An Iterated Learning Framework for Unsupervised Part-of-Speech Induction

Christos Christodoulopoulos

Doctor of Philosophy
School of Informatics
University of Edinburgh
2013

Abstract

Computational approaches to linguistic analysis have been used for more than half a century. The main tools come from the field of Natural Language Processing (NLP) and are based on rule-based or corpora-based (*supervised*) methods. Despite the undeniable success of supervised learning methods in NLP, they have two main drawbacks: on the practical side, it is expensive to produce the manual annotation (or the rules) required and it is not easy to find annotators for less common languages. A theoretical disadvantage is that the computational analysis produced is tied to a specific theory or annotation scheme.

Unsupervised methods offer the possibility to expand our analyses into more resource-poor languages, and to move beyond the conventional linguistic theories. They are a way of observing patterns and regularities emerging directly from the data and can provide new linguistic insights.

In this thesis I explore unsupervised methods for inducing parts of speech across languages. I discuss the challenges in evaluation of unsupervised learning and at the same time, by looking at the historical evolution of part-of-speech systems, I make the case that the compartmentalised, traditional pipeline approach of NLP is not ideal for the task.

I present a generative Bayesian system that makes it easy to incorporate multiple diverse features, spanning different levels of linguistic structure, like morphology, lexical distribution, syntactic dependencies and word alignment information that allow for the examination of cross-linguistic patterns. I test the system using features provided by unsupervised systems in a pipeline mode (where the output of one system is the input to another) and show that the performance of the baseline (distributional) model increases significantly, reaching and in some cases surpassing the performance of state-of-the-art part-of-speech induction systems.

I then turn to the unsupervised systems that provided these sources of information (morphology, dependencies, word alignment) and examine the way that part-of-speech information influences their inference. Having established a bi-directional relationship between each system and my part-of-speech inducer, I describe an *iterated learning* method, where each component system is trained using the output of the other system in each iteration. The iterated learning method improves the performance of both component systems in each task.

Finally, using this iterated learning framework, and by using parts of speech as the central component, I produce chains of linguistic structure induction that combine all

the component systems to offer a more holistic view of NLP. To show the potential of this multi-level system, I demonstrate its use 'in the wild'. I describe the creation of a vastly multilingual parallel corpus based on 100 translations of the Bible in a diverse set of languages. Using the multi-level induction system, I induce cross-lingual clusters, and provide some qualitative results of my approach. I show that it is possible to discover similarities between languages that correspond to 'hidden' morphological, syntactic or semantic elements.

Lay Summary

Computational approaches to linguistic analysis have been used for more than half a century. The main tools come from the field of Natural Language Processing (NLP) and are based on supervised methods. Despite their undeniable success in NLP, supervised learning methods have two main drawbacks: on the practical side, it is expensive to produce the manual annotation (or the rules) required and it is not easy to find annotators for less common languages. A theoretical disadvantage is that the computational analysis produced is tied to a specific theory or annotation scheme.

Unsupervised methods, on the other hand, offer the possibility to expand our analyses into more resource-poor languages, and move beyond the conventional linguistic theories. They are a way of observing patterns and regularities emerging directly from the data and provide new linguistic insights.

In this thesis I explore unsupervised methods for inducing parts of speech across languages. I discuss the challenges in evaluation of unsupervised learning and at the same time, by looking at the historical evolution of part-of-speech, I make the case that the compartmentalised, traditional pipeline approach of NLP (where the output of one system is the input to the next) is not ideal for the task. I present a part-of-speech induction system that makes it easy to incorporate multiple diverse features, spanning different levels of linguistic structure, like morphology, lexical distribution, syntactic dependencies and word alignment information that allow for the examination of cross-linguistic patterns.

I then turn to the unsupervised systems that provide these sources of information (morphology, dependencies, word alignment) and examine the way that part-of-speech information influences their decisions and describe an iterated learning method, where each component system is trained using the output of the other system in each iteration.

Using this iterated learning framework, and by using parts of speech as the central component, I combine all the component systems in a chain that offers a more holistic view of NLP. I describe the creation of a vastly multilingual parallel corpus based on 100 translations of the Bible in a diverse set of languages, and provide some qualitative results of my approach.

