on the Latent Dirichlet Allocation (LDA) which is also a generative probabilistic model, where the collections of the data are generated through a three-level hierarchical Bayesian model (Blei et al. 2003). When it is applied to topic discovery, the three levels consist of documents, topics and a vocabulary. Chan applies a similar approach by replacing the data collections with the suffixes, stems and paradigms, where the latent classes are the paradigms to be induced.

3.2.2.3 Log-Linear Models

In contrast to the directed generative models such as HMMs, log-linear models (also known as maximum entropy models, or exponential models) make use of the dependent features in the data. Log-linear models were first used for several supervised tasks (for machine translation by Och & Ney (2002), for sentence boundary detection by Reynar & Ratnaparkhi (1997), parsing by Johnson et al. (1999) and Johnson (2001), etc). They have also been used for several unsupervised tasks (POS tagging by Smith & Eisner (2005), coreference resolution by Poon & Domingos (2008)). However, Poon et al. (2009) is the primary work that uses log-linear models for unsupervised morphological segmentation; their model observes morphemes and their contexts as dependent features.

Poon et al. (2009) develop a log-linear model where the joint probability between the corpus and all possible segmentations is defined. Since it is not possible to derive all the pairs belonging to this joint probability, a normalisation constant Z is estimated to normalise the joint probability. A few techniques are suggested earlier to compute the normalisation constant. Smith & Eisner (2005) apply contrastive estimation by searching around the neighbourhood of the data, whereas Rosenfeld (1997) and Poon & Domingos (2008) use sampling to compute the normalisation constant.

Poon et al. (2009) use both contrastive estimation and sampling to compute the normalisation constant. The neighbourhood is searched by transposing pairs

of letters to create invalid words. Gibbs sampling is used to find the optimum segmentation. In the model, also a prior information that is inspired by the MDL model which controls the number of morpheme types in the lexicon and the morpheme tokens in the corpus is used.

3.2.3 Evaluation of Morphology Segmentation Algorithms

The evaluation of morphological segmentation requires a gold standard to compare with the suggested analyses, as do most natural language processing tasks. However, the evaluation process is not as complicated as the evaluation of other NLP tasks such as POS tagging (see 3.3.4). The reason is that the results in a morphological segmentation consist of only the split points of the words that match to a gold standard. However, in addition to matching the split points to a gold standard, any identified feature (such as ambiguity, morphophonology etc.) should also be rewarded. Another difficulty comes with obtaining a gold standard; morphology learning is a troublesome task itself that requires many issues to be handled; such as ambiguity of words, morphological complexity of languages, stem changes etc. When all these issues are considered, it is noticeable that obtaining a gold standard is a demanding task itself.

Spiegler & Monson (2010) define the features of a good evaluation metric as:

- Correlating well with other NLP tasks.
- Being computationally easy.
- Being robust.
- Being informative about the strengths and weaknesses of the system.
- Being able to account for the linguistic structure of the language, such as morphophonology, allomorphy, syncretism, and ambiguity.

The evaluation methods for morphological segmentation can be investigated using two categories: methods based on a comparison with a gold standard and methods based on embedding of the segmentation results in other NLP tasks to evaluate how the segmentations improve the performance of the task.

3.2.3.1 Evaluation Using a Gold Standard Segmentation

For morphological segmentation, precision, recall and f-score are predominantly used as evaluation scores, like most machine learning tasks. Precision evaluates how many of the suggested split points match up with the gold standard split points, whereas recall evaluates how many of the split points in the gold standard are suggested by the system. In other words, precision states the validity of the morphemes suggested and recall states whether the desired morphemes are found. F-score combines the two scores which is usually the harmonic mean of precision and recall:

$$F\text{-}score = \frac{1}{1/Precision + 1/Recall} \tag{3.6}$$

For evaluations of some work, a full gold standard that consists of a segmentation of all words is used (Goldwater et al. 2006; Poon et al. 2009), whereas in some work, only a part of the output words are used to do a random comparison with the gold standard segmentations to produce a generalised score. For the former evaluation method, either a highly accurate morphological analyser is used (for Arabic such as Habash & Rambow (2005), or some heuristics are used for the construction of a gold standard (for English, see Goldwater et al. (2006). Morpho Challenge (Kurimo et al. 2010) utilises the latter evaluation method, where only a small set of gold standard words are used to evaluate the resulting segmentation. In the gold standard, words are given with their segmentations which consist of morpheme labels and morphemes, such as:

ablatives ablative:ablative_A s:+PL
abounded abound:abound_V ed:+PAST
carriages carri:carry_V age:age_s s:+PL
detraction detract:detract_from_V ion:ion_s
entitling entitl:entitle_V ing:+PCP1

Here morpheme labels represent some information about the word forms; i.e. plural, past tense form, participle etc. To measure precision, a group of words are sampled from the resulting word list. For each morpheme in the list, another

word is found that includes the same morpheme. This will create a word pair list. Finally, word pairs are checked in the gold standard to see whether the pairs share a common morpheme. For each true guess, one point is given. The score is computed by dividing the total number of received points by the number of sampled words. Recall is measured analogously to precision, where the word pairs are sampled from the gold standard, and comparisons are made through the resulting segmentations.

Remark. The gold standard datasets and the evaluation method described in (Kurimo et al. 2010) are used to evaluate the models described in Chapter 4 and Chapter 5.

Apart from these evaluation metrics, Spiegler & Monson (2010) propose a novel evaluation metric called *EMMA* which does not perform a one-to-one comparison with the gold standard data, but instead finds the maximum matching between the suggested segmentations and the gold standard segmentations through an optimal maximum matching (in a bipartite graph) which is based on graph theory.

3.2.3.2 Evaluation through Other Tasks

Another way of evaluating the results of a morphological segmentation is to embed the suggested segmentations into a real world NLP task which utilises the analysed words. In addition to the traditional evaluation metric which is described earlier, Morpho Challenge (Kurimo et al. 2011b) performs information retrieval and machine translation tasks. In both tasks, words are replaced with the word segmentations. In information retrieval, queries are replaced with their segmentations, whereas in machine translation, task the source language is replaced with its segmentations. Finally, the tasks are evaluated using average precision and *BLEU*[4] score respectively.

[4]BLEU (bilingual evaluation understudy) score is an improved version of precision that can account for multiple translations.

3.3 A Literature Review of Unsupervised POS Tagging

The previous work on unsupervised part-of-speech (POS) tagging can be classified into two main categories regarding the methods been used: clustering and Hidden Markov Models (HMMs). Some work on unsupervised POS tagging considers tagging as a clustering problem and clustering algorithms are used to group the words into syntactic categories. From a similar perspective, some works consider the task as a sequence labelling problem.

Principally, Harris's distributional hypothesis (Harris 1955) has a great influence on most of the approaches:

"Words of similar parts of speech can be observed in the same syntactic contexts."

Harris points out the importance of the contextual similarity of words that have similar linguistic roles, such as nouns, adjectives etc.

Lamb (1961) pioneers in syntactic category induction originating from ideas on the distributional hypothesis. He makes use of the left and right neighbours of the words to measure a *token-neighbour* (token-left neighbour, token-right neighbour) ratio to construct horizontal and vertical groupings (H-groups and V-groups).

3.3.1 Clustering

A group of approaches consider POS tagging as a clustering problem, where the words are clustered into syntactic categories that each represents a POS tag.

Brown et al. (1992) employs an information theoretic approach where the word clusters yielding the greatest average mutual information between adjacent classes are discovered. To this end, initially each word is assigned to a separate cluster. Then the cluster pair which yields the minimum loss in the average

mutual information is merged. The process is repeated until a set of clusters is found. Finally, each word is replaced into another cluster, if the resulting cluster leads to a greater average mutual information. The algorithm terminates if no more moves are possible, which leads to greater average mutual information.

Some of the earlier work represents the words in terms of their context vectors, where the words in the neighbourhood are used to measure the similarity between words. To this end, vector space models are widely used to represent statistics regarding the contexts of the words.

Finch & Chater (1992) consider the two preceding and the two following words that are in the most frequent 150 words as the context. To measure the linguistic similarity between context vectors, a Spearman Rank Correlation Coefficient is used. Using the similarity measure, hierarchical agglomerative clustering is performed to capture the linguistic categories in a hierarchical structure.

Schütze (1993) uses context vectors that keep the counts of the context words in a variable size of window. Because of the unfeasibility of such large vectors, Singular Value Decomposition (SVD) (see Deerwester et al. (1990)) is used to reduce the dimensionality in the concatenated context vectors. In the reduced space, nearest neighbours are induced to form individual clusters by Buckshot clustering (Cutting et al. 1992). Schütze (1993) also uses neural networks to cluster ambiguous words which are poorly clustered by the Buckshot clustering.

Schütze (1995) improves the previous work also using the contexts of the context words, in addition to the context words itself. Another difference in this approach is that the context vectors are used separately instead of being combined in to a single context vector.

Clark (2000) follows the same distributional hypothesis within a distributional clustering algorithm. Differently from the others, he defines the contexts probabilistically where each word defines a probability distribution over all possible contexts. Instead of using context words, the clusters of the context words

are used to eliminate the sparseness problem. Kullback-Leibler (KL) divergence is used to measure the divergence between the clusters, to decide which merges will be appropriate in each step.

Remark: We adopt the clustering algorithm in Clark (2000) for morphological segmentation (see Chapter 4). A detailed explanation of the algorithm is given in the chapter concerned.

Freitag (2004) employes an information theoretic co-clustering algorithm (Dhillon et al. 2003) to induce the POS tags of the words. The algorithm makes use of both words and their contexts in a similar fashion to the other approaches given in this section. Words and their contexts are replaced in the clusters to find the clusters which will maximise the mutual information between the words and the contexts in a particular cluster. Freitag also developes a Hidden Markov Model (HMM) tagger (see Section 3.3.2) to tag low frequency words.

Biemann (2006b) employes a graph based clustering algorithm to induce POS tags. One advantage of the graph based clustering algorithms is that the number of clusters does not need to be initially defined. In a graph clustering algorithm, the number of clusters is discovered while the graph is formed. Biemann uses two graphs; one for high frequency words where there is sufficient contextual information and one for medium and low frequency words where only likelihood statistics are been used. In his approach, to assign the classes, he uses the Chinese Whispers (CW) graph-clustering algorithm (see Biemann et al. (2007) for a more detailed definition of the algorithm and its application to natural language). A graph is constructed for the high frequency words by using the context similarity of the words to draw an edge between two words. A threshold is used which employs the cosine similarity of the words. Another graph is constructed by using the log-likelihoods and the number of common neighbours shared between the words. Both graphs are partitioned by the CW algorithm which produces some syntactic categories. However, to enlarge the dataset for tagging, Biemann defines a trigram model in which the joint probability of the tags and the words are maximised in a corpus.

3.3.2 Hidden Markov Models

Rather than using clustering methods to group words into syntactic categories, this set of studies uses hidden Markov models (HMMs). These approaches employ either dependencies between words or contextual information similar to the clustering approaches given earlier. The difference is that these approaches investigate the POS tagging problem as a formulation of a Markov chain. Markov chains are greatly used as statistical models. A Markov chain defines a random process which consists of random variables $X = \{X_0, X_1, X_3, \dots\}$ with discrete values $s_0, s_1, s_3, \dots$ where the following state depends only on the current state, but not on previous states visited:

$$p(X_{t+1} = s|X_0 = s_0, X_1 = s_1, \dots, X_t = s_t) = p(X_{t+1} = s|X_t = s_t) \quad (3.7)$$

It is also possible to define a history of size m for each state where the chain becomes a Markov chain of order m:

$$\begin{aligned} p(X_t = s_t|X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \dots, X_1 = s_1) \quad = \\ p(X_t = s_t|X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \dots, X_{t-m} = s_{t-m}) \end{aligned} \quad (3.8)$$

Markov chains are the basis for HMMs. An HMM is a Markov chain whose states are hidden, and states can only be observed through the observations which are emitted from each state. A sample HMM is given in Figure 3.3. The given HMM involves states $X = \{X_1, X_2, X_3, X_4\}$. There is a probability assigned to each transition from one state to the following one. These probabilities are called transition probabilities. Transition probabilities are given as $a = \{a_1, a_2, a_3\}$ for the sample HMM. There is also a list of observations which are emitted from each state, such that $Y = \{Y_1, Y_2, Y_3, Y_4\}$. Each observation is obtained with an emission probability assigned to each state to produce that observation. These probabilities are called emission probabilities and given as $b = \{b_1, b_2, b_3, b_4\}$ on the figure.

In POS tagging, the states of the HMM become the possible tags. For a

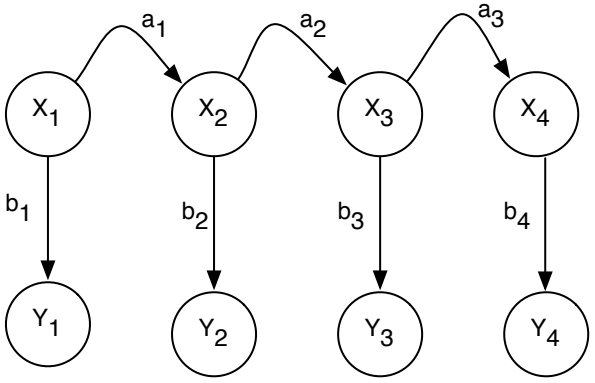


Figure 3.3: An HMM

sequence of states, the most probable sequence of tags is produced by the HMM. The ultimate goal of an HMM is to find the model that maximises the probability of a given text. The text is stated as a sequence of states and words are emitted from these states.

Merialdo (1994) employs a triclass HMM where two previous words are considered as the history (see Fig. 3.4). This is a standard trigram model for POS tagging which has influenced many studies in POS tagging. In a trigram model, each tag is only dependent on the tags of the previous two words. Therefore, the transition probability becomes:

$$p(t_i|w_{i-1}, t_{i-1}, w_{i-2}, t_{i-2}, \ldots, w_1, t_1) = p(t_i|t_{i-1}, t_{i-2}) \tag{3.9}$$

Each word is emitted from a tag with the emission probability:

$$p(w_i|w_{i-1}, t_{i-1}, w_{i-2}, t_{i-2}, \ldots, w_1, t_1) = p(w_i|t_i) \tag{3.10}$$

which only depends on the tag itself. Two different types of training are applied in Merialdo (1994), relative frequency training (RF) and maximum likelihood (ML) training. In relative frequency training, probabilities of tag trigrams and word-tag pairs are estimated based on the data that has been tagged so far;

whereas with the ML training, the model that maximises the probability of a corpus is searched. In the relative frequency training, distributions are interpolated with a uniform distribution to discard the zero probabilities. The Forward-Backward algorithm (Leonard E. Baum 1967; Jelinek 1976; Bahl et al. 1983) is used for training the HMMs, when using the ML estimation for the parameters. Once the parameters are estimated, the data is tagged by a Viterbi tagger to find the most probable sequence of tags in a text. The author shows that if a small amount of tagged data is used for training, then the ML training performs with better accuracy whereas if more tagged data is available RF training gives more accurate results.

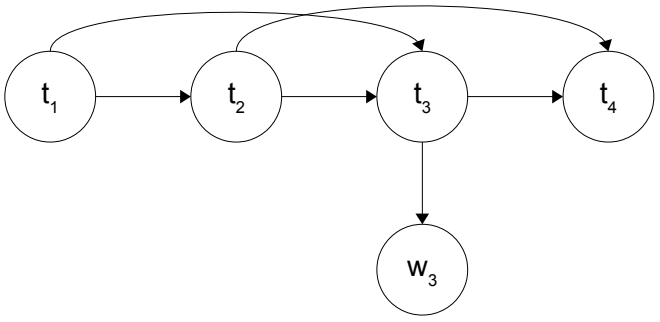


Figure 3.4: Trigram HMM tagger

Banko & Moore (2004) use a contextualised HMM tagger which also employs the part-of-speech tags of the previous and following words to tag the current word. This helps to disambiguate the tag of a word that can have multiple tags (see Figure 3.5).

In a contextualised HMM tagger, the transition and emission probabilities of the contextualised HMM tagger become:

$$p(t_i|w_{i-1}, t_{i-1}, w_{i-2}, t_{i-2}, \ldots, w_1, t_1) = p(t_i|t_{i-1}, t_{i-2}) \tag{3.11}$$

Each word is emitted from its tag with the emission probability:

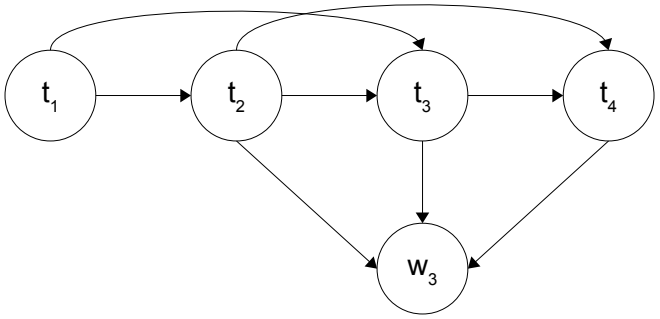


Figure 3.5: Contextualised HMM tagger

$$p(w_i|w_{i-1}, t_{i-1}, w_{i-2}, t_{i-2}, \ldots, w_1, t_1) = p(w_i|t_{i-1}, t_i, t_{i+1}) \tag{3.12}$$

Banko & Moore (2004) suggest constructing a filtered lexicon which does not include noisy part-of-speech assignments. In addition to these, a slightly different training approach is employed by the authors. In the standard HMM taggers, the transition and emission probabilities are estimated simultaneously, however, in the suggested training algorithm, this procedure has been made sequential, first the transition probabilities are estimated, while the emission probabilities remain fixed, then the emission probabilities are estimated keeping the transition probabilities fixed.

Kupiec (1992) proposes a trigram HMM tagger which employs category classes rather than considering words individually. The common words are still represented individually, but in this new approach, the rest of the words are defined in equivalence classes that consist of words which have the same part-of-speech. His experiments indicate that using word classes contributes to the robustness of the tagger.

Kupiec (1992), Merialdo (1994), and Banko & Moore (2004) focus on contextual dependencies to increase the accuracy of the suggested taggers, whereas other researchers (Church 1988; Cutting et al. 1992; Smith & Eisner 2005) focus

on parameter estimation techniques.

3.3.3 Other Approaches

While most algorithms use EM for the estimation in the taggers (Church 1988; Cutting et al. 1992), some researchers (Johnson 2007) show that the EM algorithm does not give a good estimation in real world problems i.e. learning language constituents. In POS tagging, the number of word tokens per tag is skewed. However, the EM algorithm assigns a similar number of word tokens to each POS tag. To overcome this problem, Bayesian approaches are used defining the prior probabilities for skewed distributions.

Johnson (2007) uses a Bayesian approach that uses a first order Markov model (also known as bitag model) where two different estimation techniques are compared: ML estimation with EM and Bayesian estimation with Gibbs sampling and variational Bayes. In ML formulation, each transition probability is generated from a discrete Multinomial distribution θ_t which defines a distribution over POS tags for an assigned number of POS tags, and each emission probability is generated from also a discrete Multinomial distribution ϕ_w which defines a distribution over all words w_i for a number of words given their POS tags.

$$\begin{aligned} t_i|t_{i-1} &\sim Multinomial(\theta_t) \\ w_i|t_i &\sim Multinomial(\phi_w) \end{aligned} \tag{3.13}$$

Johnson uses the Forward-Backward algorithm (a special case of Expectation-Maximisation (EM) algorithm) to estimate the model parameters (θ, ϕ). Experiments demonstrate that the EM algorithm converges slowly to a local maxima that needs a great number of iterations. In his second set of experiments, he derives the model in a Bayesian framework where the prior probabilities are defined for the model parameters:

$$\begin{aligned} \theta_t &\sim Dirichlet(\alpha_t) \\ \phi_t &\sim Dirichlet(\alpha_w) \\ t_i|t_{i-1} = t &\sim Multinomial(\theta_t) \\ w_i|t_i = t &\sim Multinomial(\phi_t) \end{aligned} \tag{3.14}$$

Here α_t controls the sparsity of the transition probabilities through a Dirichlet distribution, and α_w controls the sparsity of the word emission probabilities through another Dirichlet distribution. Therefore, unlike the EM estimation, the emission probabilities are not close to a uniform distribution and define a skewed distribution, which is controlled by the hyperparameter α_w. The same property is also true for the state-to-state transition probabilities, which are also skewed with the hyperparameter α_t. For the Bayesian estimation, Johnson uses Gibbs sampling and Variational Bayesian inference (Jordan et al. 1999). Variational Bayesian inference gives an approximate inference, defining bounds on the probabilities. The experiments show that Variational Bayesian performs better than the Gibbs sampling. In addition to this, the EM algorithm performs a comparably higher many-to-1 accuracy whereas it has a lower 1-to-1 accuracy (evaluation methods are discussed in Section 3.3.4).

Goldwater & Griffiths (2007) develop a second order HMM where similarly to the first order HMM of Johnson (2007), Dirichlet priors are defined for the Multinomial parameters of the transition and emission probabilities:

$$\begin{aligned} \theta_t &\sim Dirichlet(\alpha) \\ \phi_t &\sim Dirichlet(\beta) \\ t_i|t_{i-1} = t, t_{i-2} = t' &\sim Multinomial(\theta_t) \\ w_i|t_i = t &\sim Multinomial(\phi_t) \end{aligned} \tag{3.15}$$

Parameters are estimated using Gibbs sampling. The authors perform two different sets of experiments, one for the fixed values of the hyperparameters, and one with a hyperparameter inference. The experiments show that the Bayesian approach improves the performance, allowing skewed distributions for POS tags. It should be noted that although Goldwater & Griffiths (2007) use an unsupervised approach, a dictionary is engaged to determine the possible POS tags that can be emitted from each word.

It should be noted that, although Goldwater & Griffiths (2007) and Johnson (2007) use HMMs for POS tagging, because of the Bayesian approach adopted, they are investigated in a different section to make a discrimination between earlier standard HMM taggers.

In addition to the Bayesian approaches, there are also other attempts that make use of other estimation techniques.

Smith & Eisner (2005) use a novel unsupervised estimation technique called *contrastive estimation* which is an alternative to EM. The authors employ conditional random fields (Lafferty et al. 2001). In the learning of a probabilistic model, the aim is to shift the probability mass from unseen events to observed events. However, in contrastive estimation the aim is also to decide from where to take this probability mass. To decide, positive examples are used to derive negative examples. This is done by perturbing the positive examples to search the neighbourhood around them to generate negative examples. For example, to search through the neighbourhood of the syntax of a given sentence, a word can be removed, or any two adjacent words can be transposed. Both of these processes give ungrammatical sentences, which can be used as negative examples from which some of the probability mass can be shifted. The authors show that contrastive estimation gives a better accuracy than the EM algorithm.

Van Gael et al. (2009) propose an infinite HMM for POS tagging where the number of hidden states are estimated through a non-parametric Bayesian approach. In earlier HMM based models (Goldwater & Griffiths 2007; Johnson 2007), the number of POS tags is initially defined. However, it is difficult to set the number of hidden states. Van Gael et al. define a hierarchical Pitman-Yor

process, where the model is considered to be an infinite mixture model, with the components being in a hidden state. The authors employ one extra layer to define a common distribution for all hidden states, to be able to emit any word from any state. Otherwise, for each hidden state, there would be a separate output set which differs from the other states. The suggested model employs a beam sampler (which is a block Gibbs sampler) that alternately samples hyperparameters, states and outputs. The authors suggest an evaluation method which uses the produced POS tags for a real task in natural language processing, shallow parsing. Principally, using POS tagging in shallow parsing improves the accuracy of the shallow parsing. The shallow parsing results support this, with a higher accuracy with the use of POS tags. However, the performance of the model is not as high as a supervised model, but the scores are still at an acceptable level. A similar evaluation method is also used in Biemann (2006a).

Naseem et al. (2009) propose multilingual part-of-speech tagging. Multilingual learning incorporates various languages in the learning scheme. Multilingual learning of language draws attention to many natural language problems: word sense disambiguation (Diab & Resnik 2002; Bhattacharya et al. 2004), machine translation (Chen et al. 2008; Och & Ney 2001), grammar induction (Kuhn 2004), morphological segmentation (Snyder & Barzilay 2008). Incorporating different languages helps reduce the ambiguity, which is an important issue in POS tagging. For example, it is possible to disambiguate the word *can*, used as a modal verb or as a noun, by looking at the same sentence in another language. Naseem et al. (2009) propose two different types of the model: a merged node model, which combines the tag structures from an aligned pair of words in two different languages, and a latent variable model which constructs a model with latent variables that make use of monolingual tag structures through a set of latent variables shared between different languages. The first model can function for bilingual data whereas the second model can function for more than two languages. Naseem et al. (2009) evaluate the models for parallel corpora in 8 different languages: Bulgarian, Czech, English, Estonian, Hungarian, Romanian, Serbian, and Slovene.

3.3.4 Evaluation of the POS Tagging Algorithms

Evaluation of the unsupervised POS tagging is not very straightforward due to the lack of a qualified method for mapping the resulting tags to the gold standard tags. The main difficulty lies in the number of tag types that differ in the resulting hidden states and in the gold standard. Even having the same number of tag types with the gold standard, a method is required to find out which tag type correspond to which tag in the gold standard to be able to evaluate the results. Different approaches are proposed to tackle the evaluation issues in POS tagging. Some of the common ones are given below:

Many-to-1 accuracy: This type of accuracy maps each resulting tag to a gold standard tag which has the highest frequency of words assigned with the result tag. Therefore, it is possible to assign multiple result tags to each gold standard tag, while some gold standard tags might be left unassigned to any of the resulting tags. Clark (2003) argues that many-to-1 accuracy gives the highest accuracy if each word is tagged with a different POS tag. In fact, many-to-1 accuracy increases with the number of resulting POS tags.

One-to-one accuracy: This evaluation method is proposed by Haghighi & Klein (2006). The method maps each resulting tag to only one gold standard tag. Nevertheless, if the number of tag types differ in results and in gold standard, again it makes the evaluation demanding. If the number of resulting tag types is higher than the number of tag types in the gold standard, only a few of the tags will mapped to the gold standard tags, and the rest are left unassigned, therefore yielding a low accuracy.

Variation of Information (VI): Meilă (2003) proposes an information theoretic measure. Unlike the accuracy measures, VI does not aim to map the resulting tags to the gold standard tags, but instead, it measures how different two clusters are. To this end, the summation of the conditional entropies of each clustering is measured, which gives the variation of information *VI*:

$$VI(C,T) = H(C|T) + H(T|C) \tag{3.16}$$

where C is the resulting clustering and $H(C|T)$ is the conditional entropy of the resulting clustering conditioned on the gold standard clustering T, whereas $H(C|T)$ is vice versa measured. Unlike the accuracy measures, the lower the VI, the closer the resulting clustering is to the gold standard clustering. Therefore, VI becomes zero, if the two clusterings are the same. Goldwater & Griffiths (2007) use this type of measure to evaluate their Bayesian model.

Substitutable F-score: Frank et al. (2009) proposes a completely different measure which does not require a gold standard. The measure is based on the substitutability of different words that share the same *frame*. A *frame* is defined as a context which consists of a pair of words having a word between. If the words occurring in the same frame are assigned the same POS tag, the resulting tag is regarded as true. To this end, S-clusters, which consist of the frames in a corpus, are constructed, and each of them is mapped to a word that occurs in that *frame*. In addition to this, C-clusters are also created where each consists of the resulting clusters. Finally these two clusters are compared to measure the precision SP and the recall SR:

$$\begin{aligned} SP &= \frac{\sum_{s\in S}\sum_{c\in C}|s\cap c|(|s\cap c|-1)}{\sum_{c\in C}|c|(|c|-1)} \\ SR &= \frac{\sum_{s\in S}\sum_{c\in C}|s\cap c|(|s\cap c|-1)}{\sum_{s\in S}|s|(|s|-1)} \end{aligned} \tag{3.17}$$

A few of the evaluation measures that are commonly used are discussed here. It has to be noted that there are other measures to evaluate POS tagging results (cross-validation (Gao & Johnson 2008), V-measure (Rosenberg & Hirschberg 2007), V-beta (Van Gael et al. 2009)).

3.3.5 A Literature Review to Cooperative Learning of Morphology and Syntax

Morphology and syntax are two nested fields where each has an influence on the other. The relationship between morphology and syntax, although, true for morphologically complex languages, where the morphology of a word is a salient feature for its syntactic role in the word, it is also true for morphologically poor languages, where the context is also a useful feature to analyse the morphology of a word. Therefore, this relationship has been used for morphology learning in which the syntactic features are considered, and also it is used for the induction of the syntactic categories where the morphological features are considered.

Example 3.3.1. The rightmost suffix of a word is a useful feature to determine the part-of-speech of a word (Hu et al. 2005). For example, most adverbs end with the suffix *-ly*, whereas the suffix *-ed* usually belongs to a verb in the past tense in English. Therefore, knowing either the syntax, or the morphology will help discover the other one.

3.3.5.1 Morphology Learning by Using Syntax

Hu et al. (2005) improve the Linguistica (Goldsmith 2001) by adding syntactic information along with the morphology of the words within the same MDL framework. To this end, another list is inserted for the part-of-speech tags in addition to the list of stems, affixes, and signatures. Therefore, the entire description length of the model becomes the addition of the lengths of four lists. An illustration of the modified model is given in Fig.3.6.

In addition, Hu et al. (2005) propose an algorithm to collapse signatures. To this end, signature transforms, which consist of the signature-affix pairs are defined. Signatures are collapsed using context vector similarities, where the context vectors consist of high frequency words and the signature transforms.

Remark. We also propose an algorithm that employs the syntactic clusters to build paradigms for morphological segmentation. The details of the algorithm are presented in Chapter 4 (see also Can & Manandhar (2009)).

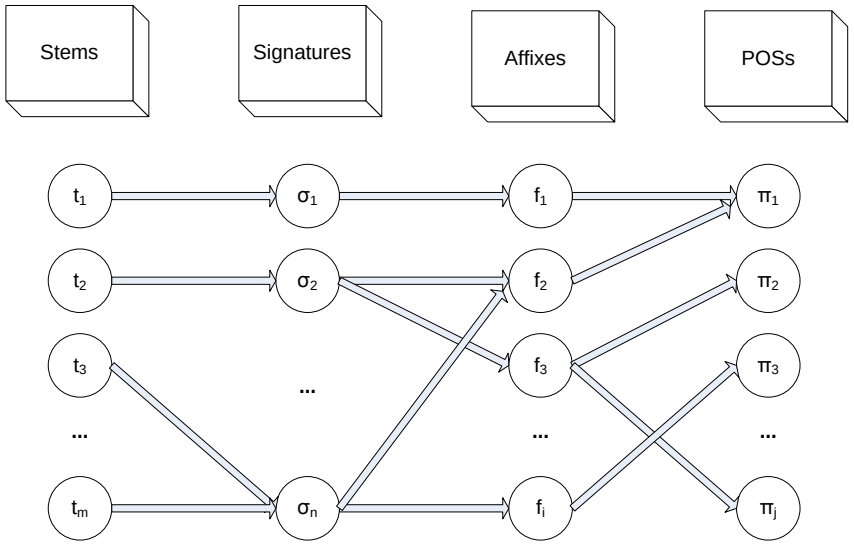


Figure 3.6: The derived model from Linguistica that employs the parts-of-speech of words as a separate list (Hu et al. 2005).

3.3.5.2 Learning Syntactic Categories Using Morphology

Clark (2003) employs morphological information to apply a distributional clustering to infer the part-of-speech tags of rare words. As Clark implies, POS tagging is not an easy task for especially infrequent words. To this end, he employs morphological information as well as the context of the words to enable the tagging process for infrequent words. Clark applies Ney-Essen clustering (Ney et al. 1994; Martin et al. 1998) for syntactic clustering, and combines the clustering with morphological information. Therefore, morphologically similar words are grouped into the same syntactic cluster. Frequency information is also added to the model. The reason is that the frequency of a word is also an indicator of its part-of-speech.

Example 3.3.2. A rare word is likely to be a noun, proper noun etc rather than a pronoun or article (Clark 2003).

Abend et al. (2010) suggest a POS tagging algorithm through prototype discovery. Prototypes are words which frequently represent each POS tag cluster.

In the algorithm, the authors define scores regarding the context of the words. These scores include the right/left adjacency score, which defines the tendency of a word to be a neighbour (the context word) of another word, and the interchangeability score, which defines the replacability of a word with another. Defining the scores in a vector, prototype clusters are formed by an average linkage clustering algorithm that clusters the words through the average scores that each cluster has. This procedure is repeated until the distance of the clusters become larger than a manually set threshold. Having the prototype clusters, unclustered words are mapped to one of the clusters with the minimum distance. The authors extend the algorithm further by using morphological segmentations of the words which are obtained from Morfessor (Creutz & Lagus 2005a). Morphological signatures are constructed, merging the words with the same endings within the same cluster. The average linkage clustering algorithm is applied in two steps, first, words are clustered through the morphological signatures, then the resulting clusters are clustered further. Similarly, unclustered words are mapped into one of the clusters to which they have a minimum distance.

3.3.5.3 Cooperation with Other Types of Linguistic Features

As can be seen from the literature, morphology and syntax have a close relationship, which enables them to be discovered in collaboration. Nevertheless, the relationship between linguistic constituents is not limited to this. For example, morphology also plays an important role in semantics, which enables an investigation of the meaning of a word by analysing its morphological segmentation. These relationships are usually mutual. Therefore, it is also possible to do a morphological segmentation using the semantics of a word.

Schone & Jurafsky (2000) employ semantic similarity between words to perform a more robust morphological segmentation. In the proposed algorithm, first, potential morphemes are discovered from the branchings, once the words have been inserted on a *trie*. Second, potential morphological variants and their rulesets are discovered from the *trie*, where the words share the same ancestors.

Example 3.3.3. Possible morphological variants could be *cars-car* or *caring-*

car, where the aim is to discover which morphological variants are legimitate (the example is taken from Schone & Jurafsky (2000)).

To determine whether the words are morphological variants, Latent Semantic Analysis (LSA) is used by Schone and Jurafsky. Latent Semantic Analysis is a method which was first initiated by Deerwester et al. (1990) with the aim of retrieving requested documents, based on the meaning of words rather than the word occurrences in the documents (for the methodology of LSA see Landauer et al. (1998)). Latent Semantic Analysis projects each word onto a semantic space by means of a method called Singular Value Decomposition (SVD). SVD decomposes a given matrix into different matrices. LSA constructs the dimensional semantic vectors of a word by using the matrices that are produced by SVD. Once the words are projected onto the semantic space, semantic correlations can be easily computed. Schone & Jurafsky (2000) use normalised cosine similarity to compute the semantic relatedness of the morphological variants.

Example 3.3.4. Some of the normalised cosine measurements of morphological variants are *car-cars*:5.6, *car-caring*:-0.71, *ally-allies*:6.5, *ally-all*:-1.3 (Schone & Jurafsky 2000). The scores prove that legitimate morphological variants have a higher cosine similarity, whereas the invalid morphological variants have a low cosine similarity.

Schone & Jurafsky (2001) extend the LSA based approach of Schone & Jurafsky (2000) using syntactic and orthographic features, as well as semantic features. The new approach also considers circumfixing[5] and employs transitive closure. The new algorithm follows these steps: 1. Potential morphological variants are discovered, considering the circumfixing where, first, potential prefixes, and second, potential suffixes are stripped off the words, to construct the potential circumfixes 2. Semantic correlations are computed through the semantic vectors that are built using LSA 3. Orthographic features (in terms of affix frequencies) are deployed 4. Syntactic contexts (left and right neighbours) of the morphological variants are examined 5. Finally, transitive closures are discovered, to

[5]A circumfix is a morphological structure which independently occurs in any part of a word by wrapping around other units. A well-known example is the German past participle *ge–s* such that the German word *spielen* has the past participle form as *gespielt*.

capture more legitimate morphological variants which were not discovered in the earlier steps.

Remark. To our knowledge, there are no remarkable contributions in the field of learning morphology and syntax simultaneously. Rather than a simultaneous learning, the literature has samples of sequential learning where first, one of the fields is learned, and then the other one is learned by using the inferred knowledge. This is the source of inspiration for Chapter 6 where a joint learning model is proposed.

3.4 Conclusion

In this chapter, a literature review of unsupervised morphology learning and POS tagging is presented. We are aware that there is a remarkable amount of work on supervised learning of morphology and syntax. However, as the scope of the thesis is bounded to unsupervised learning methods, only research based on unsupervised learning is reviewed.

The first section of the chapter (Section 3.2) reviews the literature in unsupervised morphology learning. The contributions in the field are categorised according to the type of mathematical model that forms the basis of the approach. The categorisation consists of letter successor variety models, information theoretic models and stochastic models. Some prominent examples in the area representing each methodology are presented.

The second section of the chapter (Section 3.3) reviews the literature in unsupervised POS tagging. The contributions are classified into two main categories: clustering models and HMMs. The most prominent research in the area is reviewed to give an idea of how the field has evolved.

The reader should note that the chapter reviews most of the significant contributions in the area. However, morphology learning and POS tagging are two long-standing fields in natural language processing, the literature of the fields is rather broad and it is difficult to cover all the attempts in the timeline.

CHAPTER 4

Morphological Segmentation Using Syntactic Categories

"A hundred times every day I remind myself that my inner and outer life depend on the labors of other men, living and dead, and that I must exert myself in order to give in the same measure as I have received and am still receiving."

Albert Einstein

4.1 Introduction

In this chapter, an algorithm for unsupervised morphological segmentation is presented. The algorithm uses syntactic categories to capture morphological paradigms for unsupervised morphological segmentation.

The chapter is organised as follows: Section 4.2 motivates the research presented in this chapter; Section 4.3 describes the algorithm for capturing paradigms; Section 4.4 explains the segmentation process; Section 4.5 demonstrates the experiments performed.

4.2 Motivation

The inspiration behind the work presented in this chapter comes from the morphological similarity of words that belong to the same syntactic category. For example, words ending with *-ly* are typically adverbs in English, (e.g. *quickly*, *patiently*), whereas words ending with *-ful* or *-ive* are typically adjectives (e.g. *successful*, *respective*).

Most morphological segmentation systems (Goldsmith 2006; Creutz & Lagus 2005b) do not exploit syntactic information to learn morphology of a given corpus. These systems only use word-based statistics rather than context-based statistics.

However, there has been research that makes use of the correlation between syntactic and morphological information. Some research exploits syntactic information in order to perform morphological segmentation (Freitag 2005; Hu et al. 2005). On the other hand, some research exploits morphological information in order to learn syntactic categories (Clark 2003).

The contribution in this chapter is closer to Freitag's work (Freitag 2005) where he induces transformation rules from term clusters. These clusters are rough syntactic groups. He defines transformation rules between syntactic groups to analyse words morphologically.

Differently from the other research that utilises syntactic information in order to learn morphology, we motivate our research from a paradigmatic perspective. Paradigms are great sources of morphological information by covering morphological relation between words as well as providing flexibility to generate new word forms. With this work, we aim to capture paradigms by using syntactic categorial information.

4.3 Learning Morphological Paradigms Using Syntactic Categories

In this work, we only focus on capturing paradigms and leave learning syntactic categories beyond this research. In order to learn syntactic categories, we adopt

distributional clustering algorithm of Clark (Clark 2000, 2001). Following a brief description of the distributional clustering algorithm, we will explain our algorithm for capturing paradigms.

4.3.1 Learning Syntactic Categories

In order to learn syntactic categories, we adopt Clark's distributional clustering algorithm (Clark 2000, 2001) which can be considered as an instance of average link clustering. It should be emphasised that any other method for unsupervised induction of part-of-speech (POS) tags can be substituted without affecting the method presented in this chapter. Clark (Clark 2000, 2001) uses the same intuition with other researchers who employ local distributional information of words for learning syntactic categories; that is:

"Similar words occur in similar contexts."

Following Clark's approach, each word is clustered by using its context. A context consists of the previous word and the following word. Each word has a context distribution over all ordered pairs of left-context/right-context words. To measure the distributional similarity between words, Kullback-Leibler (KL) divergence is used which is defined as:

$$D(p\|q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \tag{4.1}$$

where p, q are the context distributions of the words being compared and x ranges over contexts.

In his approach (Clark 2000), Clark defines the context as $< w_1, w_2 >$, where w_1 denotes the previous word and w_2 denotes the following word. The probability of a context for a target word is calculated as follows:

$$p(< w_1, w_2 >) = p(< c_1, c_2 >)p(w_1|c_1)p(w_2|c_2) \tag{4.2}$$

where c_1 and c_2 denote the POS cluster of words w_1 and w_2 respectively.

Hence, the KL divergence between two words becomes:

$$D(p_1 \| p_2) = \sum_{w1,w2} p_1(< w1, w2 >) \log \frac{p_1(< w1, w2 >)}{p_2(< w1, w2 >)} \tag{4.3}$$

Equation 4.3 is simplified as follows (Clark 2001):

$$\begin{aligned} D(p_1 \| p_2) &= \sum_{w1,w2} p_1(< w1, w2 >) log \frac{p_1(< w1, w2 >)}{p_2(< w1, w2 >)} \\ &= \sum_{w1,w2} p_1(< c1, c2 >) p(w_1|c_1) p(w_2|c_2) \\ &\quad \log \frac{p_1(< c1, c2 >) p(w_1|c_1) p(w_2|c_2)}{p_2(< c1, c2 >) p(w_1|c_1) p(w_2|c_2)} \\ &= \sum_{c_1,c_2} \sum_{w_1 \in c_1} \sum_{w_2 \in c_2} p_1(< c_1, c_2 >) p(w_1|c_1) p(w_2|c_2) \\ &\quad log \frac{p_1(< c1, c2 >)}{p_2(< c1, c2 >)} \\ &= \sum_{c_1,c_2} p_1(< c1, c2 >) \log \frac{p_1(< c1, c2 >)}{p_2(< c1, c2 >)} \end{aligned}$$

Thus, KL divergence between two words becomes simply the divergence between context distributions of the words over clusters.

The algorithm requires the number of clusters K to be specified in advance. In addition to K clusters, one spare cluster is employed containing all unclustered words. Initially, K clusters are filled by one of the most frequent words in the corpus. During each iteration, one word is chosen from the spare cluster having the minimum KL divergence with one of the K clusters. For each cluster, its context distribution is computed as the averaged distribution of all words in the cluster. In addition, the KL divergence between clusters is computed after each iteration and clusters are merged if the divergence is below a manually set threshold.

We set K=77, the number of tags defined in CLAWS tagset used for tagging the BNC (British National Corpus). We use the same number of clusters for Turkish and German. Final clusters show that POS clusters are related with the major syntactic categories. The algorithm finds syntactic categories that can be

Cluster 1	much, far, badly, deeply, strongly, thoroughly, busy, rapidly, slightly, heavily, neatly, widely, closely, easily, profoundly, readily, eagerly etc.
Cluster 2	made, found, held, kept, bought, heard, played, left, passed, finished, lost, changed, etc.
Cluster 3	should, may, could, would, will, might, did, does, etc.
Cluster 4	working, travelling, flying, fighting, running, moving, playing, turning, etc.
Cluster 5	people, men, women, children, girls, horses, students, pupils, staff, families, etc.

Table 4.1: Some sample syntactic categories obtained from the English dataset.

identified as proper nouns, verbs in past tense form, verbs in present continuous form, nouns, adjectives, adverbs, and so on (see Table 4.1).

Example 4.3.1. An illustration of the algorithm is given in Figure 4.1. Last cluster shows the ground cluster which involves all the unclustered words. The rest of the clusters involve the clustered words. In each iteration, the KL divergence for all the words in the ground cluster is calculated for each cluster. For each word, the cluster which yields the minimum KL divergence is kept and the others are discarded. After sorting all the KL divergences for each word, the word that yields the minimum KL divergence is chosen. The chosen word is removed from the ground cluster and placed in the cluster that yields the minimum divergence. Imagine that *beautiful* has the minimum KL divergence with the third cluster when it is compared with all the other KL divergences between each word in the ground cluster and the existing clusters. Therefore, *beautiful* is placed in the first cluster in the next iteration.

4.3.2 Identifying Potential Morphemes

Each POS cluster includes a set of potential morphemes produced by splitting each word into all possible stem-suffix combinations. For each potential morpheme, we calculate its conditional probability $p(m|c)$ where *m* denotes the morpheme, and *c* denotes the cluster. When potential morphemes are ranked according

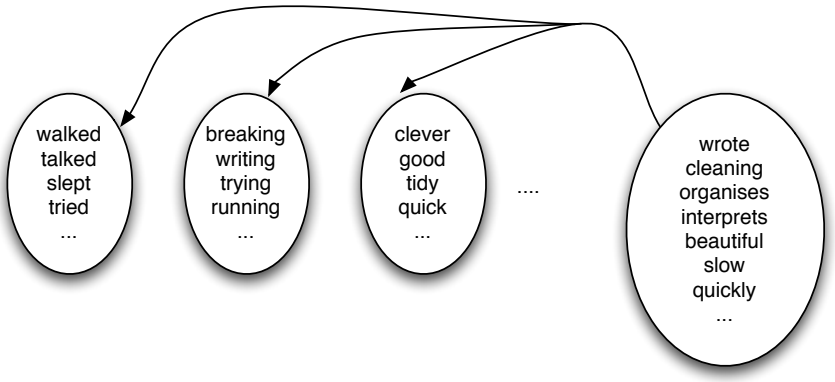


Figure 4.1: An illustration of the distributional clustering algorithm.

English		German	
Cluster	Morphemes	Cluster	Morphemes
1	-s	1	-n,-en
2	-d,-ed	2	-e,-te
3	-ng,-ing	3	-g,-ng,-ung
4	-y,-ly	4	-r,-er
5	-s,-rs,-ers	5	-n,-en,-rn,-ern
6	-ing,-ng,g	6	-ch,-ich,-lich

Table 4.2: Some high ranked potential morphemes in PoS clusters for English and German.

to their conditional probabilities, only those above a threshold (see Section 4.5) are considered in the following step. This ranking is used to eliminate the potential non-morphemes with a low conditional probability hence reducing the search space.

A list of highest ranked morphemes are given in Table 4.2 for English, German, and in Table 4.3 for Turkish.

Turkish	
Cluster	Morphemes
1	-i,-si,-ri
2	-mak,-mek,-mesi,-masi
3	-an,-en
4	-r,ar,er,-ler,-lar
5	-r,-ir,-dir,-ır,-dır
6	-e,-a

Table 4.3: Some high ranked potential morphemes in PoS clusters for Turkish.

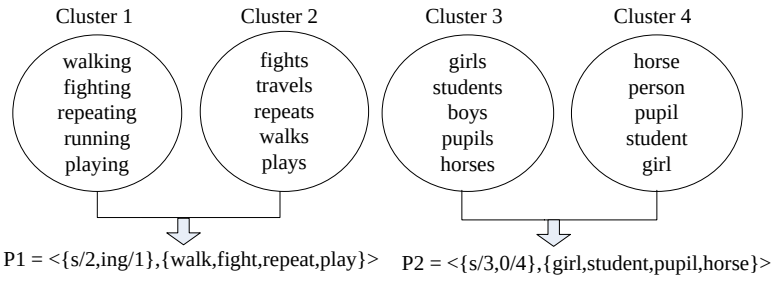


Figure 4.2: A sample set of syntactic clusters, and the potential morphemes in each cluster

4.3.3 Inducing Mophological Paradigms

Our definition of a paradigm deviates from that of Goldsmith (2001) due to the addition of POS tags. In our framework, each morpheme is tied to a POS cluster. More precisely, a paradigm *P* is a list of morpheme/cluster pairs together with a list of stems: $P = < \{m_1/c_1, \ldots, m_r/c_r\}, \{s_1, \ldots, s_k\} >$ where m_i denotes a morpheme belonging to the POS cluster c_i, and s_j denotes a stem such that $\forall m_i/c_i \in \{m_1/c_1, \ldots, m_r/c_r\}$ and $\forall s_j \in \{s_1, \ldots, s_k\} : s_j + m_i \in c_i$.

In each iteration, a potential morpheme pair across two different POS clusters with the highest number of common stems is chosen for merging. Once a morpheme pair is merged, words that belong to this newly formed paradigm are removed from their respective POS clusters. This forms the basis of the paradigm-capturing mechanism (see Fig. 4.2). We postulate that a word can only belong

Algorithm 1 Algorithm for paradigm-capturing using syntactic categories

1: Apply unsupervised PoS clustering to the input corpus.
2: Generate all possible morphemes by splitting the words in all possible stem-suffix combinations.
3: For each PoS cluster c and morpheme m, compute maximum likelihood estimates of $p(m \mid c)$.
4: Keep all m (in c) with $p(m \mid c) > t$, where t is a threshold.
5: **repeat**
6: **for all** PoS clusters c_1, c_2 **do**
7: Pick morphemes m_1 in c_1 and m_2 in c_2 with the highest number of common stems.
8: Store $P = \{m_1/c_1, m_2/c_2\}$ as the new paradigm.
9: Remove all words in c_1 with morpheme m_1 and associate these words with P.
10: Remove all words in c_2 with morpheme m_2 and associate these words with P.
11: **end for**
12: **for** each paradigm pair P_1, P_2 such that $Acc(P_1, P_2) > T$, where T is a threshold **do**
13: Create new merged paradigm $P = P_1 \cup P_2$.
14: Associate all words from P_1 and P_2 with P.
15: Delete paradigms P_1, P_2. .
16: **end for**
17: **until** No morpheme pair consisting of at least one common stem is left

to a single morphological paradigm. The above procedure is repeated until no further paradigms are created.

Algorithm 1 describes the complete paradigm capturing process. Some sample paradigms captured are given in Table 4.4, Table 4.5, and Table 4.6 for English, Turkish and German respectively.

4.3.4 Merging Paradigms

For capturing more general paradigms, paradigm merging is performed. We rank potential paradigms by the ratio of common stems with the total number of stems captured by the paradigm. More precisely, given paradigms P_1, P_2, let S be the total number of common stems. Let S_1 be the total number of stems in P_1 that are not present in P_2. Similarly, let S_2 be the total number of stems in P_2 that

ed ing	reclaim, aggravat, hogg, trimm, expell, administer, divert, register, stimulat, shap, rehabilitat, exempt, stiffen, spar, deceiv, contaminat, disciplin, implement, stabiliz, feign, mistreat, extricat, mimick, alert, seal, etc.
s d	implicate, ditche, amuse, overcharge, equate, despise, torpedoe, curse, plie, supersede, preclude, snare, tangle, eclipse, relinquishe, ambushe, reimburse, alienate, conceive, vetoe, waive, envie, negotiate, diagnose, etc.
er ing	brows, wring, worship, cropp, cater, stroll, zipp, moneymak, tun, chok, hustl, angl, windsurf, swindl, cricket, painkill, climb, heckl, improvis, scream, scaveng, panhandl, lawmak, bark, clean, lifesav, beekeep, toast, matchmak, bodybuild, etc.
e ed	subsid, liquidat, redecorat, exorcis, amputat, fertiliz, reshap, regulat, foreclos, infring, eradicat, reverberat, chim, centralis, restructur, crippl, rehabilitat, symbolis, reinstat, etc.
ly er	dark, cheap, slow, quiet, fair, light, high, poor, rich, cool, quick, broad, deep, bright, calm, crisp, mild, clever, etc.
0 s	benchmark, instrument, pretzel, wheelchair, scapegoat, spike, infomercial, catastrophe, beard, paycheck, reserve, abduction, etc.

Table 4.4: Sample paradigms in English

are not present in P_1. Then, we can define the expected paradigm accuracy of P_1 with respect to P_2 by:

$$Acc_1 = \frac{S}{S + S_1} \tag{4.4}$$

Acc_2 is defined analogously.

We use the average of Acc_1 and Acc_2 to compute the combined (averaged) expected accuracy of the merged paradigms P_1, P_2:

$$Acc(P_1, P_2) = \frac{\frac{S}{S+S_1} + \frac{S}{S+S_2}}{2} \tag{4.5}$$

During each iteration, all paradigm pairs having an expected accuracy greater than a given threshold value are merged (see Figure 4.3). Once two paradigms are merged, stems that occur in only one of the paradigms inherit the morphemes from the other paradigm. This mechanism helps create a more general paradigm

i e	zemin, faaliyetin, törenler, seçim, incelemeler, eyalet, nem, takvim, makineler, yöntemin, becerisin, görüşmeler, tekniğin, merkezin, iklim, görüntüler, etc.
i a	cevab, bakımın, mektuplar, esnaf, olayın, akışın, miktar, kayd, yaşamay, bulgular, sular, masrafların, heyecanın, kalan, hakların, anlamın, etc.
i in	sanayiin, değerlerin, eşin, denizler, duman, teminat, erkekler, kurulların, birbirin, vatandaşlarımız, gelişmesin, milletvekillerin, partisin, etc.
de e	bölgesin, düzeyin, yönetimin, dergisin, sektörün, birimlerin, bölgelerin, tümün, bölümlerin, tesislerin, dönemin, kongresin, evin, etc.
mesi en	izlen, yürütül, degiş, üretil, gerçekleştiril, desteklen, geliştiril, etc.
0 i	iman, çekim, mahkemelerin, örneklem, gaflet, yazman, sanat, trendler, mahalleler, eviniz, hamamlar, piller, öğretim, olimpiyat, etc.

Table 4.5: Sample paradigms in Turkish

r n	kurze, ehemalige, eidgenoessische, professionelle, erste, bescheidene, ungewoehnliche, ethnische, unbekannte, besondere, nationalsozialistische, deutsche, etc.
e en	praechtig, gesichert, dauerhaft, bescheiden, vereinbart, biologisch, natuerlich, oekumenisch, kantonal, unterirdisch, wissenschaftlich, nahegelegen, chinesisch, etc.
t en	funktionier, konkurrier, schneid, mitwirk, ansteig, plaedier, pfeif, aufklaer, schluck, ausgleich, weitermach, abhol, ankomm, spazier, speis, aussteig, aufhoer, etc.
er ung	versteiger, unterdrueck, erneuer, vermarkt, beschleunig, besetz, geschaeftsfuehr, wirtschaftsfoerder, finanzverwalt, verhandl, etc.
0 s	potential, instrument, flohmarkt, vorhang, pilotprojekt, idol, rechner, thriller, ensemble, bebauungsplan, empfinden, defekt, aufschwung, etc.

Table 4.6: Sample paradigms in German

and helps recover missing word forms. Thus, although some of the word forms do not exist in the corpus, it becomes possible to capture these forms.

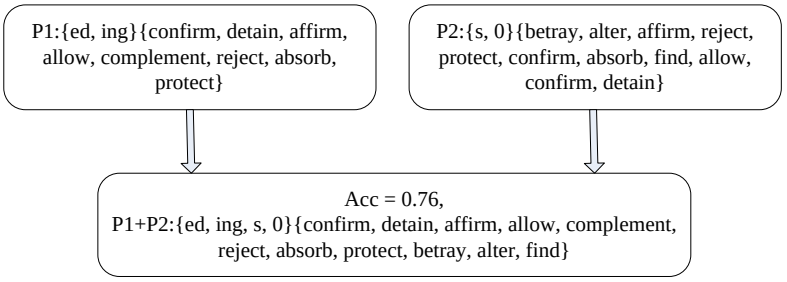


Figure 4.3: An illustration of paradigm merging. $P1$ and $P2$ are merged with an accuracy measure of $Acc = 0.76$.

es ing e ed	sketch, chew, nipp, debut, met, factor, profit, occurr, err, trudg, participat, necessitat, stomp, streak, siphon, stroll, sprint, drizzl, firm, climax, gestur, whipp, roll, tripp, stemm, dangl, shuffl, kindl, broker, chalk, latch, rippl, collaborat, chok, summ, propp, pedal, paralyz, parad, plough, cramm, slack, wad, saddl, conjur, tipp, gallop, totall, catalogu, bundl, barg, whittl, retaliat, straighten, tick, peek, jabb, slimm.
s ing ed 0	benchmark, mothball, weed, snicker, thread, queue, jack, paw, yacht, implement, import, bracket, whoop, conflict, spoof, stunt, bargain, honor, bird, fingerprint, excerpt, handcuff, veil, comment.

Table 4.7: Merged paradigms in English

Some example paradigms that are found by the system are given below in Table 4.7, Table 4.8, and Table 4.9 for English, Turkish, and German respectively.

4.4 Morphological Segmentation

Once words are clustered given a corpus thereby creating POS clusters, steps described earlier are followed to capture paradigms. Having paradigms, words are analysed by following different algorithms for known, unknown, and compound

u a e i	yapabileceklerin, kredisin, hizmetleri'n, sevdikleriniz, yeter', transferlerin, sevkin, elimiz, tehlikelerin, sas, mucizey, tehditlerin, bakir, muhasebesin, gayrimenkuller, ecevit', defterim, izlemelerin, tescilin, minarey, tahsilin, lastikler, yerlestirmey.
i lar li in	ruhsat, semt, ikilem, reaksiyonlar, harc, tip, prim, gidilmis, kaldirmis, degistirmis, bulunmayacak, aktarmis, bulunacak, kapanacak, yazilabilecek, devredilmis, degisecek, gelmemis.