Acknowledgements

A commonplace statement that I'm used to reading in acknowledgement sections is that there are lots of people without whom a dissertation would not have been possible. I always thought of it as an exaggeration or (at best) a pleasantry. That is, until I started my own PhD! Three, or three and a half years is not a lot of time for a whole PhD project and, while no-one is going to do your work for you, you need all the support you can get.

I was very fortunate to have the support of two of the best supervisors one can wish for, in Mark Steedman and Sharon Goldwater. Not only because of their tremendous academic knowledge, but also because they complemented each other in the most harmonious way for me. Mark, whose incredible depth and breadth of knowledge surprises me to this day, was an inspiration ever since he introduced me to computational linguistics during an MSc in Edinburgh. He allowed me to pursue wild ideas while steering me away from research pitfalls. Sharon embraced my project in its early stages and her energy and immense technical knowledge help me to shape the core of my thesis. Both Sharon and Mark's patience and willingness to be convinced about new potential directions led me to a better understanding of my project and (perhaps more importantly) of the underlying principles behind it. I am a better researcher thanks to them.

Apart from my supervisors, most of the weight for my support (academic, moral, or otherwise) fell on the past members of the 'Steedman gang': Michael Auli, Lexi Birch, Prachya (Arm) Boonkwan, Tom Kwiatkowski, Kira Mourão, Emily Thomforde, Luke Zettlemoyer, and its current members (also known as the 'Darkstar crew'): Bharat Ambati, Greg Coppola, Tejaswini Deoskar, Aciel Eshky, Mark Granroth-Wilding, Mike Lewis, Siva Reddy, Nathaniel Smith. Together with the rest of the students and postdocs of the ILCC, they made me feel part of a great community, and provided me with endless occasions of discussion, drinking, dining and general good times!

I would like to add my special thanks for two people in particular. First, to Mark Granroth-Wilding, for being an amazing friend since my MSc days, for giving me my British accent (and most of my Britishness in general), for the countless coffee sessions that fuelled my entire PhD with caffeine and ideas, for trusting me with his thesis (and accusing me of procrastinating in his own acknowledgements!) and for painstakingly going through this document and making sense of my ramblings. Second, to Yannis Konstas, for keeping me a bit closer to Greece, for being a constant source of academic stimulation and good music, for patiently listening to me through my endless rants

about part-of-speech induction, parsing and the state of NLP, for giving me his amazing parser and just for being a reminder of all that's good about Greece.

I'm also indebted to many researchers that helped me throughout my PhD, either by providing their code during my hunt for unsupervised part-of-speech induction systems, or by giving me their expertise on various subjects, or by reviewing my papers.

I want to thank all my friends, both here in Edinburgh and back in Greece for putting up with my academic nature and for constantly reminding me that life is more than a PhD. Special thanks to Iris Cremer for being the best flatmate in the world, and to guys from Krestena (and surrounding towns) for making me feel welcomed every time I go home.

I wouldn't be writing these lines if it wasn't for my parents Αφροδίτη and Στάθης. Not just for the obvious biological reasons, but also, for giving me a chance to come this far, for believing in me, for letting me follow my dreams even when they took me away from Greece and supporting me every step of the way (both financially and spiritually). You might not be able to understand these lines, but know that I thank you from the bottom of my heart. Σας ευχαριστώ. I also owe a great deal to my wonderful sister Ελένη for always knowing how to cheer me up and for showing me a good time every time I visited her.

Finally I want to express my deepest gratitude to Catriona Silvey, my amazing companion through my life's journey these past one and half years, for keeping my linguistics in check, for tirelessly helping me through the unavoidable darker times of my PhD, for copy-editing this thesis and for brightening up my life.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Christos Christodoulopoulos*)

Contents

CHAPTER 1

Introduction

Some of the most powerful tools in computational linguistics' arsenal come from recent advances in statistical natural language processing (NLP). These advances focus on key tasks as well as other complementary tasks or sub-tasks of the NLP *pipeline* shown in figure 1.1. However, as the figure implies, the traditional NLP pipeline approach tends to view the different components as 'black boxes'; that is, self-contained, independent tasks where the only connections come from the output of the previous task.