Table 4.8: Merged paradigms in Turkish

er 0 e en	kassiert, beguenstigt, eingeholt, genuegt, angelastet, beruehrt, beinhaltet, zurueckgegeben, beschleunigt, initiiert, abgestellt, bewirkt, mitgenommen, abgebrochen, beruhigt, besichtigt.
te ung er ten t en lich e	fahr, gebrauch, blockier, identifizier, studier, entfalt, gestalt, agier, passier, sprech, berat, tausch, kauf, such, weck, beug, erreich, bearbeit, beobacht, erleid, ueberrasch, halt, helf, oeffn, pruef, uebertreff, bezahl, spring, fuell, toet.
0 te t er	lichtenberg, limburg, hill, trier, elmshorn, dreieich, praunheim, heusenstamm, heddernheim, hellersdorf, schmitt, muehlheim, lueneburg, kassel, schluechtern, preungesheim, rodgau, bieber, osnabrueck, rodheim, muenchen, london, lissabon, seoul, wedding, treptow.

Table 4.9: Merged paradigms in German

words:

4.4.1 Handling known words

If the word exists in one of the paradigms, it is segmented by using the morpheme in the paradigm in which the word is found. For example, let a paradigm exist as given below:

s ing ed 0 : benchmark mothball weed snicker thread queue jack paw yacht implement import bracket whoop

If a word 'importing' is to be morphologically analysed, it is automatically

segmented by using the morpheme 'ing'.

4.4.2 Handling unknown words

If the word does not exist in any of the paradigms, a sequence of segmentation rules are applied. By using paradigms, we created a morpheme dictionary to split the words which do not belong to any of the paradigms. All morphemes in each paradigm are included in the morpheme dictionary if in any of the paradigm the initial letters of the morphemes are not the same. If the initial letters of all morphemes in the same paradigm are the same, the longest morpheme is included in the dictionary. Using the morpheme dictionary, the word is scanned from the right-most letter to check if any of the endings of the word exist in the dictionary. The longest letter sequence (of the word) existing in the dictionary is chosen to split the word. The same process is repeated after splitting the word until no split can be applied.

4.4.3 Handling compounds

For the compounds, such as 'hausaufgaben' in German, or 'railway' in English, for both known, and unknown words a recursive approach is performed. The compounding rules split a word recursively from the rightmost end to the left. If an ending sequence of letters exists as a word in the corpus, the sequence is split, and the same procedure is repeated until no valid internal word part is a valid word itself in the corpus. When there are multiple matches the longest match is chosen. This recursive search is also able to find the prefixes as it searches for the valid sub-words in words.

Algorithm for the segmentation of the words is given in Algorithm 2.

4.5 Experiments & Evaluation

The algorithm was evaluated in the Morpho Challenge 2009 competition. The corpus from Morpho Challenge 2009 and the Cross Language Evaluation Forum

Algorithm 2 Morphological Segmentation

1: **for all** For each given word, w, to be segmented **do**
2: **if** w already exists in a paradigm P **then**
3: Split w using P as $w = u + m$
4: **else**
5: $u = w$
6: **end if**
7: If possible, split u recursively from the rightmost end by using the morpheme dictionary as $u = s_1 + \ldots + s_n$ otherwise $s_1 = u$
8: If possible, split s_1 into its sub-words recursively from the rightmost end as $s_1 = w_1 + \ldots + w_n$
9: **end for**

(CLEF) 2009 were used for training our system on 3 different languages: English, German and Turkish. For the initial POS clustering, corpora provided in Morpho Challenge 2009[1] were used. For clustering the words in the word list to be segmented, for English and German, datasets supplied by the CLEF organization [2] were used. For Turkish, we made use of manually collected newspaper archives.

Although our model is unsupervised, two prior parameters are required to be set: *t* for the conditional probability $p(m|c)$ of the potential morphemes and T for the paradigm accuracy threshold for merging the paradigms. We set t=0.1 and T=0.75 in all the experiments.

The system was evaluated in Competition 1 & 2 of Morpho Challenge 2009. In Competition 1, proposed analyses are compared to a gold standard analysis of a word list. Details of the tasks and evaluation can be found in the overview and results of Morpho Challenge 2009, which was reported in Kurimo et al. (2009) (see also Kurimo et al. (2011a)). Evaluation results corresponding to the Competition 1 are given in Table 4.5. Our system comes 8th out of 14 participant systems in English, whereas the system comes 4th out of 15 systems in German and the system comes 8th out of 14 participant systems in Turkish. These scores are scientifically significant due to the well-established evaluation framework

[1] http://www.cis.hut.fi/morphochallenge2009

[2] http://www.clef-campaign.org/. English datasets: Los Angeles Times 1994 (425 mb), Glasgow Herald 1995 (154 mb). German datasets: Frankfurter Rundschau 1994 (320 mb), Der Spiegel 1994/95 (63 mb), SDA German 1994 (144 mb), SDA German 1995 (141 mb)

Language	Precision(%)	Recall(%)	F-measure(%)	F/Winner(%)
English	58.52	44.82	50.76	62.31
German	57.67	42.67	49.05	56.14
Turkish	41.39	38.13	39.70	53.53

Table 4.10: Obtained evaluation scores in Morpho Challenge 2009 Competition 1 with the winner participant's F-score.

Language	AP(%)	AP(%)-Winner
English	0.2940	0.3890
German	0.4006	0.4490

Table 4.11: Obtained average precisions (AP) for the Morpho Challenge 2009 Competition 2

provided by Morpho Challenge.

In the Competition 2, proposed morphological analyses are used in an information retrieval task. For this purpose, words are replaced with their morphemes. Our results for German had an average precision of 0.4006% whereas the winning system Christian Monson (2009) had an average precision of 0.4490%. Our results for English had an average precision of 0.2940% whereas the winning system Lignos et al. (2009) had an average precision of 0.3890% (see Table 4.5).

We did not perform an evaluation on the created paradigms since the evaluation requires a huge amount of corpus that involves all possible word forms in the given language. Having such a corpus, it will be possible to check each paradigm whether it consists of valid stem+suffix combinations. Preparing such a corpus requires extra work and we skipped the evaluation of accuracy of paradigms in this work. However, the morphological segmentation results provide a significant indicator on the accuracy of paradigms.

4.6 Conclusion

To our knowledge, there has been limited work on the combined learning of syntax and morphology. In Morpho Challenge 2009, our model is the only system making use of the syntactic categories. Morphology is highly correlated with

syntactic categories of words. Therefore, our system is able to find the potential morphemes by only considering conditional probabilities $p(m|c)$.

The paradigm including the most number of stems for English has the morpheme set {*s, ing, ed, 0*} where *0* denotes the NULL suffix, for Turkish it has {*u, a, e, i*}, and for German it has {*er, 0, e, en*}. Our paradigm merging method is able to compensate for the missing forms of the words. For example, as shown in Figure 4.3, although the words such as *altering*, and *finding* do not exist in the real corpus, they are produced during the merging. However, our system still requires a large dataset for POS clustering. We only consider words having a frequency greater than 10 to eliminate noise. To segment non-frequent words we propose a heuristic method based on using a morphmeme dictionary. However, the usage of such a morpheme dictionary can often have undesirable results. For example, the word *beer* is forced to be segmented as *be-er* due to the morpheme *er* found in the dictionary.

Our model allows more than one morpheme boundary. This makes our system usable for the morphological analysis of the agglutinative languages. For example, in Turkish, the word *çukurlarıyla* (which means *"with their burrows"*) has the morpheme boundaries: *çukur-lar-ı-y-la*. However, in our heuristic method, the use of morpheme dictionary causes undesirable results. For example, the same word *çukurlarıyla* is segmented by our method as: *çu-kurları-y-la* since the word *kurları* also exists in the corpus.

Our system is sensitive due to the thresholds we set for 1. identifying potential morphemes and 2. expected paradigm accuracy. In future work, we hope to address these and previously mentioned deficiencies.

Despite the observed deficiencies, we obtained promising results in Morpho Challenge 2009. Our precision and recall values are balanced and undersegmentation is not very prominent for all languages that we evaluated. We believe that our work clearly demonstrates that joint modeling of syntactic categories and morphology is the key for building successful morphological analysis system. In addition, our work demonstrates how morphological paradigms can be learnt by taking advantage of POS categories.

CHAPTER 5

Probabilistic Hierarchical Clustering for Morphology Learning

"From where we stand, the rain seems random. If we could stand somewhere else, we would see the order in it."
Tony Hillerman

5.1 Introduction

In this chapter, a probabilistic hierarchical clustering algorithm for morphological segmentation is presented. An inference algorithm is introduced to capture latent variables in data along with the hierarchical structure. Latent variables are morpheme boundaries of words in morphological segmentation.

The chapter is organised as follows: Section 5.2 motivates the research presented in this chapter both linguistically and computationally; Section 5.3 describes the probabilistic hierarchical model; Section 5.4 defines the mathematical model for morphological segmentation; Section 5.4.2 describes the employment of the clustering algorithm for morphological segmentation; and finally, Section 6.6 presents experiments and results.

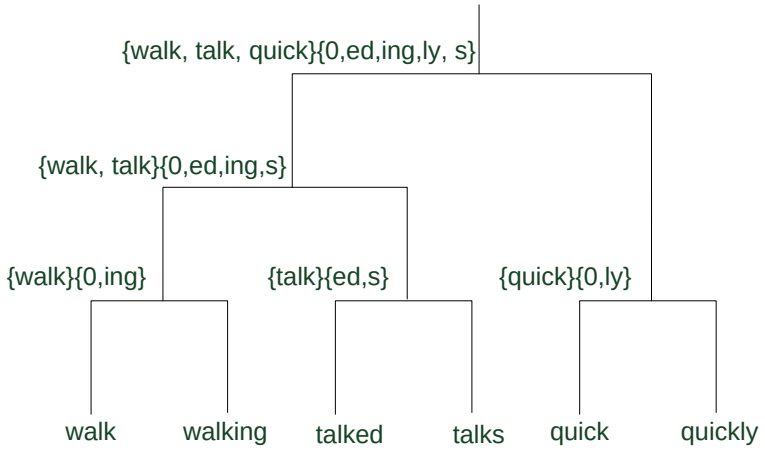


Figure 5.1: A sample tree structure.

5.2 Motivation

The proposed algorithm in this chapter is motivated both linguistically and computationally. The algorithm is linguistically motivated in the sense that it bears a new perspective for morphological segmentation. With the new perspective, the aim is to capture morphological paradigms hierarchically. A hierarchical structure will be efficient to extract morphological regularities between words which will naturally lead to discover morphological paradigms (see Figure 5.1). We also propose that if morphological paradigms are defined probabilistically, it will also overcome any sparsity in the data. For these purposes, we define a probabilistic hierarchical clustering algorithm that captures morphological paradigms to be employed for morphological segmentation. There has been research in the field of morphological segmentation that employed morphological paradigms; however, to our knowledge, there has not been any research that combines statistical hierarchical clustering with morphological segmentation through a paradigmatic structure.

Remark. Chan (2006) employs paradigms within a hierarchical structure where Latent Dirichlet Allocation (LDA) is used to discover stem-suffix matrices. However, true morphological analyses of words are assumed to be provided to the system. Therefore, the proposed model focuses only on capturing paradigms. On the contrary, the suggested model in this chapter learns both morphological segmentation and paradigms along with a hierarchical structure.

The algorithm is computationally motivated in the sense that it defines an inference algorithm to discover latent variables in data along with a hierarchical representation. Most hierarchical clustering algorithms are single-pass where once the hierarchical structure is built, the structure does not change anymore. However, it may be the case that the hierarchical representation of data may not be the only aim that needs to be achieved. Another aim may be the discovery of latent variables in data. In the circumstances, latent variables have to be discovered along with a hierarchical structure. For the problem of morphological segmentation, morpheme boundaries are latent variables that have to be discovered. We propose an inference algorithm that learns morpheme boundaries associated with a structure which will represent morphemes hierarchically. The proposed inference algorithm samples new tree structures until finding one consistent hierarchical representation of latent variables.

5.3 Probabilistic Hierarchical Model

The hierarchical clustering, proposed with this research, is different than many traditional hierarchical clustering algorithms in two aspects:

- It is not single-pass as hierarchical structure changes.
- It is probabilistic and does not use distance metric.

5.3.1 Mathematical Definition

The suggested hierarchical structure for the hierarchical clustering algorithm is a binary tree where each internal node represents a cluster. Data points are located in leaf nodes.

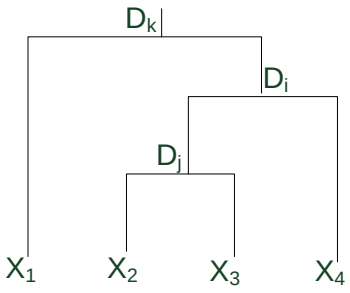


Figure 5.2: A segment of a tree with internal nodes D_i, D_j, D_k having data points $\{x_1, x_2, x_3, x_4\}$. The subtree below the internal node D_i is called T_i, the subtree below the internal node D_j is called T_j, and the subtree below the internal node D_k is called T_k.

Let a data set be $\mathbf{D} = \{x_1, x_2, \ldots, x_n\}$ and T be the entire tree where each data point x_i is located in one of the leaf nodes (see Figure 5.2). Here D_k denotes data points in the branch T_k. Each node defines a probabilistic model for words that the cluster acquires. The probabilistic model can be denoted as $p(x_i|\theta)$ where θ represents parameters of the probabilistic model. It is possible to embed different types of probabilistic models with different number of parameters. It is also possible to define a prior probability distribution over parameters of the probabilistic model; such that $p(\theta|\beta)$ with hyperparameters β.

Adopting a probabilistic model, the marginal probability of data in any node can be calculated:

$$p(D_k) = \int p(D_k|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\beta})d\boldsymbol{\theta} \tag{5.1}$$

Using conjugate priors makes the integration tractable. The likelihood of data in any subtree is defined as follows:

$$p(D_k|T_k) = \pi_k p(D_l|T_l)p(D_r|T_r) \tag{5.2}$$

where the probability of data in any node under a tree structure is defined in terms of left T_l and right T_r subtrees since the data is the combination of two different clusters. When Equation 5.2 is applied recursively for left and right subtrees

until reaching leaf nodes, the likelihood of data can be obtained under the given tree structure. The likelihood of data will be used for inference algorithm. In the equation, π_k denotes the prior probability of the node. The marginal probability of data will be exploited as prior probability (Equation 5.1). We use the marginal probability as prior information since the marginal probability bears the probability of having the data in one cluster.

5.4 Morphological Segmentation

The hierarchical model proposed earlier is employed for capturing morphological paradigms where data points are words to be clustered and each cluster represents a paradigm. In the hierarchical structure, words will be organised in such a way that morphologically similar words will be located close to each other to be grouped in the same paradigms.

5.4.1 Model Definition

As mentioned immediately above, each cluster is generated from a probabilistic model. The embedded probabilistic model is a morphological segmentation model as part of this thesis.

Let a dataset D consist of words to be analysed where each word w_i has a latent variable which is the split point that analyses the word into its stem s_i and suffix m_i:

$$\boldsymbol{D} = \{w_1 = s_1 + m_1, \ldots, w_n = s_n + m_n\}$$

The model can identify only one split point for each word yielding an analysis such that $w_n = s_n + m_n$.

A Dirichlet process model is chosen for the problem of morphological segmentation due to its flexibility to adapt to any kind of data. It should be noted that any probabilistic model can be embedded depending on the type of the data to be clustered. Depending on the probabilistic model, different set of parameters can be defined. If it is a multivariate Gaussian, then the parameters become: $\theta = (\mu, \Sigma)$. In the model that will be explained here, we use a Dirichlet process

model where the variables are generated from a Dirichlet process. Since each word is generated from a stem and a suffix, we define two Dirichlet processes to generate stems and suffixes independently. Here, we assume that stems and suffixes are independent, which is not completely true. However, morphemes are widely more independent, when other segments of words are considered.

Example 5.4.1. If we observe the split points in the word *slowly*, the probability $p(ly|slow)$ will be lower than the probability $p(wly|slo)$. The reason is that the number of possible segments that can exist before the segment -*ly* is more than the number of possible segments that can exist before the segment -*wly*. As both segments *ly* and *slow* can freely exist in the corpus by attaching to other segments, they are more independent than other segments and are assumed to be independent in our model.

Following a generative story, the probability of a morphological analysis of a word could be defined as follows:

$$p(w = s + m) = p(m)p(s|m) \tag{5.3}$$

which dictates that a suffix is generated, and a stem is generated conditioned on the generated suffix subsequently. However, applying the independence assumption, the probability of the morphological analysis can be redefined such that:

$$p(w = s + m) = p(m)p(s) \tag{5.4}$$

Stems and suffixes are generated from two distinct Dirichlet processes such that:

$$\begin{aligned} G_s &\sim DP(\beta_s, P_s) \\ G_m &\sim DP(\beta_m, P_m) \\ s &\sim G_s \\ m &\sim G_m \end{aligned} \tag{5.5}$$

where $DP(\beta_s, P_s)$ denotes a Dirichlet process that generates stems and $DP(\beta_m, P_m)$ denotes another Dirichlet process that generates suffixes. G_s and G_m are random probability distributions that are distributed according to the DPs. Here β_s and β_m are the concentration parameters and determine the number of stem types that can be generated by the Dirichlet process. The smaller the value of the concentration parameter is, less likely to generate new stem types by the process is (Goldwater & Griffiths 2007). On the contrary, the larger the value of concentration parameter is, the more likely it is to generate new stem types yielding a more uniform distribution over stem types. If $\beta_s < 1$, sparse stems will be supported yielding a more skewed distribution; however if $\beta_s > 1$, the distribution gets closer to a more uniform distribution assigning similar probabilities for each stem type; and if $\beta_s = 1$, the distribution becomes uniform where all stems are equally likely to be generated by the Dirichlet process. For these reasons, concentration parameter has a significant impact on the number of stems that can be generated from each cluster. The concentration parameter is also called strength parameter when the Dirichlet process is used as a prior in a Bayesian nonparametric model (Teh 2010). To support a small number of stem types in each cluster, we determined $\beta_s < 1$ for a skewed distribution. Otherwise, it is possible to have many stems due to an oversegmentation of words.

Here, P_s is the base distribution that determines the mean of the Dirichlet process (Teh 2010). The base distribution defines the properties of variables generated from the Dirichlet process (Goldwater et al. 2009); and it is independent from the concentration parameter. The base distribution can be either discrete or continuous. We use the base distribution as a prior probability distribution for morpheme lengths. Morpheme lengths can be modelled implicitly through letters that the morpheme consists of:

$$P_s(s_i) = \prod_{c_i \in s_i} p(c_i) \tag{5.6}$$

Letters are denoted by c_i where $p(c_i)$ is a uniform distribution defined on the alphabet letters. A letter in an alphabet with 27 letters will have a probability $1/27$. Modelling morpheme letters is a way of modelling morpheme lengths since shorter morphemes are favoured to have a smaller number of factors in

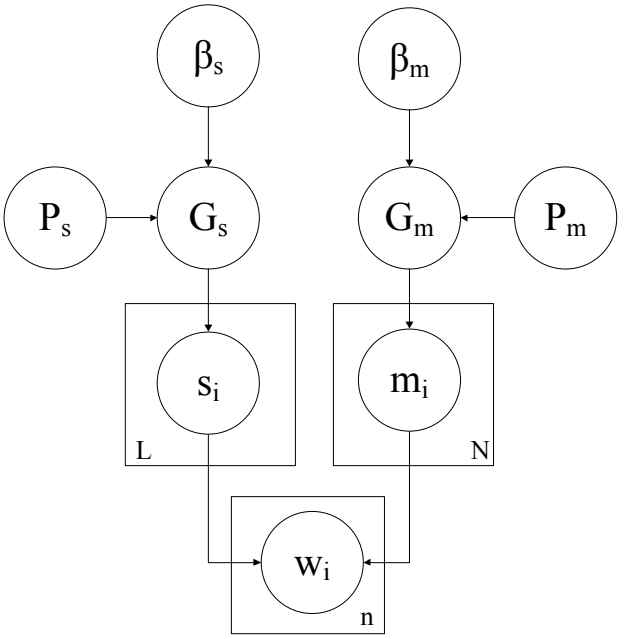


Figure 5.3: The plate diagram of the model, representing the generation of a word w_i from the stem s_i and the suffix m_i that are generated from Dirichlet processes. In the representation, solid-boxes denote that the process is repeated with the number given on the corner of each box.

Equation 5.6 (Creutz & Lagus (2005b)). Therefore, longer morphemes are less likely to be generated due to a low probability obtained from the base distribution.

The Dirichlet process, $DP(\beta_m, P_m)$, is defined for suffixes analogously. The graphical representation of the entire model is given in Figure 5.3.

Once the probability distributions $\mathbf{G} = \{G_s, G_m\}$ are drawn from both Dirichlet processes, words can be generated by drawing a stem from G_s and a suffix from G_m. However, we do not attempt to estimate the probability distributions $\mathbf{G}$; instead, $\mathbf{G}$ is integrated out. The joint probability distribution of stems after integrating out G_s becomes:

$$p(s_1, s_2, \ldots, s_L) = \int p(G_s) \prod_{i=1}^{L} p(s_i|G_s) dG_s \tag{5.7}$$

where L denotes the number of stem tokens. The joint probability distribution of stems can be tackled as a Chinese restaurant process. The Chinese restaurant process introduces dependencies between stems. Hence, the joint probability distribution of stems $S = \{s_1, \ldots, s_L\}$ becomes:

$$\begin{aligned} p(s_1, s_2, \ldots, s_L) &= p(s_1)p(s_2|s_1)\ldots p(s_L|s_1, \ldots, s_{L-1}) \\ &= \frac{\Gamma(\beta_s)}{\Gamma(L+\beta_s)} \beta_s^K \prod_{i=1}^{K} P_s(s_i) \prod_{i=1}^{K} (n_{s_i} - 1)! \end{aligned} \tag{5.8}$$

where K denotes the number of stem types. In the equation, the second and the third factor correspond to the case where novel stems are generated for the first time; last factor corresponds to the case where stems which have been already generated for n_{s_i} times previously are being generated again. The first factor consists of all denominators from both cases.

The integration process is applied for probability distributions G_m for suffixes analogously:

$$p(m_1, m_2, \ldots, m_N) = \int p(G_m) \prod_{i=1}^{N} p(m_i|G_m) dG_m \tag{5.9}$$

where N denotes the number of suffix tokens.

Hence, the joint probability distribution of suffixes is defined accordingly:

$$\begin{aligned} p(m_1, m_2, \ldots, m_N) &= p(m_1)p(m_2|m_1)\ldots p(m_N|m_1, \ldots, m_{N-1}) \\ &= \frac{\Gamma(\beta_m)}{\Gamma(N+\beta_m)}\alpha^T \prod_{i=1}^{T} P_m(m_i) \prod_{i=1}^{T} (n_{m_i} - 1)! \end{aligned} \tag{5.10}$$

where T denotes the number of suffix types and n_{m_i} is the number of stem types m_i which have been already generated.

Following the joint probability distribution of stems, the conditional probability of a stem given previously generated stems can be derived:

$$p(s_i|S^{-s_i}, \beta_s, P_s) = \begin{cases} \frac{n_{s_i}^{S^{-s_i}}}{L-1+\beta_s} & \text{if } s_i \in S^{-s_i} \\ \frac{\beta_s * P_s(s_i)}{L-1+\beta_s} & \text{else} \end{cases} \tag{5.11}$$

where $n_{s_i}^{S^{-s_i}}$ denotes the number of stem instances s_i that have been generated previously where S^{-s_i} is the set of stems excluding the new instance of the stem s_i.

The conditional probability of a suffix given the other suffixes that have been generated previously is defined similarly:

$$p(m_i|M^{-m_i}, \beta_m, P_m) = \begin{cases} \frac{n_{m_i}^{M^{-m_i}}}{N-1+\beta_m} & \text{if } m_i \in M^{-m_i} \\ \frac{\beta_m * P_m(m_i)}{N-1+\beta_m} & \text{else} \end{cases} \tag{5.12}$$

where $n_{m_i}^{M^{-m_i}}$ is the number of instances m_i that have been generated previously where M^{-m_i} is the set of suffixes excluding the new instance of the suffix m_i.

It should be noted that Equation 5.11 and Equation 5.12 create a Chinese restaurant process for stems and suffixes. The process naturally overcomes the sparsity of data without any need for smoothing.

5.4.2 Embedding Morphology into the Hierarchical Model

Hitherto, the hierarchical model and the morphological segmentation model have been described separately. Here, we combine these two models within the prob-

abilistic hierarchical clustering scheme. As mentioned earlier, each cluster is a probabilistic model. The morphological segmentation model will be embedded into the hierarchical model as a probabilistic model. If the segmentation model is embedded in the hierarchical tree, the marginal likelihood of words $\boldsymbol{D_k} = \{w_1 = s_1 + m_1, \ldots, w_n = s_n + m_n\}$ in cluster k is defined such that:

$$
\begin{aligned}
p(D_k) &= p(S_k)p(M_k) \\
&= \int p(G_s) \prod_{i=1}^{L} p(s_i|G_s) \int p(G_m) \prod_{j=1}^{N} p(m_i|G_m) \\
&= \frac{\Gamma(\beta_s)}{\Gamma(L+\beta_s)} \beta_s^K \prod_{i=1}^{K} G_s(s_i) \prod_{i=1}^{K} (n_{s_i^k} - 1)! \qquad (5.13) \\
&\quad \frac{\Gamma(\beta_m)}{\Gamma(N+\beta_m)} \beta_m^T \prod_{i=1}^{T} G_m(m_i) \prod_{i=1}^{T} (n_{m_i^k} - 1)! \qquad (5.14)
\end{aligned}
$$

where $n_{s_i^k}$ denotes the number of stem instances s_i in cluster k, whereas $n_{m_i^k}$ is the number of suffix instances m_i in cluster k. Equation represents two distinct Chinese restaurant processes for stems and suffixes that will generate words in a cluster.

Words in each cluster represents a paradigm that consists of stems and suffixes. In the hierarchical structure, the model locates words sharing the same stems or suffixes close to each other in the tree. Hence, paradigms are formed naturally in the tree. Each word is seen in all the paradigms on the path from the leaf node having that word till the root. The word can share either its stem or suffix with other words in the same paradigm. By means of this feature, infinitely many words can be generated through a paradigmatic approach that may not be even in the corpus.

Example 5.4.2. Let a corpus be $D = \{quickly, recentness, slow, quickness\}$. Let a paradigm be $P = \{quick, recent, slow\}\{\emptyset, ly, ness\}$. Naturally, the paradigm has the ability to generate the words *quick, recently, recent* although they do not appear in the corpus.

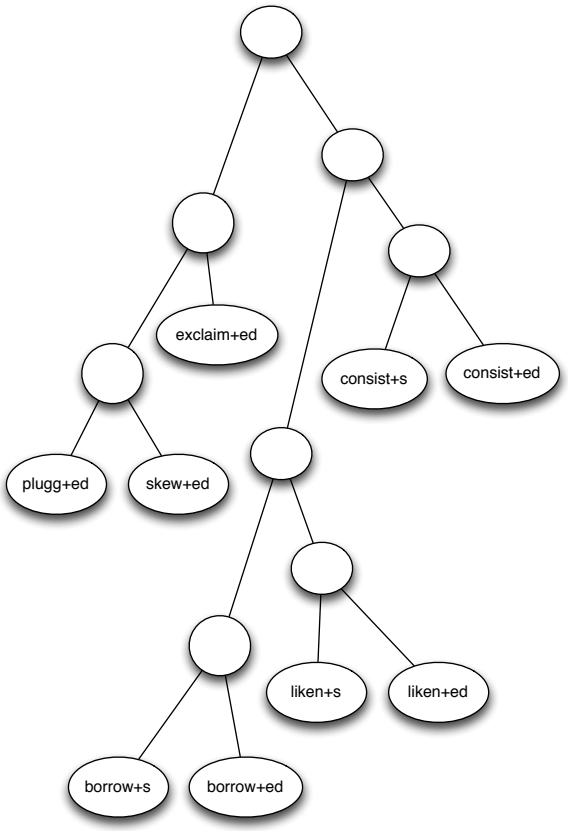


Figure 5.4: A portion of a tree where leaf nodes keep the words, and the internal nodes correspond to paradigms.

A portion of a tree is given in Figure 5.4. As seen in figure, all words are located in leaf nodes. Rest of the nodes correspond to paradigms that consist of words below that node.

5.4.3 Inference

Morphological segmentation of a corpus and the hierarchical structure that represents the morphological segmentation are learned through an inference procedure which forms the probabilistic hierarchical clustering scheme. As mentioned earlier, many traditional clustering algorithms are single-pass and do not suggest any alternative hierarchical structures, once the tree is learned. The reason is

Algorithm 3 Creating initial tree.

input: data $D = \{w_1 = s_1 + m_1, \ldots, w_n = s_n + m_n\}$,
initialise: $root \leftarrow D_1$ where $D_1 = \{w_1 = s_1 + m_1\}$
initialise: $c \leftarrow n - 1$
while $c >= 1$ **do**
    Draw a word w_j from the corpus.
    Split the word randomly such that $w_j = s_j + m_j$
    Create a new node D_j where $D_j = \{w_j = s_j + m_j\}$
    Choose a node D_k on the tree randomly to make it a sibling node to D_j
    Merge $D_{new} \leftarrow D_j \bigcup D_k$
    Remove w_j from the corpus
end while
output: Initial tree

that the data is explicit, and no latent variables are to be inferred. However, for either improving the final clustering structure or for learning latent variables, an inference step is compulsory.