Until recently, the main focus in NLP has been *supervised* systems, where the computational analysis for any given task was performed either by direct encoding of linguistic knowledge (in rule-based systems) or by trying to extrapolate the knowledge from applying probabilistic models on manually-labelled data.

The present thesis follows the opposite approach of *unsupervised learning* where statistical analysis is performed on raw (unannotated) text in an attempt to discover hidden patterns in the data. The thesis will also challenge the idea of 'black-box' tasks.

The main focus is *syntactic category induction*, the unsupervised equivalent of part-of-speech tagging that lies between levels (2) and (3) of figure 1.1. Its supervised counterpart has been used as self contained task but its purpose was to relieve some of the computational effort of the syntactic analysis task (3) by reducing the syntactic ambiguity of the words to a small set of tags (not necessarily corresponding to linguistic notions of parts of speech).

However there are a number of arguments for the existence (unsupervised) part-

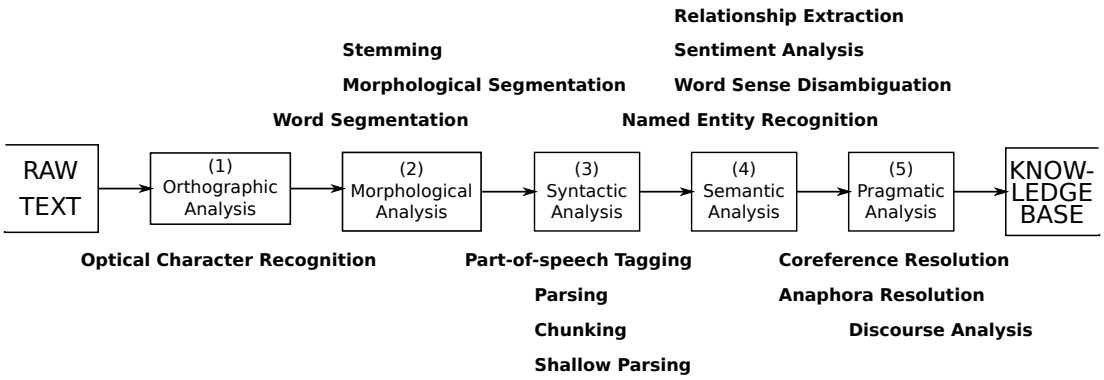


Figure 1.1: The traditional NLP pipeline (adapted from Garside et al., 1987, p. 11). The boxes represent the 5 major levels of computational analysis, along with various key NLP tasks placed at, or between their corresponding levels.

of-speech induction as a completely separate task. From an NLP perspective, having an unsupervised system to induce word classes is useful even if those classes are not labelled in any syntactically meaningful way. This is because the unsupervised classes will provided the same level of abstraction over the full lexicon (i.e. all the word types) that the supervised part-of-speech tags would and for that reason unsupervised parts of speech have been used in a variety of NLP research projects (some examples include Och & Ney, 2003; Täckström et al., 2012; Spitkovsky et al., 2011a and Koo et al., 2008).

Unsupervised part-of-speech induction as a task also makes sense from a theoretical linguistics point of view. Linguists have been trying to define parts of speech from the earliest of times and there are several competing theories as to their true nature. Something that is common to most linguistic definitions of parts of speech however, is that they rely on more than one level of linguistic structure and usually involve a mixture of morphological, syntactic, semantic and even pragmatic information. I will attempt to recreate such a holistic account of parts of speech in NLP where multiple sources of information are used as features in an unsupervised induction system. Combined with raw parallel texts (texts placed alongside their translations), this approach allows for a typological analysis of parts of speech, free of language- and formalism-specific biases that can be used to discover underlying similarities between the morphosyntactic units across languages.

1.1 The Thesis

The core statement of the current dissertation can be expressed as follows:

Unsupervised machine learning techniques that combine multiple levels of linguistic information can be used for cross-lingual[1] analysis by discovering statistical patterns or regularities contained in raw parallel text. These patterns might correspond to traditional linguistic analyses but, more interestingly, might provide us with new insights about language. The work described in the dissertation demonstrates the creation of such techniques, their theoretical properties and their application to the problem of cross-lingual part-of-speech induction.