The initial tree is constructed uniformly by adding each word from the corpus into a randomly chosen position on the tree. While choosing an arbitrary position on the tree, latent variables are assigned randomly as well; i.e. words are split at a random position (given in Algorithm 3).

Once the initial tree is built randomly, sampling is performed by relocating nodes on the tree. Iteratively, a leaf node $D_i = \{w_i = s_i + m_i\}$ is drawn from the current tree structure. The drawn leaf node is removed from the tree causing its parent node to be removed as well to adjust the tree to maintain the binary structure. Once the leaf node is removed from the tree, a node D_k is drawn uniformly from the tree to make it a sibling node to D_i. In addition to a sibling node, a split point $w_i = s_i^{'} + m_i^{'}$ is drawn uniformly. Having the tree position, and the split point, the node $D_i = \{w_i = s_i^{'} + m_i^{'}\}$ is inserted as a sibling node to D_k. After updating all the probabilities along the path to the root, the new tree structure is either accepted or rejected. The marginal likelihood of the entire tree is used for the sampling probability. Hence, the sampled tree structure is accepted with a probability of:

$$P_{Acc} = \frac{p_{next}(D|T)}{p_{cur}(D|T)} \tag{5.15}$$

where $p_{next}(D|T)$ denotes the marginal likelihood of data under the new tree structure, and $p_{cur}(D|T)$ denotes the marginal likelihood of data under the latest accepted tree structure.

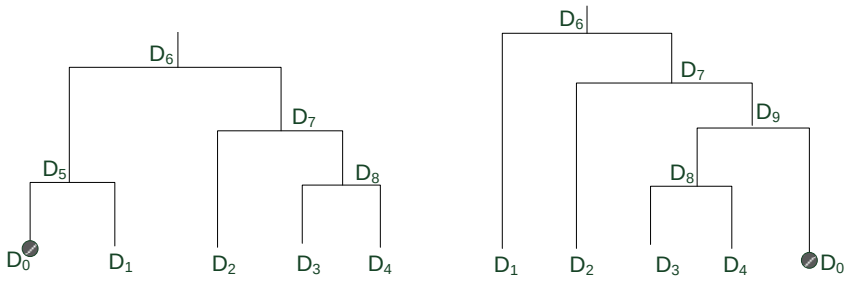
If the condition in Equation 5.15 is not met, the new tree structure is still accepted with a probability of P_{Acc}; otherwise, the new structure is rejected. The inference algorithm is accompanied by simulated annealing. A temperature γ is determined for the initial temperature of the system where all probabilities are raised to the power of the inverse of the system temperature such that:

$$P_{Acc} = \left(\frac{p_{next}(D|T)}{p_{cur}(D|T)}\right)^{\frac{1}{\gamma}} \tag{5.16}$$

The system temperature is reduced in each iteration of the inference algorithm. Most tree structures are accepted in the earlier stages of the algorithm, as if tree structures are drawn from a distribution closer to uniform. However, as the temperature becomes lower, tree structures which lead to a considerable improvement in the marginal probability $p(D|T)$ are accepted.

An illustration is given in Fig.5.5 that depicts an example for suggesting a new tree structure. In the illustration, D_0 is drawn to be removed from the tree. Once the leaf node is removed from the tree, as the parent node D_5 will consist of only one child, the parent node is removed from the tree. The node D_8 is drawn as a sibling node to D_0. Subsequently, the two nodes are merged within a new cluster that introduces a new node D_9. The inference algorithm is given in Algorithm 4.

It should be noted that while the tree structure changes, some nodes disappear; some nodes accept or deduct words; and new nodes are introduced. Due to these changes, marginal likelihoods of affected nodes should be updated in each iteration of the inference algorithm. Due to the updating process in each iteration of the inference algorithm, it is not computationally feasible to train on a large corpus. The largest data set we trained has 22K words (see Section 6.6).



(a) D_0 will be removed from the tree. (b) D_8 is sampled to be the sibling of D_0.

Figure 5.5: Sampling new tree structures. a) Before sampling a new position for the node D_0. b) After inserting the node D_0 as a sibling node to D_8

5.4.4 Morphology Segmentation

Once the optimal tree structure is inferred along with the morphological segmentation of words, any novel word can be analysed. For the segmentation of novel words, the root node is used as it contains all stems and suffixes which are already extracted from the training data. The split point yielding the maximum probability given inferred stems and suffixes is chosen to be the final analysis of the word:

$$\arg\max_{\mathbf{j}} p(w_i = s_j + m_j | D_{root}, \beta_m, P_m, \beta_s, P_s) \tag{5.17}$$

where D_{root} refers to the root of the entire tree.

Here, the probability of a segmentation of a given word given D_{root} is calculated as given below:

$$p(w_i = s_j + m_j | D_{root}, \beta_m, P_m, \beta_s, P_s) = p(s_j | S_{root}, \beta_s, P_s)\ p(m_j | M_{root}, \beta_m, P_m) \tag{5.18}$$

where S_{root} denotes all the stems in D_{root} and M_{root} denotes all the suffixes in

Algorithm 4 Inference algorithm

input: data $D = \{w_1 = s_1 + m_1, \ldots, w_1 = s_1 + m_1\}$, initial tree T, initial temperature of the system γ, the target temperature of the system κ, temperature decrement η
initialise: $i \leftarrow 1$, $w \leftarrow w_i = s_i + m_i$, $p_{cur}(D|T) \leftarrow p(D|T)$
while $\gamma > \kappa$ **do**
    Remove the leaf node D_i that has the word $w_i = s_i + m_i$
    Draw a split point for the word such that $w_i = s_i^{'} + m_i^{'}$
    Draw a sibling node D_j
    $D_m \leftarrow D_i \bigcup D_j$
    Update $p_{next}(D|T)$
    if $p_{next}(D|T) >= p_{cur}(D|T)$ **then**
        Accept the new tree structure
        $p_{cur}(D|T) \leftarrow p_{next}(D|T)$
    else
        $random \sim Normal(0, 1)$
        if $random < \left(\frac{p_{next}(D|T)}{p_{cur}(D|T)}\right)^{\frac{1}{\gamma}}$ **then**
            Accept the new tree structure
            $p_{cur}(D|T) \leftarrow p_{next}(D|T)$
        else
            Reject the new tree structure
            Re-insert the node D_i at its previous position with the previous split point
        end if
    end if
    $w \leftarrow w_{i+1} = s_{i+1} + m_{i+1}$
    $\gamma \leftarrow \gamma - \eta$
end while
output: A tree structure where each node corresponds to a paradigm.

D_{root}. Here $p(s_j|S_{root}, \beta_s, P_s)$ is calculated as given below:

$$p(s_i|S_{root}, \beta_s, P_s) = \begin{cases} \frac{n_{s_i}^{S_{root}}}{L+\beta_s} & \text{if } s_i \in S_{root} \\ \frac{\beta_s * P_s(s_i)}{L+\beta_s} & \text{otherwise} \end{cases} \tag{5.19}$$

Similarly, $p(m_j|M_{root}, \beta_m, P_m)$ is calculated as:

$$p(m_i|M_{root},\beta_m,P_m) = \begin{cases} \frac{n_{m_i}^{M_{root}}}{N+\beta_m} & \text{if } m_i \in M_{root} \\ \frac{\beta_m * P_m(m_i)}{N+\beta_m} & \text{otherwise} \end{cases} \quad (5.20)$$

5.5 Experiments

Two sets of experiments are performed to test the proposed model in this chapter. In the first set of experiments, words are split at single point that segments each word into a single stem and a single suffix. In the second set of experiments, multiple split points are allowed during segmentation by splitting each stem and suffix once more if necessary. In all experiments, words are assumed to be made of stems and suffixes where prefixes and other morphological forms are neglected.

In both set of experiments, Morpho Challenge datasets (Kurimo et al. 2011b) were used. We have performed the experiments for three different languages: English, German, and Turkish where the datasets consist of 878034, 2338323, and 617298 words respectively. Although, all datasets provide word frequencies, we did not use any frequency information. Since it is a nonparametric probabilistic model, the model is supposed to learn the underlying distributions of latent variables without any frequency information. Word frequencies are only used for choosing words to insert on the tree; i.e. words with a frequency more than 200 are chosen to construct the tree. The exceptions of the frequency threshold are mentioned separately if any other value is assigned.

For each experiment, a tree is constructed with a number of words. The number is assigned manually. Once the tree is learned by the inference algorithm, the final tree is used for the segmentation of complete datasets. Several experiments are performed for each language with various settings where the setting varies with the tree size and the model parameters. Model parameters are the concentration parameters $\beta = \{\beta_s, \beta_m\}$ of the Dirichlet processes for stems and suffixes.

In all experiments, the initial temperature of the system is set $\gamma = 2$ where it is reduced to a temperature of $\gamma = 0.01$ with decrements $\eta = 0.0001$. It is depicted in Figure 5.6 how the log likelihoods of the trees of size 10K, 16K, and

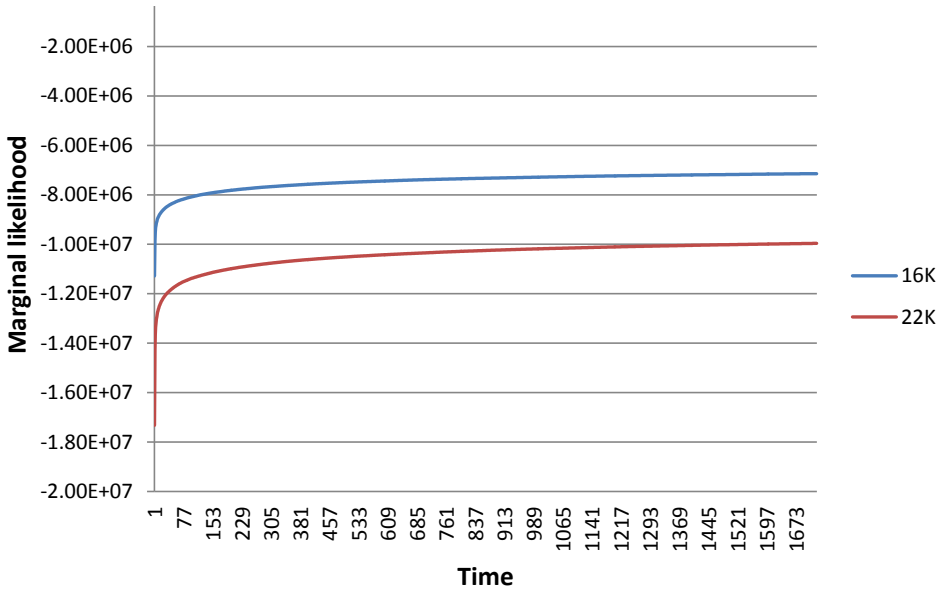


Figure 5.6: Marginal likelihood convergence in time for datasets of size 16K and 22K.

22K converge in time.

As explained earlier, trees are constructed by choosing words from the corpus randomly considering the frequency threshold defined. Since different training sets will lead different tree structures, each experiment is repeated three times keeping the experiment setting the same. Evaluation scores are presented as intervals where only the highest and lowest scores are given.

5.5.1 Experiments with Single Split Points

In this set of experiments, words are split into a single stem and a single suffix. During the segmentation, Equation 5.21 is used to determine the split position for each word.

The evaluation scores regarding the experiments with a training set of 10K words is given in Table 5.1. The maximum score is obtained with $\beta_s = 0.1$ and $\beta_m = 0.1$ which is $\%47.01$.

Some sample tree nodes obtained from trees with 10K words are given in Table 5.2. As seen from the table, morphologically similar words are grouped

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	79.87..84.42	31.20..32.51	45.20..46.21
0.002 0.002	76.25..79.27	30.61..31.38	43.68..44.96
0.01 0.005	77.17..86.08	29.76..31.85	44.26..45.09
0.01 0.01	75.66..80.86	29.71..31.58	42.66..45.42
0.02 0.02	75.66..84.21	30.65..32.46	44.61..45.90
0.1 0.1	78.95..81.48	29.59..33.03	43.05..**47.01**
0.2 0.2	74.07..82.10	32.07..32.72	44.70..46.84

Table 5.1: Evaluation scores of single split point experiments obtained from the trees with 10K words.

together. The morphological similarity both refers to the similarity in terms of the endings and the stems of words. For example, words *second+hand* and *third+largest* are grouped in the same node through the word *second+largest* which shares the same stem with *second+hand* and same ending with *third+largest*.

Remark. For each set of experiments, some sample tree nodes are demonstrated to give an idea about how the tree nodes are organised generally. The discussion about the node contents are although given separately for each set of experiments, as the algorithm is the same, similar rules are to be met in each tree structure. However, as the training set is to change in each experiment, words are expected to be different each time.

Another set of experiments has been performed with 16K words. The results are demonstrated in Table 5.3. The maximum F-measure obtained is $\%50.02$ with the setting of concentration parameters as $\beta_s = 0.001$ and $\beta_m = 0.001$. If we compare the scores of the experiments with 10K and 16K words, it is noticeable that scores are slightly higher with the larger dataset.

Some sample tree nodes obtained from the trees with 16K words is given in Table 5.4. As seen from the sample nodes, prefixes can also be segmented although they are identified as roots; such that *anti+fraud, anti+war, anti+tank, anti+nuclear*. This gives a flexibility in the model by capturing the similarities through either stems, suffixes, or prefixes. However, as mentioned before, the model does not consider any discrimination between different types of morphological forms during training. As the prefix *pre-* appears at the beginning of

scrambl+ed, scrambl+e, scrambl+ing, transferr+ed, influenc+ing, influenc+ed, plung+ed
downgrade+s, crash+ed, crash+ing, lack+ing, blind+ing, blind+, crash+, stifl+e, compris+ing, compris+es, stifl+ing, compris+ed, lack+s, assist+ing, blind+ed, blind+er,
third+-year, booth+, second+-largest, reign+ed, interpret+er, syndicat+ion, echo+ed, second+-round, second+ly, syndicat+ed, third+ly, second+-hand, interpret+ed, third+-largest, booth+s, reign+s, second+-best
near+ing, slight+est, deliberat+ing, slight+ly, afflict+ing, near+ly, downsiz+ing
foot+age, foot+ing, foot+note, foot+-tall, slight
commut+ers, bondhold+ers, steel+ers, wrestl+ers, rul+ers
anti-govern+ment, advance+ment, embezzle+ment, punish+ment
pragmat+ism, robot+ic, euphem+ism, pragmat+ic, marx+ism, popul+ism, academ+ic
hen+ley, mcsor+ley, heff+ley, wheat+ley, tit+ley, mose+ley
clijst+ers, kais+er, helicopt+ers, kilomet+res, kilomet+er, teenag+er, kilomet+ers, mak+eshift, mak+ers

Table 5.2: Some tree nodes obtained from the trees with 10K words.

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	78.75..86.84	31.84..35.92	46.45..**50.02**
0.002 0.002	76.25..81.17	30.61..34.25	43.68..48.17
0.01 0.01	78.75..85.00	31.73..34.27	45.45..48.84
0.02 0.02	79.61..83.75	32.31..34.41	46.40..48.78
0.1 0.1	80.82..86.36	29.87..34.32	44.39..48.18
0.2 0.2	71.25..86.36	32.07..33.65	45.71..47.07

Table 5.3: Evaluation scores of single split point experiments obtained from the trees with 16K words.

words, it is identified as a stem. However, identifying *pre-* as a stem does not yield to a change in the morphological analysis of the word.

Sometimes similarities may not yield a valid analysis of words. For example, the prefix *pre-* lead the words *pre+mise, pre+sumed, pre+gnant, pre+cincts* to be analysed wrongly whereas *pre-* is a valid prefix for the word *pre+face* (see Table 5.4). Another type of wrong analysis arises with common endings in English as mentioned in the previous set of experiments with 10K words. For ex-

regard+less, base+less, shame+less, bound+less, harm+less, regard+ed, relent+less
jave+lin, crome+lin, krem+lin, medel+lin, mer+lin
solve+d, high+-priced, lower+s, lower+-level, high+-level, lower+-income, histor+ians
imit+ation, sens+ation, acceler+ation, beginning+, liquid+ation, spill+s, spill+ed, beginning+s, privatis+ation
pre+mise, pre+face, pre+sumed, pre+, pre+cincts, pre+gnant
base+ment, ail+ment, over+looked, predica+ment, deploy+ment, compart+ment, embodi+ment
anti+-fraud, anti+-war, anti+-tank, anti+-nuclear, anti+-terrorism, switzer+, anti+gua, switzer+land
sharp+ened, strength+s, tight+ened, strength+ened, black+ened
reduc+es, trac+es, lifestyl+es, trac+tors, subcontrac+tors, phras+es, muffin+, accomplic+es, tamal+es, illness+es, nois+es, edn+ey, institut+es, trac+ey, bon+dage, muffin+s, bon+es,
purc+ell, coldw+ell, grenf+ell, sew+ell, ferr+ell, sew+age, orw+ell, caldw+ell

Table 5.4: Some tree nodes obtained from the trees with 16K words.

ample, in Table 5.4, the words *krem+lin, mer+lin, jave+lin, coldw+ell, sew+ell, orw+ell* are analysed wrongly due to a number of words ending with *-lin* and *ell*. On the other side, the model can easily capture the common suffixes such that *-less, -s, -ed, -ment* etc.

The final set of experiments have been performed with a training set of 22K words. The maximum F-score acquired is $\%51.28$ with the concentration parameters $\beta_s = 0.002$ and $\beta_s = 0.002$ (see Table 5.5). When the evaluation scores are compared with the first two sets of experiments' scores, it is noticeable that scores are rather higher with the largest training set. Although, the highest scores of the experiments with 16K and 22K words are not very far from each other, the overall scores are much higher with 22K words.

Sample tree nodes obtained from the trees with 22K words are presented in Table 5.6. Similar features apply here as well; such that similar stems and endings tend to be grouped together. Another nice feature about the model is that compounds are easily captured through common stems; e.g. *doubt+fire, bon+fire, gun+fire, clear+cut*.

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	82.50..84.62	35.69..36.02	50.15..50.33
0.002 0.002	83.55..89.04	34.66..36.01	49.16..**51.28**
0.01 0.01	81.51..87.18	33.49..35.48	47.85..50.43
0.02 0.02	80.86..85.00	34.60..36.16	48.47..50.36
0.1 0.1	80.67..84.18	33.45..35.36	47.70..49.17
0.2 0.2	80.26..87.50	34.52..35.59	48.28..50.50

Table 5.5: Evaluation scores of single split point experiments obtained from the trees with 22K words.

doubt+fire, bon+fire, stain+less, doubt+less, gun+fire
close+s, close+ness, close+ly
investiga+tor, defini+te, investiga+te, determin+ation, determin+ing, defini+tively, determin+ed, admir+ation, determin+es, revolv+ed, investiga+tion, revolv+ing
symbol+ic, hak+e, symbol+s, unit+s, admir+ed, hak+im, aesthetic+s, kind+er, sal+im, taci+t, sal+monella, unit+ed's, admir+ers, admir+e, kind+s, taci+s, kind+red, aesthetic+
inspir+e, inspir+ing, inspir+ed, inspir+es, earn+ing, ponder+ing
stok+es, utiliz+ing, utiliz+e, utiliz+ed
group+s, group+ing, interview+, interview+ing, account+ing, account+, group+
brig+ade, borrow+ings, fac+ade, jan+ice, earn+ed, earn+, appeal+, appeal+s, dispens+ing, appeal+ing, interview+ed, dispens+ed, interview+ers, co-ordinat+ion, co-ordinat+ed, earn+s, jan+, interview+s, fac+ed, jan+ata, appeal+ed
aid+es, inspir+ation, doubt+, prob+es, doubt+ful, doubt+s, prob+e, doubt+ed, aid+ed, aid+e, emancip+ation, prob+ation
clear+est, clear+, clear+-cut, clear+ance

Table 5.6: Some tree nodes obtained from the trees with 22K words.

5.5.2 Experiments with Multiple Split Points

The initial model that we propose can only find one split point for each word. Recall that each word is segmented at the split point which provides the maximum conditional probability given the analyses of all words in corpus, such that:

$$\arg\max_{\mathbf{j}} p(w_i = s_j + m_j | D_{root}, \beta_m, P_m, \beta_s, P_s) \tag{5.21}$$

To discover more split points, we propose a hierarchical segmentation where each segment, which is identified in the first split as a stem+suffix combination, is split further. Here, we postulate that words cannot have more than two stems, and always suffixes follow stems (not allowing any circumfixing or infixing).

In the first split, each word is analysed into two segments $[s_1][m_1]$ that yields the maximum probability according to the Equation 5.21. In the second split, we analyse each segment further. There are 4 possible analyses of the word: [sm][mm], [ss][mm], [s][sm], and [s][mm]. Therefore, the first segment s_1 can be analysed as either [s][m] or [s][s]. The decision to choose which segmentation is made as follows:

$$s_1 \leftarrow s \begin{cases} s \text{ if } p(s|S,\beta_s,P_s) > p(m|M,\beta_m,P_m) \\ m \text{ otherwise} \end{cases} \tag{5.22}$$

where S and M denote the stem and suffix lexicons captured within the tree structure. The second segment can be analysed as [m][m], where the first segment is analysed as either [s][m] or [s][s]. Therefore, the first split yields two different analyses: [sm][mm] and [ss][mm].

Another possibility is that the first segment cannot be split further and left as a stem [s]. In this case, the second segment can be analysed as [s][m] or [m][m]. The decision to choose which segmentation is made as follows:

$$m_2 \leftarrow \begin{cases} s \text{ if } p(s|S,\beta_s,P_s) > p(m|M,\beta_m,P_m) \\ m \text{ else} \end{cases} \quad m \tag{5.23}$$

Thus the second split yields two different analyses in the second case: [s][sm] and [s][mm].

An example for finding multiple split points is given in Figure 5.7. The word *housekeeper* is analysed as [s][sm].

These changes do only affect the segmentation step in the algorithm keeping the inference step the same as before. Therefore, in all experiments with multiple split points, trees are generated following the same MCMC sampling process.

The first set of experiments with multiple split points has been performed by constructing a tree with 10K words (see Table 5.7). The maximum F-measure

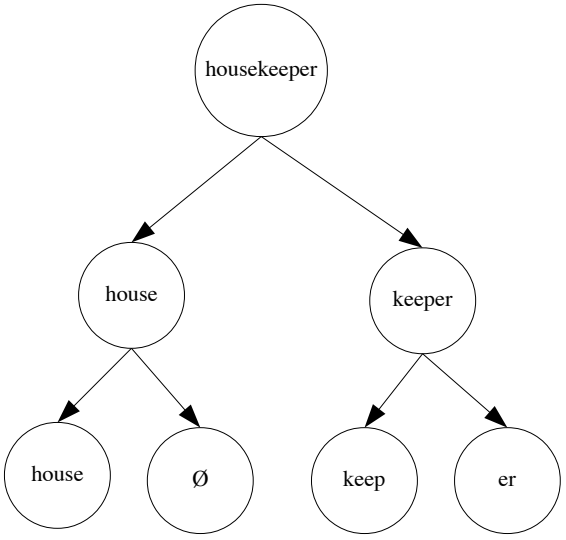


Figure 5.7: An example that depicts how the word *housekeeper* can be analysed further to find more split points.

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	58.59..62.37	52.06..53.95	55.13..58.14
0.002 0.002	56.77..62.02	49.97..54.97	53.83..58.28
0.01 0.01	56.16..64.39	51.87..53.68	53.93..58.09
0.02 0.02	57.02..67.64	51.05..53.77	53.87 ..59.17
0.1 0.1	61.21..63.39	57.42..54.73	58.74..**59.98**
0.2 0.2	59.91..61.70	55.33..57.30	57.53..59.42

Table 5.7: Evaluation scores of multiple split point experiments obtained from the trees with 10K words.

obtained is $\%59.98$. The second set of experiments has been performed with a tree of size 16K words (see Table 5.8). The maximum F-measure obtained is $\%62.36$. Final set of experiments has been performed on a dataset of size 22K (see Table 5.9) where we obtained the highest F-measure of $\%62.56$.

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	58.93..64.89	53.20..53.99	55.92..56.72
0.002 0.002	56.92..67.80	55.54..57.72	56.92..**62.36**
0.01 0.01	62.10..64.76	53.05..55.16	57.22..59.58
0.02 0.02	64.12..65.95	55.42..56.04	60.23..60.48
0.1 0.1	61.27..63.23	54.93..57.59	57.40..59.37
0.2 0.2	62.20 ..65.82	56.60..59.70	59.21..61.39

Table 5.8: Evaluation scores of single split point experiments obtained from a tree with 16K words.

β_s, β_m	Precision(%)	Recall(%)	F-measure(%)
0.001 0.001	64.81..68.71	55.64..57.42	60.68..**62.56**
0.002 0.002	63.11..67.18	56.94..58.82	60.89..61.64
0.01 0.01	60.11..67.81	55.82..57.61	57.89..62.30
0.02 0.02	64.11..66.58	56.46..56.81	60.24..61.10
0.1 0.1	62.29..65.28	55.09..58.97	58.86..60.58
0.2 0.2	60.88..66.18	56.07..58.28	58.73..61.15

Table 5.9: Evaluation scores of single split point experiments obtained from a tree with 22K words.

5.5.3 Comparison with Other Systems

We compare our results with other unsupervised systems participated in Morpho Challenge 2010 (Kurimo et al. 2010). In order to compare the system with the other systems, first we used a separate development set to find the best parameter values. Then, we applied this model on the actual evaluation data. Given comparison between the post-evaluation results and the original evaluation results is not completely fair.

The highest F-measure that was obtained in Morpho Challenge 2010 was the base inference algorithm of Lignos (2010). We also compare our model with one of the state of art systems in unsupervised morphology learning, Morfessor Baseline (Creutz & Lagus 2002, 2005b, 2007) and Morfessor CATMAP. Our model outperforms both members of Morfessor family with the multiple split setting. These scores are scientifically significant since they were obtained as a result of several experiments, where the results were all evaluated by the Morpho

System	Precision(%)	Recall(%)	F-measure(%)
Prob.Clustering (single)	70.76	36.51	48.17
Prob.Clustering (multiple)	57.08	57.58	57.33
Morf. Baseline (Creutz & Lagus 2002)	81.39	41.70	55.14
Morf. CatMAP (Creutz & Lagus 2005a)	86.84	30.03	44.63
Base Inference (Lignos 2010)	80.77	53.76	**64.55**
Iterative Comp. (Lignos 2010)	80.27	52.76	63.67
Aggressive Comp. (Lignos 2010)	71.45	52.31	60.40
Nicolas (Nicolas et al. 2010)	67.83	53.43	59.78

Table 5.10: Comparison of our model with other unsupervised systems participated in Morpho Challenge 2010 for English.

Challenge organisers.

Remark. It should be noted that we only show the unsupervised systems participated in Morpho Challenge 2010 in the table since the supervised systems use training sets provided by the Morpho Challenge to tune parameters of their systems. Therefore their systems are more likely to be overfitted.

The official results of the Morpho Challenge 2010 are presented in Table 5.10 (Virpioja et al. 2011). Since the development sets that are used for the official evaluation differ from the publicly available sets (Kurimo et al. 2011b), the evaluation scores may differ from the ones presented in authors' published papers. Here, since our model is also evaluated with the official development sets, the scores may differ from the ones that we presented in previous sections.

We also performed experiments with Morpho Challenge 2009 English dataset. The dataset consists of 384904 words. Our results and other participant systems' results are given in Table 5.11 (Kurimo et al. 2009). As seen from the table, our system comes 5th out of 16 systems.

5.5.4 Experiments in Various Languages

We also provide results for German and Turkish datasets using the Morpho Challenge 2010 (Kurimo et al. 2010) datasets. As both languages are morphologically rich (German is a compound language and Turkish is an agglutinative language),

System	Precision(%)	Recall(%)	F-measure(%)
Allomorf (Virpioja et al. 2009)	68.98	56.82	62.31
Morf. Base. (Creutz & Lagus 2002)	74.93	49.81	59.84
PM-Union (Christian Monson 2009)	55.68	62.33	58.82
Lignos (Lignos et al. 2009)	83.49	45.00	58.48
Prob. Clustering (multiple)	70.04	59.06	**57.33**
PM-mimic (Christian Monson 2009)	53.13	59.01	55.91
MorphoNet (Bernhard 2009)	65.08	47.82	55.13
Rali-cof (Lavallée & Langlais 2009)	68.32	46.45	55.30
CanMan (Can & Manandhar 2009)	58.52	44.82	50.76
Morf. CatMAP (Creutz & Lagus 2005a)	84.75	35.97	50.50
Prom-1 (Spiegler et al. 2009)	36.20	64.81	46.46
Rali-ana (Lavallée & Langlais 2009)	64.61	33.48	44.10
Prom-2 (Spiegler et al. 2009)	32.24	61.10	42.21
Prom-com (Spiegler et al. 2009)	32.24	61.10	42.21
MetaMorf (Tchoukalov et al. 2009)	68.41	27.55	39.29
Ungrade (Golénia et al. 2009)	28.29	51.74	36.58

Table 5.11: Comparison with other unsupervised systems participated in Morpho Challenge 2009 for English.

we performed only the experiments with multiple split points. We set the concentration parameters as $\beta_s = 0.01$ and $\beta_m = 0.005$, and the tree size as 22K. We set the frequency threshold as 50 for Turkish since the Turkish dataset is not large enough and we set the frequency threshold 200 for German similarly to the previous experiments. Each experiment was repeated three times for both German and Turkish.

The results of the experiments for German is given in Table 5.12. The table also demonstrates other participants' scores in Morpho Challenge 2010.

The results of the experiments for the Turkish dataset is given in Table 5.13 with other participants in Morpho Challenge 2010.

System	Precision(%)	Recall(%)	F-measure(%)
Prob. Clustering (multiple)	57.79	32.42	41.54
Morf. Baseline (Creutz & Lagus 2002)	82.80	19.77	31.92
Morf. CatMAP (Creutz & Lagus 2005a)	72.70	35.43	**47.64**
Base Inference (Lignos 2010)	66.38	35.36	46.14
Iterative Comp. (Lignos 2010)	62.13	34.70	44.53
Aggressive Comp. (Lignos 2010)	59.41	37.21	45.76

Table 5.12: Experiment results with concentration parameters: $\beta_s = 0.01$ and $\beta_m = 0.005$ for German with other participants in Morpho Challenge 2010.

System	Precision(%)	Recall(%)	F-measure(%)
Prob. Clustering (multiple)	72.36	25.81	38.04
Morf. Baseline (Creutz & Lagus 2002)	89.68	17.78	29.67
Morf. CatMAP (Creutz & Lagus 2005a)	79.38	31.88	**45.49**
Base Inference (Lignos 2010)	72.81	16.11	26.38
Iterative Comp. (Lignos 2010)	68.69	21.44	32.68
Aggressive Comp. (Lignos 2010)	55.51	34.36	42.45
Nicolas (Nicolas et al. 2010)	79.02	19.78	31.64

Table 5.13: Experiment results with concentration parameters: $\beta_s = 0.01$ and $\beta_m = 0.005$ for Turkish with other participants in Morpho Challenge 2010.

5.6 Conclusion

In this chapter, we present a novel probabilistic model for unsupervised morphology learning. The model adopts a hierarchical structure where words are organised in a tree in a way that morphologically similar words are located close to each other. We present the results that we have obtained with Morpho Challenge datasets for English, German and Turkish. We also experimented with the Morpho Challenge 2009 datasets in English. Our system outperforms other systems in Morpho Challenge 2009 for English.

For German and Turkish languages, although our model gives lower scores compared to other systems, the model can easily capture the morphological similarity between words. Capturing similarity between words facilitates handling

frequent morphological forms naturally; i.e. compounds. However, German and Turkish are morphologically richer than English, and these languages require more sophisticated methods to handle the concatenation of various morphological forms of words.

CHAPTER 6

Joint Learning of Morphology and POS Tagging

"It takes two to tango."
British idiom

6.1 Introduction

This chapter presents a joint model for learning morphology and POS tags simultaneously. The proposed method adopts a finite mixture model that groups words having similar contextual features thereby assigning the same POS tag to those words. While learning POS tags, words are analysed morphologically by exploiting the morphological features of the learned POS tags.

The chapter is organised as follows: Section 6.2 motivates the research contributed with this chapter; Section 6.3 describes the model mathematically; Section 6.4 explains the inference algorithm; Section 6.3 presents the experiments and their results; and finally Chapter 6.8 concludes with a brief summary of the chapter.

6.2 Motivation

The correlation between morphology and syntax has been of great interest in the field. Most research focus on learning either morphology or POS tags assuming the other is already provided. Chapter 3 reviews the research that can be considered as an instance of a single model that learns only one type of latent variable (either morphology or POS tags). Here we review more literature relevant to the joint learning problem to motivate the research presented in this chapter.

In the recent years, there have been several attempts to learn morphology and syntax cooperatively. Hasan & Ng (2009) propose a model that exploits the trigram model of Goldwater & Griffiths (2007) to improve POS tagging by using suffixes. Words that exist in the word lexicon (which is adopted for learning) are tagged using the original model of Goldwater & Griffiths (2007), whereas words that do not appear in the word lexicon are tagged by using a suffix lexicon where suffixes are emitted from each tag instead of words. Using a suffix lexicon that has the possible tag assignments for each suffix, the authors could use a smaller word lexicon. However, it should be noted that the model is not unsupervised due to the employment of a tagged lexicon. Their model can be considered as weakly supervised since the lexicon size is small compared to the one used in Goldwater & Griffiths (2007).

Lee et al. (2011) exploits context to learn morphology. Their model is inspired by our approach that has been described in Chapter 4. The method that we propose in Chapter 4 considers POS tagging as a separate step, whereas Lee et al. (2011) combine POS tagging and morphology learning within the same learning mechanism. However, the main focus of the study is to learn morphology by benefiting from the context as much as possible. Hence, the contribution can be easily counted as a part of morphology learning rather than a joint learning.

Although in the recent years, there have been attempts to combine morphology learning and POS tagging, none of the existing work has proposed a joint learning process for morphology and POS tags. Although Lee et al. (2011) share a similar goal with us for combining two learning processes, the final aim of their

model is to learn morphology by exploiting the context, Thus, the model can be considered more on the morphology learning side of the field. As a matter of fact, authors do not present any evaluation for POS tagging, but only for morphology learning. It should be also noted that in their experiments, the number of POS tags is determined as 5.

The model that will be described in this chapter brings a new perspective in the field by combining morphology and POS tagging within the same learning mechanism where both are learned simultaneously and without using any tagged lexicon. The model can be considered as a sophisticated version of the algorithm suggested in Chapter 4. However, it should be noted that this model does not adopt a paradigmatic approach like the one in Chapter 4.

6.3 Model Definition

The model proposed in this chapter adopts a Bayesian approach. We will describe the model in two parts: POS tagging and morphology learning. However, it should be noted all the time that the model adopts a joint learning where two parts are learned simultaneously.

6.3.1 POS Tagging

The model adopts a finite mixture model for POS tagging. By definition, a mixture model consists of a set of mixture components. In our model, each mixture component c_i corresponds to a POS tag. Therefore, each POS tag is a mixture component indicator. Each mixture component c_i consists of words and their contexts. Words are denoted by w_i and every word that belongs to c_i is meant to have the respective component indicator as its POS tag. Each context is a tag pair $< c_{i-1}, c_{i+1} >$, where the first tag $< c_{i-1} >$ corresponds to the POS tag of the previous word w_{i-1} and the second tag corresponds to the POS tag of the following word w_{i+1}.

Example 6.3.1. Let a phrase be *'an/*c_1 *interesting/*c_2 *study/*c_3*'*. Here, the word *'interesting/*c_2*'* belongs to the mixture component c_2, which is also its POS tag.

Moreover, *'interesting/c_2'* has the context words $< an/c_1, study/c_3 >$ and its context is defined as $< c_1, c_3 >$.

Mathematical representation of the model is given in terms of underlying distributions as follows:

$$
\begin{aligned}
c_i &\sim Mult(\phi) && (6.1)\\
\phi &\sim Dir(\pi) && (6.2)\\
w_i|c_i &\sim Mult(\theta_w) && (6.3)\\
\theta_w &\sim Dir(\kappa) && (6.4)\\
c_{i-1,i+1}|c_i &\sim Mult(\theta_{c,c'}) && (6.5)\\
\theta_{c,c'} &\sim Dir(\beta) && (6.6)
\end{aligned}
$$

Graphical representation of the model is given in Figure 6.1.

Mixture component indicators c_i are drawn from a Multinomial distribution with parameters ϕ (see Equation 6.1). For the Multinomial parameters, we define a prior distribution which is distributed according to a Dirichlet distribution with hyperparameters π (see Equation 6.2). The reason for defining a Dirichlet distribution is to obtain Multinomial-Dirichlet conjugacy. Multinonial-Dirichlet conjugation shapes the component indicators within a Chinese Restaurant Process where each indicator is distributed proportionally with the number of words in the mixture component.

Since each mixture component consists of a set of words w_i distributed according to a Multinomial distribution with parameters θ_w (see Equation 6.3). Dirichlet distribution with hyperparameters κ is defined as prior information for the Multinomial parameters θ_w (see Equation 6.4).

Each mixture component c_i also consists of a set of contexts $c_{i-1,i+1}$, which are the contexts of the words in c_i. The contexts are distributed according to a Multinomial distribution with parameters $\theta_{c,c'}$ (see Equation 6.5). For the context multinomials, we again define prior information in the form of a Dirichlet distribution with hyperparameters β (see Equation 6.6).

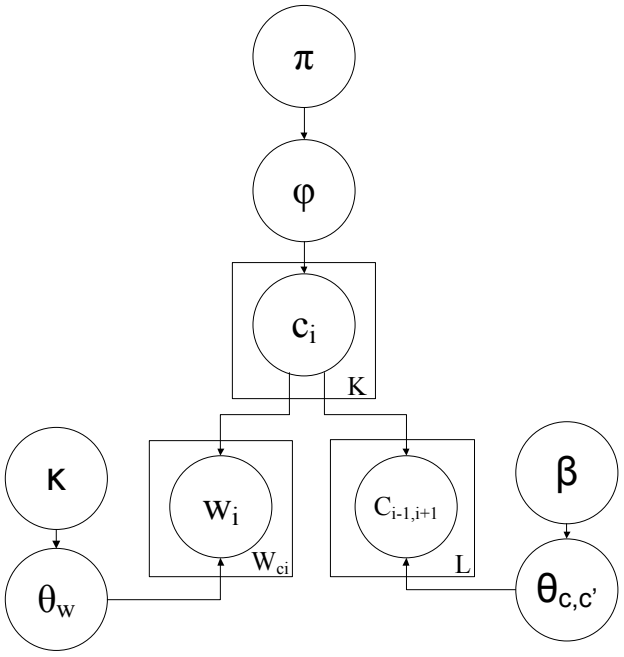


Figure 6.1: Plate diagram of POS tagging part of the model.

Following the model definition, conditional probabilities can be derived to be used for the inference (see Section 6.4). The probability of a class indicator is derived as follows (see Chapter 2 for the derivation):

$$p(c_i|\pi) = \frac{n_{c_i} + \pi}{N + K\pi} \tag{6.7}$$

where n_{c_i} denotes the number of word tokens tagged with c_i, N denotes the number of total word tokens, and K is the number of class indicators. Class indicators with more words are more likely to be assigned to new words with the rich-get-richer principle.

The conditional probability of a context given a tag is derived as follows:

$$p(< c_{i-1}, c_{i+1} > |c_i, \beta) = \frac{n_{c_{i-1},c_i,c_{i+1}} + \beta}{n_{c_i} + L\beta} \tag{6.8}$$

where $n_{c_{i-1},c_i,c_{i+1}}$ denotes the number of contexts $< c_{i-1}, c_{i+1} >$ in the mixture component c_i. The total number of contexts in c_i is denoted by n_{c_i} whereas L denotes the possible number of different contexts in the model. The possible number of different contexts is limited by the number of tags in the model. Maximum number of context types is $K * K$.

Similarly, the conditional probability of a word given a tag is derived as follows:

$$p(w_i|c_i, \kappa) = \frac{n_{w_i,c_i} + \kappa}{n_{c_i} + W_{c_i}\kappa} \tag{6.9}$$

where n_{w_i,c_i} is the number of word-tag pairs $< w_i, c_i >$; n_{c_i} is the number of word tokens having the tag c_i; and W_{c_i} is the number of word types that are tagged with c_i.

Each process can be interpreted as a Chinese Restaurant Process (CRP):

- Let one of the restaurants be a class indicator restaurant where each table has the same meal i.e. c_i. Customers are words that decide which table to sit themselves accordingly with the number of customers already sitting at the table. Thus, each table represents the class indicator with a number of customers having the same tag.

- The second one is a context restaurant chain where each restaurant is a member of a different tag. Each table in each restaurant has the same meal i.e. a context $< c_{i-1}, c_{i+1} >$. Customers are various contexts that decide which table in which restaurant to sit themselves accordingly with the other contexts sitting at the same table. Therefore, same contexts tend to appear more.

- The final restaurant is a word restaurant chain where each restaurant is again a member of a different tag. Each table has the same meal i.e. word w_i. Customers are words. Each word sit themselves at a table accordingly with the number of same words already sitting at the table.

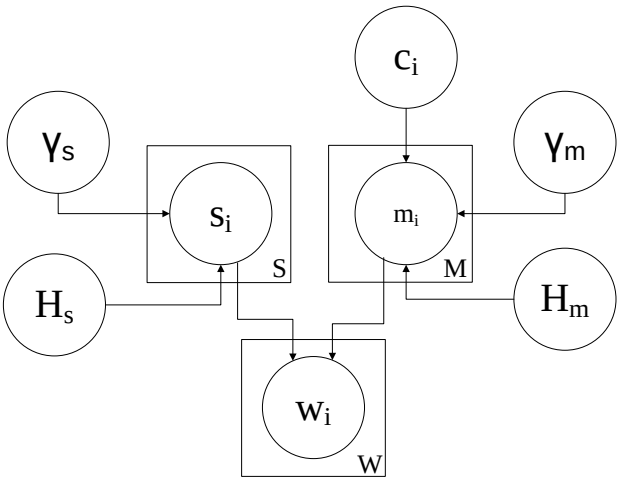


Figure 6.2: Plate diagram of morphology part of the model.

6.3.2 Morphology Learning

Morphology is modelled by Dirichlet processes in the model. The model splits each word into two segments: a stem and a suffix. Stems are generated by a Dirichlet process $DP(\gamma_s, H_s)$ with the concentration parameter γ_s and the base distribution H_s. Analogously, suffixes are generated by another Dirichlet process $DP(\gamma_m, H_m)$ with concentration parameter γ_m and base distribution H_m. Hence, the model is summarised as follows:

$$\begin{aligned} s_i &\sim DP(\gamma_s, H_s) \\ m_i|c_i &\sim DP(\gamma_m, H_m) \end{aligned} \tag{6.10}$$

The plate diagram of the model is given in Figure 6.2.

The reason for using Dirichlet process for morphemes (stems and suffixes) is the nature of a Dirichlet process to generate a Chinese restaurant process. Morphology is established on morphemes which are repeated; therefore, morphemes can be discovered by a rich-get-richer behaviour. While generating a Chinese res-

taurant process, a Drichlet process can adopt a probability distribution to define the features of the items that will be generated by the Dirichlet process. This probability distribution is called the base distribution of the process. We use the base distribution in the Dirichlet processes to embed prior information for the lengths of the morphemes.

The base distribution H_s is an implicit length prior that favours shorter stem lengths (Creutz & Lagus 2005b):

$$p(s_i) = p(c_{ij})^{|s_i|} \tag{6.11}$$

where $|s_i|$ denotes the length of the stem in characters. Each character has a probability of $p(c_{ij})$ where characters are assumed to be distributed uniformly in an alphabet. Thus, a letter in English alphabet will have a probability of $1/26$. We also assume that each morpheme ends with a special character; i.e. end of morpheme marker.

Following the Dirichlet process, while drawing a stem from $DP(\gamma_s, H_s)$ rather than generating a new stem, an existing stem is preferred. If the stem is not generated before, then base distribution forces shorter stems to be preferred. Since longer stems will generate more terms in the base distribution, shorter stems are forced to be generated. Suffixes are generated from $DP(\gamma_m, H_m)$ similarly.

Here, it is important to emphasise that $DP(\gamma_s, H_s)$ is a global Dirichlet process where stems may belong to any POS tag. However, suffixes are generated based on each POS tag locally. The reason for defining the model such that is to enable stems to be shared amongst different tags. However, generally words with the same tag have similar endings leading us to define local distribution for suffixes instead of a global one.

We can derive the conditional probability of stems and suffixes to be used for the inference. The conditional probability of a stem is derived as follows:

$$p(s_i|\mathbf{s}, \gamma_s, H_s) = \frac{n_{s_i} + \gamma_s H_s(s_i)}{N_s + S\gamma_s} \tag{6.12}$$

where n_{s_i} is the count of stem type s_i that is generated previously and N_{s_i} is the number of all stems generated. Finally, S is the number of stem types generated by the model. The conditional probability of a suffix is derived as follows:

$$p(m_i|\mathbf{m}_{c_i}, \gamma_m) = \frac{n_{m_i}^{c_i} + \gamma_m H_m(m_i)}{N_m^{c_i} + M\gamma_m} \tag{6.13}$$

where $n_{m_i}^{c_i}$ is the count of suffix types m_i that are previously generated in c_i, and $N_m^{c_i}$ is the number of all suffixes assigned with tag c_i. Finally M is the number of suffix types generated in the model so far.

6.4 Inference

The full model to be learned, by combining POS tagging and morphological segmentation, is given in Figure 6.3. The model has got both observed and unobserved variables. Observed variables are words w_i and the hyperparameters of the prior distributions: $\pi, \kappa, \beta, \gamma_s, \gamma_m$, which are determined empirically. Unobserved variables are POS tags, stems, suffixes and other parameters: $\phi, \theta_w, \theta_{c,c'}$. However, we do not aim to estimate the parameters $\phi, \theta_w, \theta_{c,c'}$, instead we integrate those parameters out by using the Multinomial-Dirichlet conjugacy. Therefore, we only address to infer POS tags, stems and suffixes as unobserved latent variables.

For the inference of latent variables, we use Gibbs sampling. In Gibbs sampling, POS tags, stems and suffixes are sampled interchangeably. We divide the inference into two steps where first a POS tag is sampled, and then a stem and a suffix are sampled for each word.

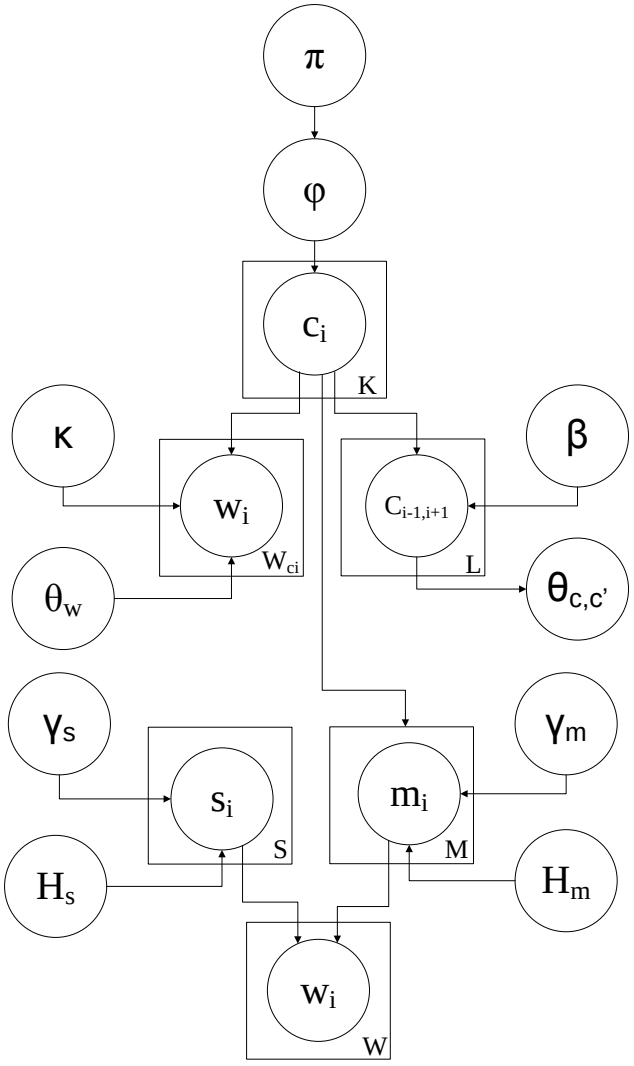


Figure 6.3: The complete joint model.

6.4.1 Inferring POS

Each word's POS tag is sampled subject to its context. Let a word be w_i and imagine that it occurs in context $< w_{i-1}, w_{i+1} >$ where w_{i-1} belongs to c_{i-1} and w_{i+1} belongs to c_{i+1}. The sampling probability of a POS tag c_i for a given word w_i is defined as follows:

$$\begin{aligned} p(c_i| < w_{i-1}, w_{i+1} >, w_i) &\propto p(< w_{i-1}, w_{i+1} >, w_i|c_i)p(c_i) \\ &\propto p(w_i|c_i)p(< w_{i-1}, w_{i+1} > |c_i)p(c_i) \end{aligned} \tag{6.14}$$

Bayes' rule is applied here to be able to calculate the sampling probability in terms of likelihood and prior probability. We omit the denominator in the Bayes' rule that is the triple $< w_{i-1}, w_i, w_{i+1} >$; and it is the same for any c_i. We also assume that $< w_{i-1}, w_{i+1} >$ and w_i are independent since it is possible to remove w_i from $< w_{i-1}, w_{i+1} >$ and insert another word instead. It should be noted that this is only an assumption and it is not completely true in the real world.

Example 6.4.1. Let the sample phrase be *'an interesting study'*. In the context $< an, study >$, it is possible to insert another adjective (e.g. *boring, nice, formal, etc.*) instead of *'interesting'*. This is only true for a set of words, say adjectives. For example, inserting a noun in the given context does not yield a grammatically correct phrase.

In the equation, it is simple to calculate $p(w_i|c_i)$ and $p(c_i)$ using Equation 6.9 and Equation 6.7 respectively. However, both equations require w_i to be removed from the corpus (as a requirement in Gibbs sampling). Therefore, Equation 6.9 can be rewritten for Gibbs sampling:

$$p(w_i|c_{-i}, \kappa) = \frac{n_{w_i,c_{-i}} + \kappa}{n_{c_{-i}} + W_{c_{-i}}\alpha} \tag{6.15}$$

where c_{-i} denotes the mixture component c_i excluding w_i.

Equation 6.7 can be rewritten as follows:

$$p(c_i|c_{-i}, \pi) = \frac{n_{c_{-i}} + \pi}{N_{-i} + K\pi} \tag{6.16}$$

To calculate the context probability in a given mixture component which is denoted by $p(< w_{i-1}, w_{i+1} > |c_i)$ in Equation 6.14, we use an approximation

suggested by Clark (2000). When the actual words are used for a context, it yields to sparsity. Thus, the approximation is applied to eliminate the sparsity. In the approximation, each context ($< w_{i-1}, w_{i+1} >$) is approximated with the tags of the context words such that:

$$\begin{aligned} p(< w_{i-1}, w_{i+1} > |c_i) &= p(< c_{i-1}, c_{i+1} > |c_i) \\ & \quad p(w_{i-1}|c_{i-1})p(w_{i+1}|c_{i+1}) \end{aligned} \tag{6.17}$$

where the approximation is weighted by the probabilities of context words in the respective mixture components: $p(w_{i-1}|c_{i-1})$ and $p(w_{i+1}|c_{i+1})$. Conditional probabilities of words are calculated using Equation 6.15. To calculate $p(< c_{i-1}, c_{i+1} > |c_i)$, Equation 6.8 is rewritten by omitting the respective contexts. Since w_i is the right context word of the previous word, and also left context word of the following word, $< c_{i-2}, c_i >$ and $< c_i, c_{i+2} >$ have to be removed from the respective mixture mixture components, in addition to the context of the current word being sampled: $< c_{i-1}, c_{i+1} >$. Therefore, the final context sampling probability becomes:

$$p(< c_{i-1}, c_{i+1} > |c_i^{-<c_{i-1},c_{i+1}>}, c_{i-1}^{-<c_{i-2},c_i>}, c_{i+1}^{-<c_i,c_{i+2}>}, c_{-i}, \beta) = \frac{n_{c_{i-1},c_i,c_{i+1}} + \beta}{n_{c_{-i}} + L\beta} \tag{6.18}$$

where $c_i^{-<c_{i-1},c_{i+1}>}$ denotes the mixture component c_i that excludes the context $< c_{i-1}, c_{i+1} >$, $c_{i-1}^{-<c_{i-2},c_i>}$ denotes the mixture component c_{i-1} that excludes the context $< c_{i-2}, c_i >$, and $c_{i+1}^{-<c_i,c_{i+2}>}$ denotes the mixture component c_{i+1} that excludes the context $< c_i, c_{i+2} >$.

6.4.2 Inferring Morphology

Subsequent to sampling of the POS tag of a given word, its morphology is sampled. In the morphology, two different latent variables are to be inferred: stems and suffixes. The sampling probability for the morphology is defined as

follows:

$$p(w_i = s_i + m_i|\mathbf{s}_{-i}, \mathbf{m}_{-i}^{c_i}) \quad = \quad p(s_i|\mathbf{s}_{-i})p(m_i|\mathbf{m}_{-i}^{c_i}) \tag{6.19}$$

where $\mathbf{s}_{-i}$ is the stem lexicon that consists of all stems in the model excluding s_i, and $\mathbf{m}_{-i}^{c_i}$ denotes all suffixes assigned with c_i excluding m_i. Equation 6.12 and Equation 6.13 are rewritten by excluding s_i and m_i. Therefore, the conditional probability of a stem becomes:

$$p(s_i|\mathbf{s}_{-i}, \gamma_s, H_s) = \frac{n_{s_{-i}} + \gamma_s H_s(s_i)}{N_{s_{-i}} + S_{-i}\gamma_s} \tag{6.20}$$

where $n_{s_{-i}}$ is the count of the stem type s_i generated previously and $N_{s_{-i}}$ is the number of all stems excluding s_i.

The conditional probability of a suffix is rewritten as follows:

$$p(m_i|\mathbf{m}_{-i}^{c_i}, \gamma_m) = \frac{n_{m-i}^{c_i} + \gamma_m H_m(m_i)}{N_{m_{-i}^{c_i}} + M_{-i}\gamma_m} \tag{6.21}$$

where $n_{m-i}^{c_i}$ is the count of suffixes m_i previously generated, and $N_{m_{-i}^{c_i}}$ is the number of all suffixes assigned with tag c_i that excludes m_i.

6.5 Algorithm

Algorithm starts by assigning random POS tags to each word and splitting randomly. Inference algorithm goes through each word iteratively by sampling a POS tag, a stem and a suffix for the word. Before sampling, all constituents of the respective word (tag, stem, suffix, context, contexts of adjacent words) are removed from the model. After a number of iterations, the distributions from which POS tags and stem-suffix are sampled converge to the target distributions. The inference algorithm is summarised in Algorithm 5.

Algorithm 5 The inference algorithm to infer POS tags and morphology interchangeably.

1: **input:** Corpus $W = \{w_1, \ldots, w_n\}$
2: **initialise:** number of clusters $c \leftarrow T$, $W = \{w_1/c_1, \ldots, w_n/c_n\}$, $W = \{w_1 = s_1 + m_1, \ldots, w_n = s_n + m_n\}$, number of iterations J
3: **for** $j = 1 \rightarrow J$ **do**
4: **for** $i = 1 \rightarrow n$ **do**
5: Sample a POS tag for w_i with the sampling probability:

$$p(c_i| < w_{i-1}, w_{i+1} >, w_i) \tag{6.22}$$

6: Sample a stem and a suffix for w_i with the sampling probability:

$$p(s_i, m_i|s_{-i}, m_{-i}^{c_i}) \tag{6.23}$$

7: **end for**
8: **end for**
9: **output:** POS tags, stems, suffixes of words in W.

6.6 Experiments & Evaluation

To evaluate the proposed model in this chapter, we use Penn WSJ treebank (Marcus et al. 1993) in all experiments. Each experiment is performed on a corpus with a different size which is obtained from the Penn WSJ treebank.

The model is evaluated for both POS tagging and morphology learning individually. For the evaluation of POS tagging, different evaluation methods are applied.

We manually set the hyperparameters and concentration parameters for each experiment; i.e. $\pi = 10^{-6}, \beta = 10^{-6}, \kappa = 10^{-6}, \gamma_s = 10^{-6}, \gamma_m = 10^{-6}$. These values are set as a result of several experiments with different settings.