This dissertation looks at the problem of part-of-speech induction from raw text, drawing inspiration from linguistic theory, where most of the definitions of parts of speech rely upon multiple sources of linguistic information, and tries to bring this insight to NLP research by using a part-of-speech induction system that can incorporate multiple sources of features.

1.1.1 Contributions of the thesis

This dissertation offers an in-depth analysis of unsupervised part-of-speech induction, alongside a comprehensive review of part-of-speech induction systems and evaluation metrics. There are also, two computational contributions: First, the creation of a new part-of-speech induction system called Bayesian Multinomial Mixture Model (BMMM) which allows the use of multiple sources of features and second, the *iterated learning* framework, a method that lets unsupervised NLP multiple systems to be combined with the BMMM by training each component system in the output of the other system in each iteration and whose performance allows for a more holistic view of NLP. Together, these contributions provide a better way of analysing cross-lingual data than the compartmentalised *pipeline* approaches, as demonstrated by empirical tests on standard NLP tasks.

The success of this approach is exemplified not only by performance improvements in traditional NLP tasks (see chapters 5 and 6), but also by providing a tool that can perform a multilevel linguistic analysis on multiple languages to induce clusters that reveal latent cross-language similarities. Since these tools are fully unsupervised, they

[1] I use the term cross-lingual to describe both parallel data and more generally, data in multiple languages (not parallel).

can be used for resource-poor languages where linguistic research is scarce, and also for an unbiased view of the data.

1.2 The structure of the thesis

The rest of the thesis is structured as follows. Since the main focus of the thesis is parts of speech (or syntactic categories), chapter 2 offers a review of the historical evolution of part-of-speech systems both in traditional linguistic research and as part of modern corpus-driven NLP. I will present some of the challenges in defining what parts of speech are; I will also discuss to what extent computational accounts of parts of speech—seen as a gateway for parsing—align with linguistic predictions.

In chapter 3 I will present an overview of unsupervised part-of-speech induction. The chapter will discuss issues concerning evaluation of unsupervised systems in general and examine empirically some of the most commonly used evaluation metrics before presenting a comparison of a number of unsupervised part-of-speech induction systems.

Chapter 4 will present a new probabilistic model that incorporates the most successful features of the systems examined in the previous chapter. The Bayesian Multinomial Mixture Model (BMMM) is based on the generative Bayesian framework and can be easily extended to use multiple local and non-local features such as contextual, morphological and multilingual word alignment information.

The BMMM is further extended in chapters 5 and 6 where I develop the idea of the *iterated learning* framework. Using this framework, dependency relations (chapter 5), morphology segmentations and word alignments (chapter 6) can not only be used as features, but also be induced alongside parts of speech, in an iterative manner, taking advantage of the interdependency between these structures and part-of-speech tags. In this way, parts of speech become a mediator between the many levels of natural language—morphology, lexicon and syntax—and, through word alignments, allow for a cross-lingual analysis across those levels. This is a small step towards a holistic view of computational linguistics, contrasted to the traditional modular pipeline view.

Finally, chapter 7 brings together the ideas from the previous three chapters in a proof-of-concept demonstration of chains of linguistic structure induction using a verse-aligned Bible corpus in 100 languages. I discuss the challenges in the creation of the corpus and present some qualitative analysis of the cross-lingual clusters. I show that it is possible to discover similarities between languages that correspond to ‘hidden’

morphological, syntactic or semantic elements. For example, by examining Greek and English aligned clusters, I present evidence that subjunctive mood might semantically present in English even though it rarely manifests overtly in the morphosyntactic level.

I conclude the thesis and present ideas for further research in chapter 8.

CHAPTER 2

Parts of Speech

[...] for we form no judgement till we have got language, and we must have the parts of speech before we can predicate anything.

Martineau (1866, p.277)

Part-of-Speech tagging is one of the first textual tasks in the NLP pipeline in figure 1.1, often considered to be self-contained. As a (supervised) machine learning task, compared to other tasks down the pipeline, it has a limited search space and large amounts of annotated data (in English at least). Finally, parts of speech have been shown to be a very useful source of information for downstream tasks (especially for parsing). All these factors make part-of-speech tagging an attractive task for the NLP community.