In all experiments, we leave the punctuation since punctuation helps learning the syntax. We also leave words starting with an upper case character as they are. The reason is that proper nouns usually begin with an upper case character and these words can easily be distinguished from other nouns having the same spelling. As a preprocessing, we only insert a special character for sen-

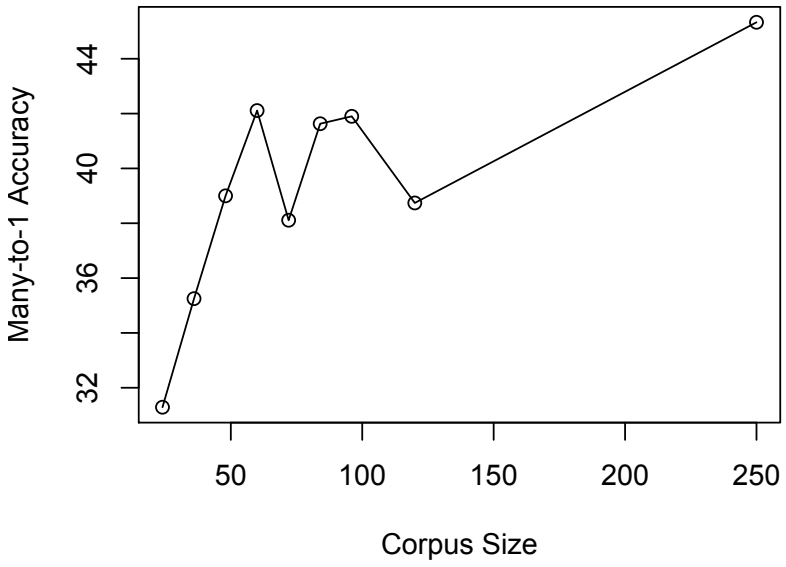


Figure 6.4: Many-to-1 accuracy scores obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K, and 250K.

tence boundaries. The special character for the sentence boundary is assigned a separate POS tag and no other words can be assigned this tag.

For POS tagging, a token-based approach is followed due to the significance of the context for each individual token. Therefore, each token is considered individually during sampling. However, for morphological segmentation, types are considered during sampling. When tokens are considered for morphology, then words with a high frequency are likely to be split since they are recognised as distinct words to be analysed.

We present the results in two sections where POS tagging and morphological segmentation results are demonstrated and discussed separately.

6.6.1 POS Tagging Results

Penn WSJ tree bank has 45 distinct POS tags (see Appendix A). In all of our experiments we constrain the number of POS tags induced by the model as 45 as

well. This helps evaluating the model using the well-known evaluation methods for unsupervised POS tagging. As discussed in Chapter 3, different evaluation methods are suggested for POS tagging. The first evaluation method we apply is many-to-one accuracy. If we recall briefly, in the many-to-one accuracy evaluation, each result tag is assigned to a gold standard tag that has the highest frequency among the words assigned with the result tag. Therefore, it is possible to assign each gold standard tag more than one result tag. We ran several experiments with different size of corpora. Many-to-1 accuracy scores are depicted in Figure 6.4 for various corpora with different size. As it seen on the figure, the accuracy increases with the corpus size. One reason is that in larger corpora there is more context information and less sparsity of words. More context information leads the probability distributions over contexts to reflect the truth more, therefore leading to a higher accuracy.

We also applied one-to-one accuracy. In the one-to-one accuracy evaluation, the number of assignments (goldstandard tag to result tag) is constrained by one where each gold standard tag can be assigned only one result tag. Thus, in one-to-one accuracy there is one-to-one correspondence. We adopt a greedy algorithm where each gold standard tag is assigned a result tag randomly. In each step of the greedy algorithm, the two gold standard tags swap their assigned result tags which lead to the highest increment in the accuracy. The algorithm terminates when there is no more improvement in the accuracy. One-to-one accuracy results are depicted in Figure 6.5 for different size of corpora. The same situation holds for the one-to-one accuracy scores. The larger the datasets are the higher one-to-one accuracy scores are. The final results of the one-to-one accuracy are lower than the state-of-art system (Clark 2003), however, smaller number of tag classes would be more meaningful and would yield much higher scores. On the other hand, our system should not be interpreted solely as a POS tagging system. It is one step towards joint learning and requires more work to improve the scores.

We also measure variation of information (VI) between two clusterings. As explained in Chapter 6, VI measures how much information is lost from one clustering to another. Differently from the previous two evaluation scores (many-to-one and one-to-one), the smaller VI is, the better the result clustering is. Figure 6.6 shows the VI scores for various sizes of corpora. Although there is not a

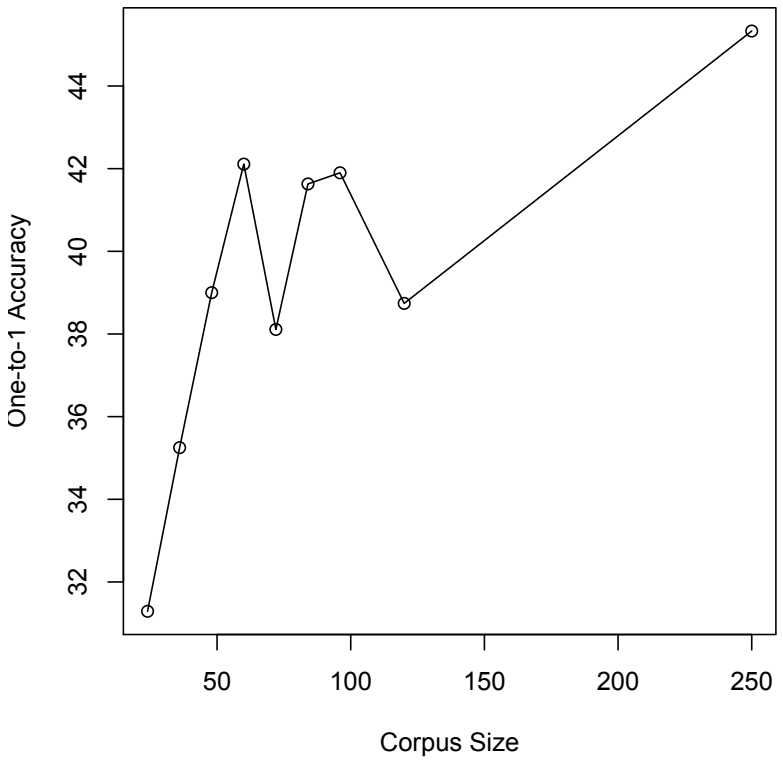


Figure 6.5: One-to-1 accuracy scores obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K and 250K.

smooth decrement in VI measure, it improves with the larger datasets in average. VI score is not consistent with the previous two scores mentioned. For example, we obtain a higher one-to-one accuracy with 48K words than 36 words (Figure 6.5). However, it may not be the case with the VI. Figure 6.6 shows that VI is lower with 36K words than 48 words. VI is more informative than one-to-one accuracy and many-to-one accuracy. For example, two different clusterings may have the same one-to-one accuracy, but may have different VI measures (Goldwater & Griffiths 2007). The clustering in which the erroneous tag assignments are more scattered with different tags would have a higher VI measure.

For each experiment, Gibbs sampler starts to converge in different iterations due to the corpus size. We ran each experiment for a sufficient number of it-

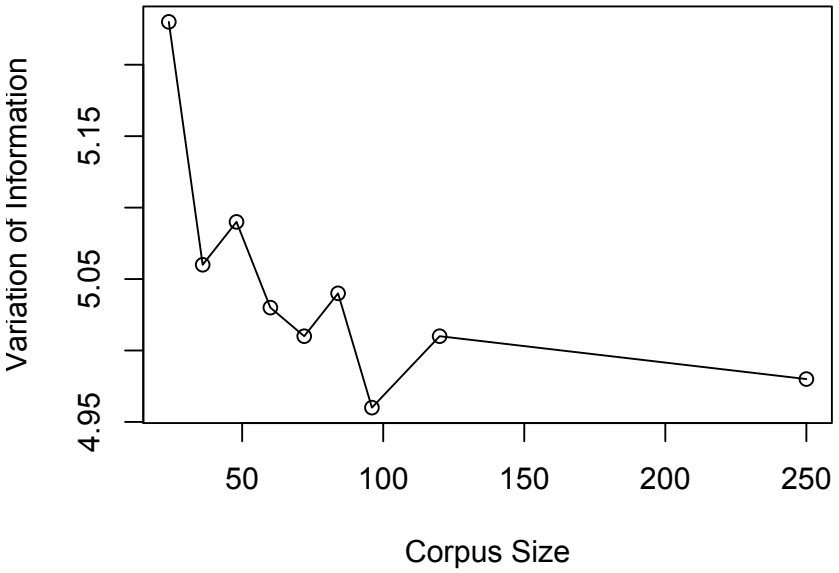


Figure 6.6: Variation of Information (VI) obtained from corpora of size 24K, 36K, 48K, 60K, 72K, 84K, 96K, 120K and 250K.

erations by ensuring that the sampler has enough time to converge. Figure 6.7 depicts one-to-one accuracy scores for various corpora of size 24K, 36K, 48K, 60K in different iterations of sampling.

If tags are considered individually, determiners, modal verbs, prepositions, pronouns, conjunctions, and numbers are captured generally correctly. Proper nouns are also distinguished from other nouns. However, they are spread over different tags. The most common errors are due the confusion of nouns and adjectives. Normally, nouns are over-spread in several tags. Verbs and adverbs are also generally confused; and spread over different tags.

We report our results with comparison to other systems in Table 6.1. We use a small portion of Penn WSJ treebank for the comparison. The dataset involves 250K words where the number of word types is 20957. The other systems are also tested on a small portion of WSJ involving 16850 word types, which is reported in Christodoulopoulos et al. (2011). One of the unsupervised state-of-

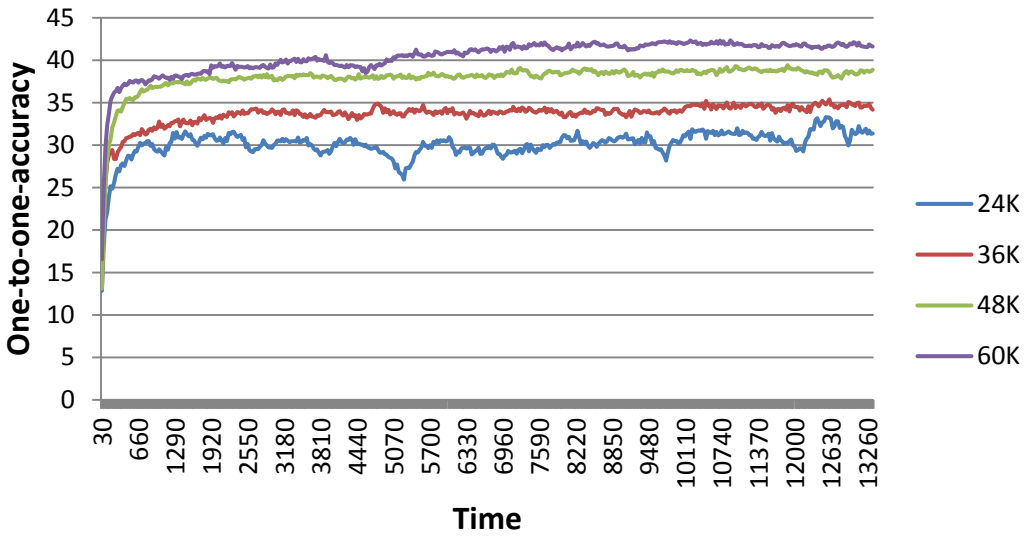


Figure 6.7: One-to-one scores that vary in different iterations of Gibbs sampler.

art system in POS tagging is Clark (2003), which is the best published score reported in (Christodoulopoulos et al. 2010). The system follows a type-based setting. Christodoulopoulos et al. (2010) reports higher scores for a small set in WSJ with their Bayesian multinomial mixture model (BMMM). However, it is worthwhile to say that Clark (2003) still outperforms Christodoulopoulos et al. (2010) on the full dataset of WSJ. However, both systems are type based. Since it will make more sense scientifically to compare our system with another token based system, we also report the scores of Christodoulopoulos et al. (2011). Christodoulopoulos et al. (2011) formulates POS tagging as a Bayesian mixture model similarly to our system and also employs morphological features. The main setting of their system is type-based. However, they present results in both token and type-based settings. We compare our results with their token-based setting. Our system outperforms Christodoulopoulos et al. (2011) with the many-to-one evaluation. However, Christodoulopoulos et al. (2011) performs better than our system based on V-measure evaluation. These scores are scientifically significant due to the common datasets and the common evaluation methods used for all experiments.

	V-measure	Many-to-one
Christodoulopoulos et al. (2011)	48.6	57.8
Joint	41.11	59.67
Clark (2003)	63.8	68.8
Christodoulopoulos et al. (2010)	67.7	72.0

Table 6.1: Comparison with other systems.

6.6.2 Morphological Segmentation Results

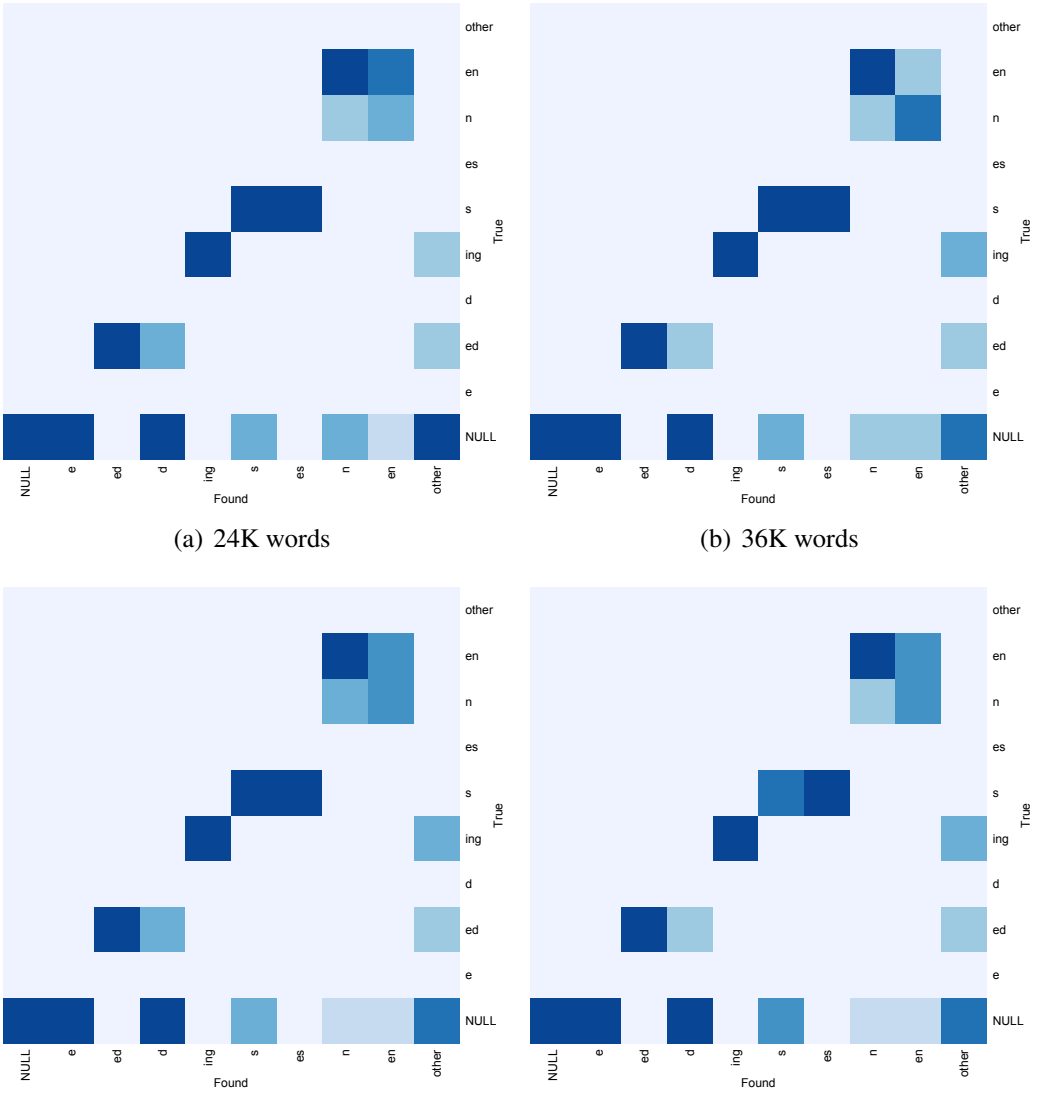
For the evaluation of morphological segmentation, a similar approach with Goldwater (2007) is applied. Evaluation is performed over verbs. To prepare a gold standard for verbs, common endings of verbs are stripped off. To this end endings such that *-ed, -d, -ing, -s, -es* are stripped off from verbs ending with one of those morphemes. In addition, irregular verb forms are also considered. If a verb is ending with *-n* or *-en*, they are again stripped off from the verb provided that the stem is also a word; such that *begun* is left unsplit whereas *broken* is split with the morpheme *-n*. Other irregular verb forms (i.e. *torn, sworn, spun*, etc.) are exceptionally introduced to be left as they are (with a NULL suffix).

Confusion matrices depicting found morphemes against true morphemes are given in Figure 6.8 and Figure 6.9 for various sizes of corpora. The darker shades denote a higher matching between found and true morphemes, whereas lighter shades correspond to a lower matching between found and true morphemes.

The common ending *-e* is usually recognised as a valid morpheme as also other most morphological segmentation systems do. The other common mistake comes with the identification of morphemes *-ed* with other adjacent letters for some words; such that *-ted*. The same mistake is made with the morpheme *ing* by recognising in some cases as *-ting*. Another type of common mistake comes from words identified with NULL suffix while in some cases it is true.

In addition to the common verb endings, additional morphemes such as *-ize, ized, -ify, -ied, -ped*, etc can also be identified by the model (see Table 6.2). These are not considered in the evaluation since they are rare in the corpus.

Obtained morphology results are compared with Morfessor Baseline. The experiment results obtained from 96K setting is compared with Morfessor Baseline.



(a) 24K words

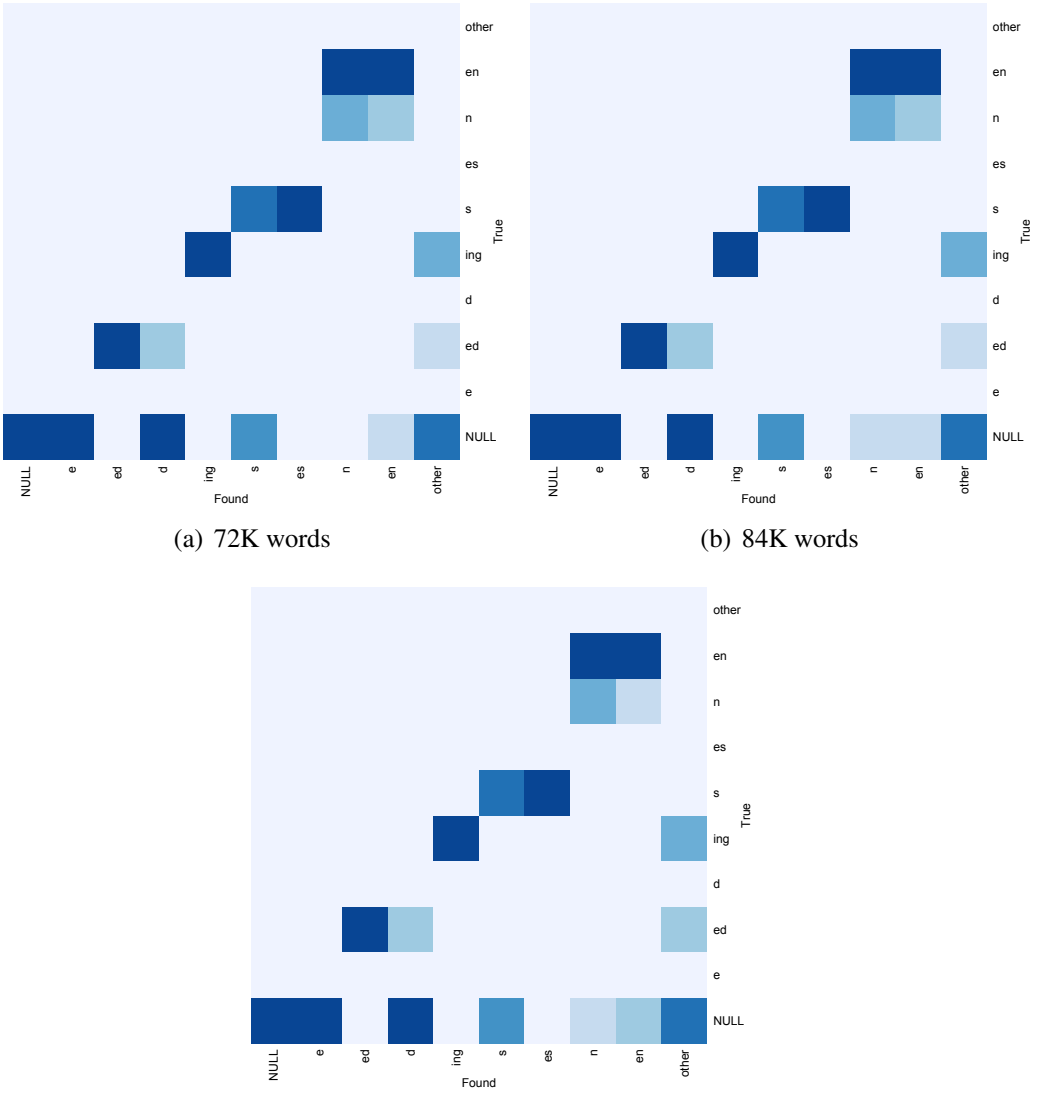
(b) 36K words

(c) 48K words

(d) 60K words

Figure 6.8: Confusion matrices obtained from different corpora show how correlated found morphemes and true morphemes are.

We run Morfessor Baseline on only the verbs to have the same dataset. Table 6.3 shows a comparison between our model (MorSyntax) and Morfessor Baseline regarding various values. The comparison contains percentages of: *missing types* which means that gold standard suggests a suffix but no suffix is identified in the results, *extra suffixes* denotes that gold standard does not identify any suffixes but



(a) 72K words

(b) 84K words

(c) 96K words

Figure 6.9: Confusion matrices obtained from different corpora show how correlated found morphemes and true morphemes are.

results contain suffixes, *wrong suffixes* mean that both gold standard and results identify suffixes but they are not the same, and finally *correct types* mean that both gold standard and results contain suffixes and they match. Our model identifies 12257 suffix types, whereas Morfessor Baseline identifies 2309 due to undersegmentation. This leads many suffixes to be missed by Morfessor Baseline.

-ize	standard+ize, material+ize
-ized	critic+ized, public+ized, balkan+ized, special+ized
-ped	drop+ped, scrap+ped, swap+ped
-ify	test+ify
-ied	appl+ied, clarif+ied,
-ified	unjust+ified

Table 6.2: Additional morphemes captured by the system.

	missing types	extra types	wrong types	correct types
MorSyntax	0.72%	28.55%	10.13%	60.60%
Morfessor Baseline	15.07%	7.23%	10.22%	67.48%

Table 6.3: Comparison with Morfessor Baseline.

Although correct types have a lower percentage in our model, when the number of identified morphemes are considered, our results cover more morphological information since wrong types have a similar percentage in both models.

6.7 Discussion

We proposed a new model that learns POS tags and morphology jointly. When the performance of both are considered, they are far from the state-of-art systems in both fields. However, our model is significant in both fields since it demonstrates that it is possible to learn POS tagging and morphology simultaneously.

We applied POS tagging with a Bayesian mixture model which is different than most traditional approaches in POS tagging. Most traditional approaches view POS tagging as a sequence labelling problem. With the proposed model, we tried a new approach where we viewed POS tagging as a clustering problem in a probabilistic perspective. The idea is similar to Clark's distributional clustering approach. However, our model considers other features as well as the contextual information. We benefit from mixture modelling where each mixture component is weighted by the number of words it contains. Therefore, mixture components that have more members attract more members.

Another difference in our model is its token based aspect. Most POS tag-

ging research adopts a type based approach where each word type is assumed to have only one tag. However, with our approach we made it possible to assign different tags for each word token. Thus, we consider each occurrence of word tokens belonging to the same word type individually by exploiting their context. For constraining the number of possible tags for each word type, we defined a probability distribution over words for each mixture component. It works like an emission probability in HMMs. The distribution favours having the same word types in each mixture component whereas still leaving probability mass for including the word types in other mixture components. The token-based aspect is the main reason that affects the accuracy in POS tagging.

We applied morphology learning by adopting Dirichet process. The Dirichlet process enables introducing an additional distribution to define some features about morphemes to be favoured. We used a length prior by exploiting this property of a Dirichlet process. Since a Dirichlet process shows rich-get-richer behaviour, it is possible to capture morphemes which are repeated within different words. This behaviour also supports MDL principle by preferring same morphemes throughout the corpus. The model can capture most common morphemes in words. Therefore, it forms a baseline system for morphology learning by identifying common endings. The common errors are similar to other systems' errors; such that identifying *-ed* as *-ted*.

Our results show that learning POS tags and morphology can be combined and performed cooperatively. In our model, morphology exploits POS tags; however, we did not employ morphological information for POS tagging. The model can be improved by adding more cooperation between morphology and POS tagging. We believe that this will increase the accuracy of the model.

6.8 Conclusion

A Bayesian model is proposed in this chapter to perform joint learning of morphology and POS tagging. The model definition is given along with the inference algorithm to infer POS tags and morphology simultaneously.

Results provide an evidence that a joint learning of morphology and POS tagging is possible. Although, there is still a lot to improve the model, the proposed

approach can be considered as a baseline system for morphological segmentation and POS tagging within the same learning mechanism.

CHAPTER 7

Morpheme Labelling

"Science is the systematic classification of experience."
George Henry Lewes

7.1 Introduction

In this chapter, an algorithm is presented for morpheme labelling. The algorithm employs hierarchical agglomerative clustering to group morphemes (mainly inflectional ones) according to their functions. The algorithm aims to capture allomorphs (for the definition see Chapter 2) and homophones [1].

7.2 Previous Work

There is little work on morpheme labelling in the literature. Spiegler (2011) presents two algorithms for morpheme labelling: one of them learns morpheme labels once morphological segmentation is completed and the other one learns

[1] Homophonous morphemes which are the same in writing, however have different roles such as the plural morpheme *-s* and present tense morpheme *-s*.

labels concurrently with morphological segmentation. Both algorithms are supervised where ground truth morphemes are provided.

Virpioja et al. (2009) focus on discovering allomorphs by extending the Morfessor Baseline (Creutz & Lagus 2007). Allomorphic variants are learned via mutations, which are modifications (i.e. substitution and deletion) that are applied on morphemes to produce various word forms. The concept of allomorphs in their work is slightly different than ours, where Virpioja et al. (2009) focus on learning base forms of words by using the fact that whether they are allomorphic variants of each other. For example, *glue* and *blue* are not allomorphic variants of each other, whereas *priest* and *priest's* are allomorphic variants.

Bernhard (2008) suggests another morpheme labelling algorithm which labels morphemes as a stem, suffix, base, or prefix. However, Bernhard (2008) does not address allomorphs or homophonous morphemes in her work.

7.3 Intuition

Most morphological segmentation algorithms consider only segmenting words into its morphemes and ignore labelling morphemes. However, morpheme labels are not only useful for other NLP problems (such as POS tagging), but also they give a better understanding on the morphological analysis of words. As mentioned in Chapter 2, there are different types of morphemes having different grammatical functions. The algorithm presented in this chapter aims to group morphemes according to their functionalities. This grouping is accomplished by considering two types of distinction among morphemes: allomorphs and homophonous morphemes.

7.3.1 Allomorphs

Morphemes may differ in the shape but still can carry out the same function in words such as the plural morpheme *-s* and *-ies* in English. Allomorphs are also seen quite often in some languages where vowel harmony (see Chapter 2 for the definition) takes place, such as in Turkish, Hungarian, Finnish etc. Some examples are given below in Turkish:

- The plural form (*-lar*, *-ler*): e.g. *elma-lar* (apples), *ev-ler* (houses).
- The possessive case (*-in*, *-un*, *-ün*, etc): e.g. *Ali'n-in* (Ali's), *Banu'n-un* (Banu's), *Üstün'-ün* (Üstün's).
- The present tense (*-ar*, *-ir*, etc): e.g. *yap-ar* (he does), *gel-ir* (he comes).
- The prepositional case (*-de*, *-da*): e.g. *ev-de* (at home), *okul-da* (in the school).

Vowel harmony is not the only phonological change which causes allomorphs in Turkish, also some consonants at the end of words are mutated depending on the following morpheme which is called a consonant mutation. The mutation occurs with the unvoiced consonants which are evolved into the voiced morphemes. Words ending with one of the unvoiced consonants (i.e. *p*, *ç*, *t*, *k*, *s*, *ş*, and *h*) force the added morpheme to mutate into an unvoiced consonant:

- The ablative case (*den*, *ten*): e.g. *ülke-den* (from the country), *sepe***t**-**t***en* (from the basket).
- The locative case (*de*, *te*): e.g. *şehir-de* (in the city), *ken***t**-**t***e* (in the town).
- The third person singular: e.g. *nefi***s**-**t***ir* (it is delicious), *zeki-dir* (she is clever).

Due to the vowel harmony and the consonant mutation, Turkish has many examples of morphemes which have the same functions but are phonological variants of each other. To this end, it is useful to group these morphemes into the same cluster assigning the same label.

7.3.2 Homophonous morphemes

On the contrary to allomorphs, some morphemes can have the same phonological properties but however they function differently. These morphemes are called homophonous morphemes. Homophonous morphemes should belong to different clusters due to the difference in their functions. Some examples in Turkish are given below:

- *kalem-i*: *-i* may correspond to an accusative (e.g. *his/her pen*) or a possessive case (e.g. *give me the pen*) depending on the context of the word.
- *yap-ın* (*do it*) and *kapın-ın* (*the door's*): *-ın* corresponds to an imperative in *yap-ın* , whereas it is a possessive in *kapın-ın*.
- *geliyor-lar* (*they are coming*) and *yatak-lar* (*the beds*): *-lar* refers to a 3rd person plural in *geliyor-lar*, whereas it is a plural in *geliyor-lar*.

Although homophonous morphemes do not occur very often like allomorphs, it is crucial to determine homophony to be able to distinguish morphemes having different functions. Homophonous morphemes should be grouped into separate clusters, whereas allomorphs should be grouped in the same cluster.

7.4 Background

In this section, hierarchical clustering algorithms are briefly described.

7.4.1 Hierarchical Clustering

Hierarchical clustering builds a hierarchy during the construction of clusters. The construction of the hierarchy can be accomplished in two different ways, agglomerative or divisive. In agglomerative clustering, each data point forms it own cluster initially. In each iteration, the most similar cluster pair is merged until having a single cluster. On the contrary, in divisive clustering, the algorithm starts with a single cluster and iteratively clusters are divided into two dissimilar clusters until having each data point as a separate cluster (see Figure 7.1).

Agglomerative clustering falls in three different types depending on how the distance between clusters are measured: single-linkage clustering, complete-linkage clustering and average-linkage clustering.

7.4.1.1 Single-Linkage Clustering

In single linkage agglomerative clustering (or nearest neighbour technique, shortest distance), the distance between two clusters is the distance between the closest members of two clusters (see Figure 7.2):

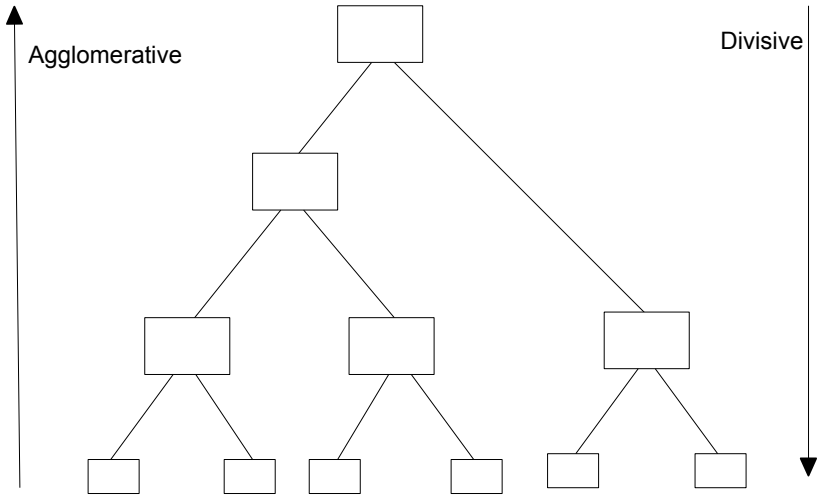


Figure 7.1: Agglomerative vs divisive clustering.

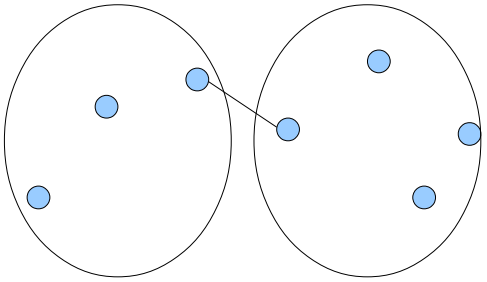


Figure 7.2: Distance measuring in single-linkage agglomerative clustering.

$$D(R,S) = Min_{r \in R, s \in S} d(r,s) \tag{7.1}$$

where $d(r,s)$ is the distance between the members r and s which are in cluster R and S respectively.

In each iteration, two clusters having the closest two members are merged. Since the distance between two members is considered in clustering, this approach may force two clusters, even though the rest of the data are dissimilar, if they have the closest members.

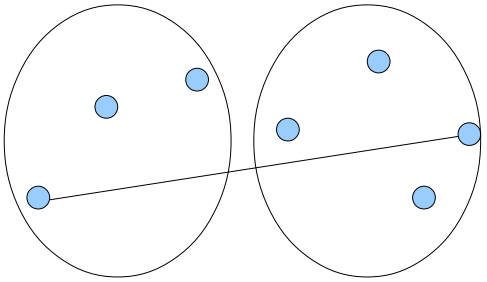


Figure 7.3: Distance measuring in complete-linkage agglomerative clustering.

7.4.1.2 Complete-Linkage Clustering

In complete-linkage agglomerative clustering (or farthest neighbour), in contrast to single-linkage clustering, the distance between two clusters is the distance between the most distant members of two clusters (see Figure 7.3):

$$D(R, S) = Max_{r \in R, s \in S} d(r, s) \tag{7.2}$$

where $d(r, s)$ is the distance between the members r and s which are in cluster R and S respectively.

Clusters having the minimum distance according to the given distance measurement are merged iteratively until having all the data in one cluster. Complete-linkage clustering has the same drawback as single-linkage clustering, because the distance depends on only two members in clusters, therefore the rest of the data is not considered.

7.4.1.3 Average-Linkage Clustering

In average linkage agglomerative clustering, the distance between two clusters is the average distance which is calculated through all pairs of data points between the clusters (see Figure 7.4):

$$D(R, S) = \frac{1}{N_R \times N_S} \sum_{i=1}^{N_R} \sum_{j=1}^{N_S} d(r_i, s_i) \tag{7.3}$$

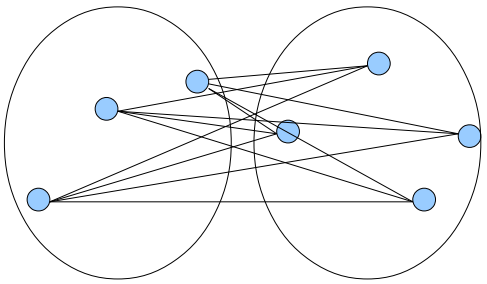


Figure 7.4: Distance measuring in average-linkage agglomerative clustering

where the total distance between two clusters R and S with sizes N_R and N_S is the summation of distances between each members of the clusters. The distance is normalised with the number of pairs.

The cluster pair having the minimum distance is merged in each iteration.

On the contrary to single-linkage and complete-linkage clustering, average-linkage clustering takes into account each data member which leads to a more realistic measurement.

7.5 The Algorithm for Clustering Morphemes

For morpheme labelling, we suggest a bottom-up agglomerative hierarchical clustering where morphemes showing functional similarities are clustered together. Functional similarities of morphemes are defined by a set of features as an input to the algorithm. Therefore, a feature vector is constructed to represent each morpheme by a vector. Each feature vector consists of a sequence of features which is given below:

- Current morpheme to be clustered (*CurMor*).
- Previous morpheme that precedes the current morpheme in the analysis of the same word (*PreMor*).
- Following morpheme that follows the current morpheme in the same word (*FolMor*).

	CurMor	**PreMor**	**FolMor**	**Stem**
Turkish	-ıl	-dır	-acak	ceza
English	ion	-	s	calculate

Table 7.1: Clustering features

	PreWMor	**FolWMor**	**pos**	**len**
Turkish	-lar	-	1	2
English	-	-ly	0	3

Table 7.2: Clustering features (cont')

- Stem of the word (*Stem*).
- The last morpheme of the preceding word (*PreWMor*).
- The last morpheme of the following word (*FolWMor*).
- Morpheme position in the word (*pos*) (i.e. if the morpheme comes just after the stem, then it is 0. If the morpheme is the last morpheme of the word, then it is 2, and if it surrounded by other morphemes, the value is 1.).
- Morpheme length in letters (*len*).

An example is given in Table 7.1 and Table 7.2. The first example is for the morpheme *-ıl* in the context *"O-n-lar ceza-lan-dır-ıl-acak-lar."* (they will be punished) in Turkish. The second example is for the morpheme *-ion* in the context *"She made the calculat-ion-s quick-ly.'*.

Constructing the feature vector of each morpheme initially, morphemes are placed in individual clusters to initiate the clustering algorithm. In each iteration, two clusters having the minimum distance between is chosen to be merged. The distance between two clusters is measured via Kullback-Leibler (KL) divergence through all features. Recall that KL divergence is not called as a distance metric since it is not symmetric:

$$KL(p \parallel q) = \sum_i p(i) log \frac{p(i)}{q(i)} \tag{7.4}$$

We use Jensen-Shannon divergence (also called information radius) which is the symmetric version of the KL divergence:

$$D(p \parallel q) = KL(p \parallel q) + KL(q \parallel p) \tag{7.5}$$

In average linkage agglomerative clustering, the distance between two clusters is the average distance which is calculated through all pairs of data points between the clusters. In our approach, instead of measuring the distance between all member pairs, we represent each cluster with a feature vector that keeps all the information that comes from each morpheme in that cluster. For example, previous morphemes of a cluster is the combination of all previous morphemes of the morphemes in that cluster. While qualitative features are combined, quantitative features, such as morpheme position and morpheme length, are averaged. Having a feature vector for each cluster, the similarity between two clusters, c_1 and c_2, is measured as follows:

$$\begin{aligned} Sim(c_1, c_2) &= D(CurMor_{c_1} \parallel CurMor_{c_2}) \\ &+ D(PreMor_{c_1} \parallel PreMor_{c_2}) \\ &+ D(FolMor_{c_1} \parallel FolMor_{c_2}) \\ &+ D(Stem_{c_1} \parallel Stem_{c_2}) \\ &+ D(PreWMor_{c_1} \parallel PreWMor_{c_2}) \\ &+ D(FolWMor_{c_1} \parallel FolWMor_{c_2}) \\ &+ |pos_{c_1} - pos_{c_2}| + |len_{c_1} - len_{c_2}| \end{aligned} \tag{7.6}$$

where $CurMor_{c_1}$ denotes the current morpheme set belonging to the cluster c_1, $PreMor_{c_1}$ is the set of previous morphemes, $FolMor_{c_1}$ denotes the following

morphemes, $Stem_{c_1}$ denotes the stems, $PreWMor_{c_1}$ is the set of last morphemes of previous words and $FolWMor_{c_1}$ is the set of last morphemes of following words in cluster c_1. In addition to the qualitative features, quantitative features pos_{c_1} and len_{c_1} refer to the average position and the average length of the morphemes respectively belonging to the cluster c_1.

The algorithm starts with N morphemes, each belonging to an individual cluster. In each iteration, two clusters having the minimum KL divergence are merged until having all the morphemes in one cluster, which is to be the root node in the hierarchical tree. Before computing KL divergence between two feature vectors, we apply add-n smoothing to eliminate counts having a zero value in the vectors.

7.6 Experiments & Results

We used the gold standard word lists in Turkish and English provided by Kurimo et al. (2011b) for the experiments. The word lists contain 552 words in English and 783 words in Turkish. The datasets are small, however, it is sufficient to show that the algorithm can capture a good number of allomorphs.

The gold standard lists consist of morphological analyses of words, such as:

abacuses	*abacus*	N	PL
abstained	*abstain*	V	PAST

where both the segmentations of words and the labels of the morphemes are provided. For example, the analysis of the *abacuses* is provided with the label of the suffix, which is plural. Moreover, part-of-speech tags are also provided in the gold standard lists (e.g. verb (V), noun (N), etc.). However, we do not use part-of-speech tags.

To constitute the training sets for our clustering algorithm, we replaced morpheme labels in the gold standard sets with the actual morphemes manually, as follows:

Morphemes	Words
-ism, -ion,	*heroism, deduction etc.*
-ed, -ing	*inserted, roofed, leaked, arising, pulsing, rating etc.*
-ness, -ity	*extensiveness, community, earthiness etc.*
-s	*townsman, yachts, yachtsman etc.*
-er	*baby-sitters, planners, matchmakers etc.*
-s'	*humanities', protestants', swimmers', reductions' etc.*

Table 7.3: Some clustered morphemes in English.

abacuses *abacus* es

abstained *abstain* ed

As an input to the clustering algorithm, we extracted all morphemes in the training sets. The final lists consist of 567 morphemes in English and 1749 morphemes in Turkish. Once having morphemes in the gold standard sets, we constructed feature vectors of morphemes. Subsequently, we applied the hierarchical clustering algorithm. Once the tree is constructed, we cut the tree at different levels to retrieve final clusters. Some resulting clusters in English is given in Table 7.3.

As English is not a morphologically rich language, no homophonous morphemes or allomorphs could be captured. The reason is that morphemes do not have sufficient contextual information. Nevertheless, morphemes that show similar functional properties (i.e. tenses, derivative morphemes) are captured by the clustering algorithm. For example, both *ism* and *-ion* are derivative morphemes that make the resulting word a noun, *-ed* and *-ing* are inflectional morphemes that define the tense of a verb and *-ness* and *-ity* are derivative morphemes. There are many redundant clusters that have only one type of morpheme, such as plural morpheme *-s*, possessive morpheme *-s'* etc.

Experiments in Turkish provide a better understanding of what type of clusters are obtained from the clustering algorithm. Some resulting clusters in Turkish are given in Table 7.4. It is easier to see from the Turkish results that a good number of allomorphs are captured due to the widely used vowel harmony in Turkish.

Morphemes	**Words**
-a, -e, -i, ı, -in	*faturaların-ı, kongreler-i, bilinmelerin-e, bağışıklığı-n, mağazaların-a etc.*
-dır, -dir	*almakta-dır, ödeyecekler-dir, değinilmeli-dir etc.*
-let, -t	*iş-let-ecek, kuru-t-urken, uza-t-abilir etc.*
-lığ, -liğ, -yış	*başarısız-lığ-ı, başla-yış-ını, isteksiz-liğ-inin etc.*
-nı, -ni, -ne, -na	*bırakabileceği-ni, yakalandığı-nı, düzeyleri-ne, mağazaları-na etc.*

Table 7.4: Some clustered morphemes in Turkish.

For example, allomorphs *-i* and *ı*; *-dır* and *-dir*, and *-nı* and *-ni* are captured. In addition to allomorphs, functionally similar morphemes *-a*, *-e*, *-i* and *ı*, *-in* that refer to the dative, accusative and genitive case respectively are also captured.

Here, we did not evaluate how much morpheme labelling helped in evaluation compared to without clustering. If the actual morphemes in the training sets instead of obtained cluster labels are used for the evaluation, we get a 100% accuracy since the evaluation is based on morphological segmentation and we already use the gold standard sets for the training. We leave the evaluation to measure how much morpheme labelling helped in evaluation compared to without clustering as a future work, which should be performed on the output of a real morphological segmentation system.

To evaluate our results, we replaced the morphemes in training datasets with the cluster labels which are obtained from the clustering algorithm, such that:

commutation	c_{50}	*mutate*	$+c_{34}$
contradiction	*contradict*	$+c_{34}$	
decoded	c_{50}	*code*	$+c_{43}$
knifed	*knife*	$+c_{43}$	

Here, the numerated ids of the clusters are used to define the cluster labels. Suffixes are inserted with a plus sign, whereas the rest of the morphemes are

	Non-affixes	Affixes	Total
Precision	84.53	62.14	68.02
Recall	77.62	28.40	42.86
F-measure	80.93	38.98	52.58

Table 7.5: Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem and morpheme position as features.

inserted solely with their labels. This provides a more comprehensive analysis on affixes and non-affixes separately.

We applied the evaluation method that Morpho Challenge (see Kurimo et al. (2011b)) follows. In the Morpho Challenge evaluation method, words are analysed through word pairs that share common morphemes. For example, two words *book+s* and *pen+s* share a common morpheme in gold standard. To analyse if they are correctly segmented, it is checked whether the two words share a common morpheme in the results. Therefore, it does not make difference to use morphemes or labels.

We tested our algorithm with different combinations of features. Results for Turkish exploiting the features, previous morpheme, following morpheme, current morpheme, stem and morpheme position are given in Table 7.5. The results consist of 162 clusters. The number of clusters is chosen accordingly with the highest evaluation score obtained.

Here, two types of analyses are presented: non-affixes and affixes. As mentioned shortly before, evaluation with non-affixes considers only non-affixes; whereas evaluation with affixes considers the rest of morphemes, stems and prefixes.

Results from another experiment, employing previous morpheme, following morpheme, current morpheme, stem, morpheme position and morpheme length are given in Table 7.6. The results are analysed based on the same number of clusters, to investigate the impact of using different features. Here we can observe that using morpheme length as a feature improves the results.

Another experiment explores the impact of using the last morphemes of the

	Non-affixes	**Affixes**	**Total**
Precision	87.15	57.45	65.04
Recall	79.51	31.76	45.79
F-measure	83.15	40.91	53.74

Table 7.6: Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem, morpheme position and morpheme length.

	Non-affixes	**Affixes**	**Total**
Precision	87.93	46.95	61.06
Recall	73.05	12.03	29.96
F-measure	79.80	19.15	40.20

Table 7.7: Evaluation results with 162 clusters in Turkish, using previous morpheme, following morpheme, current morpheme, stem, morpheme position, last morphemes of the previous and following word.

previous word and the following word. Results of the experiment, using previous morpheme, following morpheme, current morpheme, stem, last morpheme of the previous word and last morpheme of the following word are given in Table 7.7. Results show that using last morphemes of the previous and following word does not improve, but reduce the scores.

We carried out another experiment by weighting features. The weights are set as a result of several experiments, as follows:

$$\begin{aligned} D(c_1, c_2) &= 0.3D(CurMor_{c_1} \parallel CurMor_{c_2}) \\ &+ 0.2D(PreMor_{c_1} \parallel PreMor_{c_2}) \\ &+ 0.2D(FolMor_{c_1} \parallel FolMor_{c_2}) \\ &+ 0.2D(Stem_{c_1} \parallel Stem_{c_2}) \\ &+ 0.1|pos_{c_1} - pos_{c_2}| \end{aligned} \tag{7.7}$$

	Non-affixes	Affixes	Total
Precision	93.82	69.64	80.23
Recall	86.34	44.08	74.41
F-measure	89.92	53.98	77.21

Table 7.8: Evaluation results, by weighting features, previous morpheme, following morpheme, current morpheme, stem and morpheme position in Turkish

	Non-affixes	Affixes	Total
Precision	95.60	90.72	92.93
Recall	84.79	34.46	70.59
F-measure	89.87	49.95	80.24

Table 7.9: Evaluation results by weighting features previous morpheme, following morpheme, current morpheme, stem, morpheme position an morpheme length. The experiment is performed on English and observed through 100 clusters.

The results of the weighted clustering algorithm, using the previous morpheme, following morpheme, current morpheme, stem and morpheme position are given in Table 7.8 in Turkish.

We also evaluated the algorithm for English. We employed previous morpheme, following morpheme, current morpheme, stem, morpheme position and morpheme length as features. We obtained the results from 100 clusters. Results are given in Table 7.9. In the experiment, the features are also weighted the same as the previous experiment.

7.7 Discussion

We tested the proposed clustering algorithm with various combinations of features. It should be noted that using previous and following morpheme in English is not very beneficial due to the simple morphology of the language. However, we still used these two features because of a number of words having more than one morpheme. Since Turkish is richer in morphology than English, previous

and following morpheme are more helpful in clustering of Turkish morphemes.

Another issue in Turkish morphology that needs to be considered is the ambiguity of morphemes. Words can be segmented in different ways depending on the context in the sentence, which can be discovered by looking at the meaning of the word. Hence it is meaningful to use the context of a morpheme in clustering.

In all experiments we assign weights to features manually. Weighting features improves results since features are not equally important in clustering. We leave the issue of estimating weights to be explored in the future.

7.8 Conclusion

In this chapter, an agglomerative hierarchical clustering algorithm is presented for morpheme labelling. The algorithm aims to capture allomorphs and homophonous morphemes for a deeper analysis of segmentation results of a morphological segmentation system. Most morphological segmentation systems focus on only segmentation rather than labelling morphemes according to their functions in words, i.e. inflectional (cases, tenses etc.) vs. derivational. Nevertheless, it is helpful to have a better understanding of the functions of morphemes in a word to be able to judge the grammatical function of that word in a sentence; i.e. the syntactic category. We believe that a good morpheme labelling system will help POS tagging, as well.

The presented algorithm can discover allomorphs in Turkish by clustering them together. However, as far as we could observe from the results, it cannot show the same accuracy for homophonous morphemes.

CHAPTER 8

Conclusion and Future Work

"Every end is a new beginning."
Proverb

8.1 Introduction

In this chapter, we provide a summary of the main conclusions of the thesis. The chapter provides also a list of research directions to lead subsequent research in the fields of unsupervised morphology learning and POS tagging.

8.2 Thesis Summary

This thesis presents various research directions in the fields of unsupervised morphology learning and POS tagging. In Chapter 2, the essential background knowledge is presented to prepare the reader for the rest of the thesis. In the chapter, linguistic terms are defined, along with prominent statistical parameter estimation methods, which are either used for the research in the thesis, or for other research in morphology and POS tagging. Chapter 3 presents a literature review

of unsupervised morphology learning and POS tagging. The most prominent research is reviewed in both fields. In Chapter 4, we present a novel algorithm that learns morphology by using syntactic categories. The algorithm learns the morphology along with a set of morphological paradigms that are captured through syntactic categories. In Chapter 5, we present another novel approach for morphological segmentation, which suggests adopting hierarchical tree structures to capture morphological paradigms. In Chapter 6, we illustrate a joint learning model where morphology and POS are learned simultaneously within the same learning mechanism. Finally, in Chapter 7, we present an algorithm for clustering morphemes according to the functions they fulfil in a sentence. The clustering algorithm is used for labelling morphemes where each label refers to a different function that a morpheme can adopt.

8.3 Contributions

This thesis makes the following contributions to the research in the context of two fields, unsupervised morphology learning and POS tagging:

- **Learning morphology through syntactic categories:** We define a novel algorithm that incorporates syntax to conduct morphological segmentation. The approach we propose is a paradigmatic approach that performs morphological segmentation through morphological paradigms. Morphological paradigms are very influential in morphological segmentation, and have been adopted extensively in the field of morphological learning. However, the incorporation of syntax into paradigm learning has not been applied in the field before.

- **Probabilistic hierarchical clustering of morphological paradigms:** We define a probabilistic hierarchical clustering algorithm for morphological segmentation. The proposed clustering algorithm captures paradigms in a hierarchical structure, i.e. trees. This method not only provides a hierarchical organisation of paradigms, but also efficiently captures morphological similarities between words. To our knowledge, current paradigmatic approaches in the field are flat and do not provide any hierarchical structure.

The proposed hierarchical clustering algorithm also contributes to the field of machine learning, by proposing an inference algorithm to learn latent variables in data, while learning the tree structure that will represent the data, within an optimum hierarchical structure.

- **Joint learning of morphology and POS tagging:** We define a novel approach that adopts a joint learning mechanism for morphology and POS tags. The proposed model contributes to the field in three ways: firstly, it is the first method to learn morphology and POS simultaneously; secondly, it is different from the traditional POS tagging approaches, which mostly view POS tagging as a sequence labelling problem, where our model adopts a mixture model, rather than a sequence modelling (i.e. HMMs); thirdly, it adopts a token-based setting in which words are tagged individually depending on their different contexts. Our results cannot outperform the type-based approaches, which is sensible. Nevertheless, our results outperform Christodoulopoulos et al. (2011) in their token-based setting. Morphology results are also promising, by suggesting a lot more segmentation than Morfessor Baseline (Creutz & Lagus 2002), where Morfessor Baseline suffers from undersegmentation. Although, there is still a lot to improve in the current method, our results are promising and prove that a joint learning of morphology and POS is possible.

- **Morheme labelling:** We propose a simple clustering algorithm to label morphemes according to their functionalities in sentences. The results show that morphemes can be distinguished according to their functions by using several features of morphemes (such as context of morphemes) within a hierarchical clustering scheme.

8.4 Future Work

We make several contributions to the research with this thesis. However, there is still a lot to improve the proposed approaches in this thesis:

- **Extension of capturing paradigms through syntactic categories:** The proposed model in Chapter 4 can be improved by eliminating the use of a dictionary. In the current approach, if a dictionary is not used, the number of paradigms is not sufficient to propose morphological segmentation for every word, especially for large vocabularies such as the one provided by Morpho Challenge 2010. To tackle this problem, the model can be settled in a Bayesian framework to capture more word forms, which leads to more paradigms and naturally to a more robust morphological segmentation system.

- **Extension of probabilistic hierarchical clustering:** The probabilistic hierarchical clustering algorithm suggested in Chapter 5 can be enhanced further to learn various features from the tree structure, i.e. POS tags. Learned tree structures provide a natural organisation of words, in a way that words with similar endings are grouped together. Since words with similar endings show similar syntactic features in general, it is possible to extract the syntactic categorial information (i.e. POS tags) from the tree structure.

  In the same probabilistic hierarchical clustering algorithm, instead of extracting syntactic features, we can use POS tags to improve morphological segmentation. In the current model, words with similar endings, regardless of their syntactic category, are grouped together. We can utilise the POS information to group words which are both morphologically and syntactically similar. This will improve the tree structure, and therefore morphological segmentation. To do this, we can also embed context information of words in each tree node, in addition to stem and suffix lists. Thus, while sampling a new position for a word in the tree, both its morphology and syntax are considered to make a coherent decision.

- **Extension of joint learning of morphology and POS tagging:** In Chapter 6, we use simple prior information for morphemes that favours shorter morphemes. The prior information can be replaced with more linguistically mo-

tivated prior information. This would improve the morphological segmentation in both models.

Another open issue with the model is the hyperparameter settings. The model can be improved by adding a hyperparameter training step in the current inference algorithm. Therefore, it would be possible to tune the hyperparameters to adapt to different languages, and different corpora.

One more possible improvement for the joint learning model is to use infinite mixture models for a flexible number of POS tags. Infinite HMMs have been used for POS tagging (Van Gael et al. 2009), however, infinite mixture models have not been used. Most systems define the number of POS tags apriori. The model can also be applied with a type-based setting, instead of a token-based setting. We believe that type-based setting will improve the current scores both for POS tagging and morphological segmentation.

- **Morpheme labelling:** The clustering algorithm proposed in Chapter 7 can be improved by deriving the approach in a nonparametric probabilistic environment by adopting mixture components for each morpheme label. Therefore, the model will be able to handle sparsity in the data. It is also possible to adopt infinite mixture models to introduce flexibility for the number of morpheme labels. In addition, mixture model formulation will enable using multiple features of morphemes.

8.5 Final Words

In this thesis, we focus on unsupervised morphology learning and POS tagging. Along with several contributions made to the research with this thesis, there is still a lot to contribute to the field by improving the proposed approaches.

As George Bernard Shaw said:

"Science never solves a problem without creating ten more."

APPENDIX A

Penn Treebank tags

Name	Category	Name	Category
$	dollar	“	opening quotation mark
”	closing quotation mark	(	opening paranthesis
)	closing paranthesisk	,	comma
–	dash	.	sentence terminator
:	colon or ellipsis	CC	conjunction, coordinating
CD	numeral, cardinal	DT	determiner
EX	existential there	FW	foreign word
IN	preposition or conjunction, subordinating	JJ	adjective or numeral, ordinal
JJR	adjective, comparative	JJS	adjective, superlative
LS	list item marker	MD	modal auxiliary
NNPS	noun, proper, plural	NNS	noun, common, plural
NN	noun, common, singular or mass	NNP	noun, proper, singular

PDT	pre-determiner	POS	genitive marker
PRP	pronoun, personal	PRP$	pronoun, possessive
RB	adverb	RBR	adverb, comparative
RBS	adverb, superlative	RP	particle
SYM	symbol	TO	“to” as preposition or infinitive marker
UH	interjection	VB	verb, base form
VBD	verb, past tense	VBG	verb, present participle or gerund
VBN	verb, past participle	VBP	verb, present tense, not 3rd person singular
VBZ	verb, present tense, 3rd person singular	WDT	WH-determiner
WP	WH-pronoun	WH-pronoun	possessive
WRB	WH-adverb		

Table A.1: 45 tags in full Penn Treebank tag set.

References

Abend, O., Reichart, R., & Rappoport, A. (2010). Improved unsupervised pos induction through prototype discovery. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, (pp. 1298–1307)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, *2*(6), pp. 1152–1174.

Argamon, S., Akiva, N., Amir, A., & Kapah, O. (2004). Efficient unsupervised recursive word segmentation using minimum description length. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, (pp. 1058–1064)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Arısoy, E., Dutağacı, H., & Arslan, L. M. (2006). A unified language model for large vocabulary continuous speech recognition of Turkish. *Signal Process.*, *86*, pp. 2844–2862.

Aunimo, L., Heinonen, O., Kuuskoski, R., Makkonen, J., Petit, R., & Virtanen, O. (2003). Question answering system for incomplete and noisy data. In F. Sebastiani (Ed.), *Advances in Information Retrieval*, volume 2633 of *Lecture Notes in Computer Science* (pp. 545–545). University of Helsinki Department of Computer Science, Finland: Springer Berlin / Heidelberg.

Avramidis, E. & Koehn, P. (2008). Enriching morphologically poor languages for statistical machine translation. In *ACL*, (pp. 763–770).

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

Bahl, L. R., Jelinek, F., & Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-5*, pp. 179–190.

Banko, M. & Moore, R. C. (2004). Part of speech tagging in context. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, (pp. 556–561)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Bauer, L. (2003). *Introducing Linguistic Morphology* (3rd ed.). 22 George Square, Edinburgh: Edinburgh University Press.

Bauer, L. (2004). *A Glossary of Morphology*. Edinburgh University Press Ltd.

Bernhard, D. (2008). Simple morpheme labelling in unsupervised morpheme analysis. In C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. W. Oard, A. Peñas, V. Petras, & D. Santos (Eds.), *Advances in Multilingual and Multimodal Information Retrieval* (pp. 873–880). Berlin, Heidelberg: Springer-Verlag.

Bernhard, D. (2009). Morphonet: Exploring the use of community structure for unsupervised morpheme analysis. In *Working Notes for the CLEF 2009 Workshop*, (pp. 598–608).

Berton, A., Fetter, P., & Regel-Brietzmann, P. (1996). Compound words in large-vocabulary German speech recognition systems. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, (pp. 1165 –1168).

Bhattacharya, I., Getoor, L., & Bengio, Y. (2004). Unsupervised sense disambiguation using bilingual probabilistic models. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, (pp. 287–294)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Biemann, C. (2006a). Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, (pp. 73–80)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Biemann, C. (2006b). Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, COLING ACL '06, (pp. 7–12)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Biemann, C., Giuliano, C., & Gliozzo, A. (2007). Unsupervised part-of-speech tagging supporting supervised methods. In *Proceedings of RANALP '07*.

Bilotti, M. W., Katz, B., & Lin, J. (2004). What works better for question answering: Stemming or morphological query expansion? In *Proceedings of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004*.

Blackwell, D. & MacQueen, J. B. (1973). Ferguson distributions via polya urn schemes. *The Annals of Statistics*, *1*, 353–355.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, pp. 993–1022.

Bordag, S. (2005). Unsupervised knowledge-free morpheme boundary detection. In *Proceedings of the RANLP '05*.

Bordag, S. (2006). Two-step approach to unsupervised morpheme segmentation. In *Proceedings of 2nd Pascal Challenges Workshop*, (pp. 25–29).

Bordag, S. (2008). *Unsupervised and Knowledge-Free Morpheme Segmentation and Analysis*, (pp. 881–891). Berlin, Heidelberg: Springer-Verlag.

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

Brent, M. R., Murthy, S. K., & Lundberg, A. (1995). Discovering morphemic suffixes a case study in mdl induction. In *Fifth International Workshop on AI and Statistics,Ft*, (pp. 264–271).

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, ANLC '92, (pp. 152–155)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Brown, P. F., Della Pietra, V. J., Della Pietra, S. A., & Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, *19*(2), 263–311.

Brown, P. F., deSouza, P. V., Mercer, R. L., Della Pietra, V. J., & Lai, J. C.

(1992). Class-based n-gram models of natural language. *Comput. Linguist.*, *18*(4), 467–479.

Bubenik, V. (1999). *An Introduction to the Study of Morphology*. LINCOM Europa.

Cai, J. F., Lee, W. S., & Teh, Y. W. (2007). Improving word sense disambiguation using topic features. In *Proceedings of EMNLP-CONLL'07*, (pp. 1015–1023).

Can, B. & Manandhar, S. (2009). Clustering morphological paradigms using syntactic categories. In *Proceedings of the 10th Cross-Language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments*, CLEF '09, (pp. 641–648)., Berlin, Heidelberg. Springer-Verlag.

Chan, E. (2006). Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, SIGPHON '06, (pp. 69–78)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Chan, E. (2008). *Structures and distributions in morphology learning*. PhD thesis, University of Pennsylvania.

Chandrasekar, R. & Srinivas, B. (1997). Using syntactic information in document filtering: A comparative study of part-of-speech tagging and supertagging. In *RIAO*, (pp. 531–546).

Chen, Y., Eisele, A., & Kay, M. (2008). Improving statistical machine translation efficiency by triangulation. In *LREC '08*. European Language Resources Association.

Chomsky, N. (1965). *Aspects of the Theory of Syntax* (2nd ed.). The United States of America: Massachusetts Institute of Technology.

Chowdhury, A. & McCabe, C. (1998). Improving information retrieval systems using part of speech tagging. Technical Report TR 1998-48, University of Maryland.

Christian Monson, Kristy Hollingshead, B. R. (2009). Probabilistic ParaMor. In *Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments*, CLEF '09.

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2010). 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 '10, (pp. 575–584)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2011). A Bayesian mixture model for part-of-speech induction using multiple features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, ANLC '88, (pp. 136–143)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Clark, A. (2003). Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, (pp. 59–66)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Clark, A. S. (2000). Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL-2000 and LLL-2000*, (pp. 91–94).

Clark, A. S. (2001). *Unsupervised Language Acquisition*. PhD thesis, University of Sussex.

Clark, S., Curran, J. R., & Osborne, M. (2003). Bootstrapping pos taggers using unlabelled data. In *Proceedings of CoNLL '03*, (pp. 49–55).

Creutz, M. (2003). Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, (pp. 280–287)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Creutz, M. (2006). *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. PhD thesis, Computer and Information Science, University of Technology, Espoo, Finland.

Creutz, M., Hirsimäki, T., Kurimo, M., Puurula, A., Pylkkönen, J., Siivola, V., Varjokallio, M., Arisoy, E., Saraçlar, M., & Stolcke, A. (2007). Morph-

based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Trans. Speech Lang. Process.*, *5*, pp. 1–29.

Creutz, M. & Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, (pp. 21–30)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Creutz, M. & Lagus, K. (2004). Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, SIGMorPhon '04, (pp. 43–51)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Creutz, M. & Lagus, K. (2005a). Inducing the morphological lexicon of a natural language from unannotated text. In *In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, (pp. 106–113).

Creutz, M. & Lagus, K. (2005b). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. *Technical Report A81*.

Creutz, M. & Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech Language Processing*, *4*, 1–34.

Croft, W. B., Turtle, H. R., & Lewist, D. D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th Annual International Conference on Research and Development in Information Retrieval (SIGIR'91)*, (pp. 32–45)., Chicago, USA.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, (pp. 318–329)., New York, NY, USA. ACM.

de Gispert, A. & Mariño, J. (2008). On the impact of morphology in English to Spanish statistical mt. *Speech Communication*, *50*, pp. 1034–1046.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R.

(1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), pp. 391–407.

Déjean, H. (1998). Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, NeMLaP3/CoNLL '98, (pp. 295–298)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Demberg, V. (2007). A language-independent unsupervised model for morphological segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (pp. 680–685).

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic coclustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, (pp. 89–98)., New York, NY, USA. ACM.

Diab, M. & Resnik, P. (2002). An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (pp. 255–262)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Emonds, J. E. (1985). *A Unified Theory of Syntactic Categories* (1st ed.). Dordrecht, The Netherlands: Foris Publications Holland.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. In Rizvi, H. & Rustagi, J. (Eds.), *In Recent Advances in Statistics*, (pp. 287–303). New York: Academic Press.

Finch, S. & Chater, N. (1992). Bootstrapping syntactic categories using statistical methods. In *In Background and Experiments in Machine Learning of Natural Language*, (pp. 229–235).

Ford, A., Singh, R., & Martohardjono, G. (1967). *Pace Panini*. Peter Lang.

Frank, S., Goldwater, S., & Keller, F. (2009). Evaluating models of syntactic category acquisition without using a gold standard. In *Proceedings of CogSci '09*.

Freitag, D. (2004). Toward unsupervised whole-corpus tagging. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Freitag, D. (2005). Morphology induction from term clusters. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, (pp. 128–135)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Gao, J. & Johnson, M. (2008). A comparison of Bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, (pp. 344–352)., Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. In *Natural Language Engineering*, volume 12, (pp. 353–371).

Goldwater, S. & Griffiths, T. (2007). A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (pp. 744–751)., Prague, Czech Republic. Association for Computational Linguistics.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2006). Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, Cambridge, MA. MIT Press.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. In *In 46th Annual Meeting of the ACL*, (pp. 398–406).

Goldwater, S. & McClosky, D. (2005). Improving statistical mt through morphological analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, (pp. 676–683)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Goldwater, S. J. (2007). *Nonparametric Bayesian Models of Lexical Acquisition*. PhD thesis, Department of Cognitive and Linguistic Sciences, Brown University.

Golénia, B., Spiegler, S., & Flach, P. (2009). Ungrade: Unsupervised graph decomposition. In *Working Notes for the CLEF 2009 Workshop*.

Grünwald, P. (2005). A tutorial introduction to the minimum description length

principle. In *Advances in Minimum Description Length: Theory and Applications*. MIT Press.

Habash, N. & Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, (pp. 573–580)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Habash, N. & Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, (pp. 49–52)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Hafer, M. A. & Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval*, *10*(11-12), pp. 371 – 385.

Haghighi, A. & Klein, D. (2006). Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, (pp. 320–327)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Hänig, C., Bordag, S., & Quasthoff, U. (2008). Unsuparse: Unsupervised parsing with unsupervised part of speech tagging. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., & Tapias, D. (Eds.), *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Harman, D. (1991). How effective is suffixing. *Journal of the American Society for Information Science*, *42(1)*, pp. 7–15.

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

Hasan, K. S. & Ng, V. (2009). Weakly supervised part-of-speech tagging for morphologically-rich, resource-scarce languages. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, (pp. 363–371)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Haspelmath, M. (2002). *Understanding Morphology* (1st ed.). New York: Oxford University Press Inc.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, pp. 97–109.

Hu, Y., Matveeva, I., Goldsmith, J., & Sprague, C. (2005). Using morphology and syntax together in unsupervised learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, PMHLA '05, (pp. 20–27)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Ishwaran, H. & James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, *96*, pp. 161–173.

Järvelin, K. & Pirkola, A. (2005). Morphological processing in mono- and cross-lingual information retrieval. In Arppe, A., Lauri Carlson, Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., & Yli-Jyrä, A. (Eds.), *Inquiries into Words, Constraints and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, (pp. 214–226)., Stanford, California. CSLI Publications.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, *64*, pp. 532–556.

Johnson, M. (2001). Joint and conditional estimation of tagging and parsing models. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, (pp. 322–329)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Johnson, M. (2007). Why doesn't em find good hmm pos-taggers. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, (pp. 296–305).

Johnson, M., Geman, S., Canon, S., Chi, Z., & Riezler, S. (1999). Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, (pp. 535–541)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An intro-

duction to variational methods for graphical models. *Machine Learning*, *37*, pp. 183–233.

Katamba, F. & Stonham, J. (2006). *Modern Linguistics Morphology* (2nd ed.). New York: Palgrave Macmillan.

Kazakov, D. (1997). Unsupervised learning of naive morphology with genetic algorithms. In *ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks, Prague*, (pp. 105–112).

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

Keshava, S. & Pitler, E. (2006). A simpler, intuitive approach to morpheme induction. In *In PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, (pp. 31–35).

Kettunen, K., Kunttu, T., & Järvelin, K. (2005). To stem or lemmatize a highly inflectional language in a probabilistic ir environment? *Journal of Documentation*, *61*(4), pp. 476–496.

Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., & Stolcke, A. (2006). Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*, *20*(4), pp. 589–608.

Kirchhoff, K. & Yang, M. (2005). Improved language modeling for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, (pp. 125–128)., Ann Arbor, Michigan. Association for Computational Linguistics.

Koehn, P. & Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (pp. 868–876).

Kohonen, O., Virpioja, S., Leppänen, L., & Lagus, K. (2010). Semi-supervised extensions to Morfessor baseline. In Kurimo, M., Virpioja, S., Turunen, V., & Lagus, K. (Eds.), *Proceedings of the Morpho Challenge 2010 Workshop*, (pp. 30–34)., Aalto University, Espoo, Finland.

Koskenniemi, K. (1983). *Two-level morphology: a general computational model for word-form recognition and production*. Department of General linguistics, University of Helsinki.

Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, ACL '84, (pp. 178–181)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Kristina Toutanova, Hisami Suzuki, A. R. (2008). Applying morphology generation models to machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (pp. 514–522)., Columbus, Ohio, USA. Association for Computational Linguistics.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, (pp. 191–202)., New York, NY, USA. ACM.

Kuhn, J. (2004). Experiments in parallel-text based grammar induction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kupiec, J. (1992). Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, *6*, pp. 225–242.

Kurimo, M., Lagus, K., Virpioja, S., & Turunen, V. (2011a). Morpho challenge 2009. `http://research.ics.tkk.fi/events/morphochallenge2009/`.

Kurimo, M., Lagus, K., Virpioja, S., & Turunen, V. (2011b). Morpho challenge 2010. `http://research.ics.tkk.fi/events/morphochallenge2010/`.

Kurimo, M., Virpioja, S., & Turunen, V. (2010). Proceedings of the morpho challenge 2010 workshop. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, (pp. 87–95)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Kurimo, M., Virpioja, S., Turunen, V. T., Blackwood, G. W., & Byrne, W. (2009). Overview and results of morpho challenge 2009. In *Proceedings of the 10th*

cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments, CLEF'09, (pp. 578–597)., Berlin, Heidelberg. Springer-Verlag.

Kurimo, M., Virpioja, S., Turunen, V. T., Gólenia, B., Spiegler, S., Ray, O., Flach, P., Kohonen, O., Leppänen, L., Lagus, K., Lignos, C., Nicolas, L., Farré, J., & Molinero, M. (2010). Proceedings of the Morpho Challenge 2010 workshop. *Technical Report TKK-ICS-R37*.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, (pp. 282–289)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lamb, S. M. (1961). On the mechanisation of syntactic analysis. In *1961 Conference on Machine Translation of Languages and Applied Language Analysis*, volume 2, (pp. 674–685)., HMSO, London.

Landauer, T. K., Folts, P. W., & Laham, D. (1998). Introduction to latent semantic analysis. *25*, pp. 259–284.

Larson, M., Willett, D., Köhler, J., & Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In *International Conference on Spoken Language Processing*, (pp. 945–948).

Lavallée, J. F. & Langlais, P. (2009). Morphological acquisition by formal analogy. In *Working Notes for the CLEF 2009 Workshop*.

Lee, Y. K., Haghighi, A., & Barzilay, R. (2011). Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, (pp. 1–9)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Leonard E. Baum, J. A. E. (1967). An inequality with the applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematicians Society*, *73*, pp. 360–363.

Lignos, C. (2010). Learning from unseen data. In Kurimo, M., Virpioja, S., Turunen, V., & Lagus, K. (Eds.), *Proceedings of the Morpho Challenge 2010*

Workshop, (pp. 35–38)., Aalto University, Espoo, Finland.

Lignos, C., Chan, E., Marcus, M. P., & Yang, C. (2009). A rule-based unsupervised morphology learning framework. In *Working Notes for the CLEF 2009 Workshop*.

Lioma, C. & Blanco, R. (2009). Part of speech based term weighting for information retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, (pp. 412–423)., Berlin, Heidelberg. Springer-Verlag.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing* (2nd ed.). Cambridge, Massachusetts: The MIT Press.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, *19*(2), pp. 313–330.

Martin, S., Liermann, J., & Ney, H. (1998). Algorithms for bigram and trigram word clustering. *Speech Communication*, *24*, pp. 19–37.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, (pp. 173–187).

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, *20*, pp. 155–171.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, pp. 1087–1092.

Ming Li, P. V. (1997). *Introduction to Kolmogorov complexity and its applications* (2nd ed.). New York: Springer.

Minkov, E., Toutanova, K., & Suzuki, H. (2007). Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (pp. 128–135)., Prague, Czech Republic. Association for Computational Linguistics.

Monson, C., Carbonell, J., Lavie, A., & Levin, L. (2008). Paramor: Finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science* (pp. 900–907). Springer Berlin / Heidelberg.

Monz, C. (2011). Statistical machine translation with local lanaguage models.

In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (pp. 869–879)., Edinburgh, Scotland. Association for Computational Linguistics.

Morrison, D. R. (1968). Patricia - practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM*, *15*, pp. 514–534.

Naseem, T., Snyder, B., Eisenstein, J., & Barzilay, R. (2009). Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, *36*, pp. 341–385.

Neuvel, S. & Fulop, S. A. (2002). Unsupervised learning of morphology without morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, (pp. 31–40)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Ney, H., Essen, U., & Kneser, R. (1994). On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, *8*, pp. 1–38.

Nicolas, L., Farré, J., & Molinero, M. A. (2010). Unsupervised learning of concatenative morphology based on frequency-related form occurrence. In Kurimo, M., Virpioja, S., Turunen, V., & Lagus, K. (Eds.), *Proceedings of the Morpho Challenge 2010 Workshop*, (pp. 39–43)., Aalto University, Espoo, Finland.

Och, F. J. & Ney, H. (2001). Statistical multi-source translation. In *MT Summit 2001*, (pp. 253–258).

Och, F. J. & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (pp. 295–302)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Orbanz, P. & Teh, Y. W. (2010). Bayesian nonparametric models. In *Encyclopedia of Machine Learning* (pp. 81–89).

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, *102*(2), pp. 145–158.

Pitman, J. & Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, *25*(2), pp. 855–900.

Poon, H., Cherry, C., & Toutanova, K. (2009). 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 '09, (pp. 209–217)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Poon, H. & Domingos, P. (2008). Joint unsupervised coreference resolution with Markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, (pp. 650–659)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Rasmussen, C. E. (2000). The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, (pp. 554–560). MIT Press.

Reynar, J. C. & Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, (pp. 16–19)., Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry Theory*. River Edge, NJ, USA: World Scientific Publishing Co., Inc.

Ritchie, G. (1992). Languages generated by two-level morphological rules. *Computational Linguistics*, *18*, pp. 41–59.

Roeland Ordelman, A. V. H. & Jong, F. D. (2003). Compound decomposition in Dutch large vocabulary speech recognition. In *Proceedings of Eurospeech 2003*, (pp. 225–228).

Rosenberg, A. & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Empirical Methods in Natural Language Processing*.

Rosenfeld, R. (1997). A whole sentence maximum entropy language model. In *Proceedings of the IEEE Workshop on Speech Recognition and Understanding*.

Schleicher, A. (1859). *Zur Morphologie der Spreche*, volume 1. St. Pétersburg: Mémoires de l'Académie Impériale des Sciences de St. Pétersburg Series VII.

Schlünz, G. I., Bernard, E., & van Huyssteen, G. B. (2010). Part-of-speech ef-

fects on text-to-speech synthesis. In *Proceedings of the 2010 Conference of the Pattern Recognition Association of South Africa*, (pp. 257–262)., Stellenbosch, South Africa.

Schone, P. & Jurafsky, D. (2000). Knowledge-free induction of morphology using latent semantic analysis. In *In Proceedings of CoNLL-2000 and LLL-2000*, (pp. 67–72).

Schone, P. & Jurafsky, D. (2001). Knowledge-free induction of inflectional morphologies. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, (pp. 1–9)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Schütze, H. (1993). Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, (pp. 251–258)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of the Seventh Conference on European Chapter of the Association for Computational Linguistics*, EACL '95, (pp. 141–148)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*, pp. 639–650.

Smith, N. A. & Eisner, J. (2005). Contrastive estimation: training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, (pp. 354–362)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Snyder, B. & Barzilay, R. (2008). Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, (pp. 737–745)., Columbus, Ohio. Association for Computational Linguistics.

Spiegler, S. (2011). *Machine Learning For The Analysis Of Morphologically Complex Languages*. PhD thesis, Merchant Venturers School of Engineering, University of Bristol.

Spiegler, S., Golénia, B., & Flach, P. A. (2009). Unsupervised word decomposition with the promodes algorithm. In *Working Notes for the CLEF 2009*

Workshop.

Spiegler, S. & Monson, C. (2010). Emma: A novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*.

Stevenson, M. & Gaizauskas, R. (2000). Improving named entity recognition using annotated corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC2000) Workshop on Information Extraction Meets Corpus Linguistics*, (pp. 24–30)., Athens.

Sun, M. & Bellegarda, J. R. (2011). Improved pos tagging for text-to-speech synthesis. In *ICASSP*, (pp. 5384–5387). IEEE.

Tallerman, M. (1998). *Understanding Syntax*. New York: Oxford University Press Inc.

Tchoukalov, T., Monson, C., & Roark, B. (2009). Morphological analysis by multiple sequence alignment. In *Working Notes for the CLEF 2009 Workshop*.

Teh, Y. W. (2010). Dirichlet process. In *Encyclopedia of Machine Learning* (pp. 280–287).

Van Gael, J., Vlachos, A., & Ghahramani, Z. (2009). The infinite hmm for unsupervised pos tagging. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, (pp. 678–687)., Stroudsburg, PA, USA. Association for Computational Linguistics.

Virpioja, S., Kohonen, O., & Lagus, K. (2009). Unsupervised morpheme discovery with Allomorfessor. In *Working Notes for the CLEF 2009 Workshop*.

Virpioja, S., Turunen, V. T., Spiegler, S., Kohonen, O., & Kurimo, M. (2011). Empirical comparison of evaluation methods for unsupervised learning of morphology. In *Traitement Automatique des Langues*.

Wang, W., Huang, Z., & Harper, M. (2007). Semi-supervised learning of part-of-speech tagging of Mandarin transcribed speech. In *ICASSP*, (pp. 137–140).

Watson, R. (2006). Part-of-speech tagging models for parsing. In *Proceedings of CLUK 2006*.

Wilks, Y. (2000). Is word sense disambiguation just one more nlp task? *Computers and the Humanities*, *34*(1/2), pp. 235–243.

Wilks, Y. & Stevenson, M. (1998). The grammar of sense: Using part-of-speech

tags as a first step in semantic disambiguation. *Nat. Lang. Eng.*, *4*, pp. 135–143.

Yang, M. & Kirchoff, K. (2006). Phrase-based backoff models for machine translation of highly inflected languages. In *Proceedings of the 21st International Conference on Computational Linguistics*, (pp. 1017–1020).

Yoon, Y., Seon, C.-N., Lee, S., & Seo, J. (2006). Unsupervised word sense disambiguation for korean through the acyclic weighted digraph using corpus and dictionary. *Inf. Process. Manage.*, *42*, pp. 710–722.

Youssef, I., Sakr, M., & Kouta, M. (2009). Linguistic factors in statistical machine translation involving Arabic language. *IJCSNS International Journal of Computer Science and Network Security*, *9*(11).

Index

Harris (1955), 38, 66
Abend et al. (2010), 96
Virpioja et al. (2009), 142
Antoniak (1974), 59
Argamon et al. (2004), 72
Arısoy et al. (2006), 27
Aunimo et al. (2003), 28
Avramidis & Koehn (2008), 27
Baayen (2001), 77
Bahl et al. (1983), 87
Banko & Moore (2004), 87, 88
Bauer (2003), 37–39
Bauer (2004), 40
Leonard E. Baum (1967), 87
Bernhard (2009), 142
Berton et al. (1996), 27
Bhattacharya et al. (2004), 92
Biemann et al. (2007), 84
Biemann (2006b), 84
Bilotti et al. (2004), 28
Blei et al. (2003), 78
Bordag (2005), 67
Bordag (2006), 68
Bordag (2008), 68
Brent et al. (1995), 69
Brent (1999), 77
Brown et al. (1992), 82
Brown et al. (1993), 27
Bubenik (1999), 40
Cai et al. (2007), 29
Can & Manandhar (2009), 142
Chan (2006), 78, 118
Chan (2008), 74
Chandrasekar & Srinivas (1997), 29
Chen et al. (2008), 92
Chomsky (1965), 42
Chowdhury & McCabe (1998), 29
Church (1988), 88, 89
Clark (2000), 83, 155
Clark (2003), 101, 162, 163
Creutz & Lagus (2005b), 101, 151
Creutz & Lagus (2002), 72, 75, 77, 142
Creutz & Lagus (2005a), 76, 97

Creutz & Lagus (2004), 72, 75, 76
Creutz (2003), 75–77
Creutz et al. (2007), 27
Croft et al. (1991), 29
Cutting et al. (1992), 88, 89
Deerwester et al. (1990), 83, 98
Déjean (1998), 67
Demberg (2007), 73
Diab & Resnik (2002), 92
Emonds (1985), 43
Ferguson (1983), 59
Finch & Chater (1992), 83
Frank et al. (2009), 94
Freitag (2004), 84
Freitag (2005), 101
Gao & Johnson (2008), 94
de Gispert & Mariño (2008), 27
Goldsmith (2006), 69, 101
Goldsmith (2001), 69
Goldwater et al. (2009), 122
Christodoulopoulos et al. (2011), 161, 163
Goldwater et al. (2006), 76, 80
Goldwater & McClosky (2005), 27
Goldwater & Griffiths (2007), 90, 91, 94, 145, 160
Goldwater (2007), 47
Golénia et al. (2009), 142
Grünwald (2005), 50
Habash & Rambow (2005), 80
Habash & Sadat (2006), 30
Hafer & Weiss (1974), 67
Haghighi & Klein (2006), 93
Hänig et al. (2008), 30
Harman (1991), 28
Hasan & Ng (2009), 145
Haspelmath (2002), 40
Hastings (1970), 61
Ishwaran & James (2001), 57
Järvelin & Pirkola (2005), 28
Jelinek (1976), 87
Johnson et al. (1999), 78
Johnson (2001), 78
Johnson (2007), 89
Katamba & Stonham (2006), 37, 39, 43
Kazakov (1997), 73
Kazakov & Manandhar (2001), 73
Keshava & Pitler (2006), 73
Kettunen et al. (2005), 28
Kirchhoff & Yang (2005), 30
Kirchhoff et al. (2006), 27
Koehn & Hoang (2007), 27
Krovetz (1993), 28
Kuhn (2004), 92
Kupiec (1992), 88
Kurimo et al. (2010), 80, 81
Lamb (1961), 82
Landauer et al. (1998), 98
Larson et al. (2000), 27
Lavallée & Langlais (2009), 73, 142
Lee et al. (2011), 145
Ming Li (1997), 71
Lignos et al. (2009), 74, 114
Lignos (2010), 74, 141
Lioma & Blanco (2009), 29

Manning & Schütze (1999), 43
Marcus et al. (1993), 157
Martin et al. (1998), 96
Hu et al. (2005), 101
Meilă (2003), 93
Merialdo (1994), 86, 88
Metropolis et al. (1953), 61
Minkov et al. (2007), 27
Monz (2011), 30
Kurimo et al. (2009), 113
Kurimo et al. (2011b), 81, 132
Morrison (1968), 68
Naseem et al. (2009), 92
Neuvel & Fulop (2002), 73
Ney et al. (1994), 96
Nicolas et al. (2010), 141
Och & Ney (2001), 92
Och & Ney (2002), 78
Orbanz & Teh (2010), 54
Roeland Ordelman & Jong (2003), 27
Pitman (1995), 57
Pitman & Yor (1997), 57
Poon & Domingos (2008), 78
Poon et al. (2009), 78, 80
Christian Monson (2009), 76, 114, 142
Reynar & Ratnaparkhi (1997), 78
Rissanen (1989), 50
Rissanen (1978), 50
Rosenberg & Hirschberg (2007), 94
Rosenfeld (1997), 78
Schleicher (1859), 26
Schlünz et al. (2010), 30
Schone & Jurafsky (2000), 98
Schone & Jurafsky (2001), 98
Schütze (1993), 83
Schütze (1995), 83
Sethuraman (1994), 57
Smith & Eisner (2005), 78, 91
Snyder & Barzilay (2008), 77, 92
Spiegler & Monson (2010), 79, 81
Spiegler et al. (2009), 142
Stevenson & Gaizauskas (2000), 30
Sun & Bellegarda (2011), 30
Tallerman (1998), 42
Tchoukalov et al. (2009), 142
Teh (2010), 122
Kristina Toutanova (2008), 27
Van Gael et al. (2009), 91, 94
Watson (2006), 30
Wilks & Stevenson (1998), 29
Wilks (2000), 29
Yang & Kirchoff (2006), 27
Yoon et al. (2006), 29
Youssef et al. (2009), 30
Monson et al. (2008), 74
Christodoulopoulos et al. (2010), 162, 163