Similar to its supervised counterpart, an unsupervised part-of-speech induction system is designed to label each word—or each lexical unit—with a tag that effectively groups these units into categories (the parts of speech). Before we examine how unsupervised systems can perform this task (chapter 3), we need to look at the historical evolution of parts of speech as a means of linguistic analysis, the reason they evolved, as well as how they have been used in corpus-driven approaches to linguistic and computational linguistic analyses. As we will see, there is no consensus about the definition of parts of speech; however, most definitions rely on multiple sources of features (morphological, syntactic, semantic), something that as the main thesis in chapter 1 proposes, should be the goal of our NLP models.

2.1 Parts of Speech, Syntactic Categories or Word Classes?

In the literature of parts of speech there is often terminological disagreement between the use of the names 'parts of speech', 'syntactic categories', 'word classes', 'morphosyntactic tags', etc. This problem is perpetuated in the recent NLP literature, especially in the area of unsupervised learning. As we will see in section 3.2 this is not a mere definitional dispute. In defining *what* an unsupervised system learns we can more easily agree on what the evaluation criteria should be. Furthermore, by agreeing on the nature of parts of speech we can inform the nature of the unsupervised models themselves.

In fact there is no standard definition of syntactic categories, parts of speech or word classes. As Langacker (1987, p. 2) puts it, "every linguist relies on these concepts but few [...] are prepared to define them". Under some definitions (e.g. Haspelmath, 2001) these three terms are the same. However, some linguists maintain that syntactic categories are not the same as parts of speech—at least in their traditional sense. Gisa Rauh defines syntactic categories as "sets of linguistic items that can occupy the same portions in the [syntactic] structures of the sentences of a given language" (Rauh, 2010, p. 8). Under Rauh's definition phrasal structures such as noun-phrases (NPs) and prepositional-phrases (PPs) are also syntactic categories and therefore concludes that their number far exceeds the number of parts of speech. Perhaps a further distinction between *phrasal* and *lexical* syntactic categories would be more useful.

For the purpose of this thesis I will equate 'parts of speech' (a term the NLP community is more familiar with) with lexical syntactic categories. To the extent that parts of speech can characterise sub-word or super-word units they are also equivalent to syntactic categories in the general sense.

2.2 Historical overview of Parts of Speech

The historical evolution of part-of-speech systems is the history of linguistics as a science. All of us, to various degrees, have a culturally evolved understanding of our language; a kind of meta-linguistic self-consciousness. It is this linguistic awareness that has developed into linguistic enquiry and subsequently linguistic science in many cultures. It is, however, this same linguistic egocentricity that prevents us from focusing at the other end of the spectrum—looking at cross-lingual differences and similarities—and instead leads us to language-specific conclusions and even dismissive treatment of

other languages (Robins, 1969, p. 1). The history of parts of speech is split between these two extremes of linguistic introspection and cross-lingual analysis, a split that still remains to this day and will be pointed out throughout this chapter.

In his very enjoyable book *A Short History of Linguistics*, Robins (1969) starts his account with the ancient Greeks. Robins points out that this is not because the Greeks were the first to think about linguistics (or parts of speech). Indeed as early as the 6th or 5th century BCE, the Sanscrit grammarian Yāska defined 4 main categories of words: nouns (*nāma*), verbs (*ākhyāta*), prefixes (*upasarga*) and particles (*nipāta*). They belonged to either the inflected (nouns, verbs) or the uninflected (prefixes, particles) classes (Matilal, 1990, p. 18). This was the first recorded use of meta-language: linguistic labels to describe linguistic phenomena. However, the main discussion begins with the Greeks because there is an unbroken line of historic linguistic scholarship starting with the Greeks and continuing with the Romans, the Medieval scholars and the Renaissance thinkers into modern linguistics (Robins, 1969, p. 6).

Plato (360 BCE., 262a) is the first of the western philosophers to make a distinction between the nominal (*ónoma*) and the verbal (*rhēma*) component of the sentence (*lógos*). This was a purely semantic distinction: *rhēma* was 'the indication which relates to action' and *ónoma* 'those who perform the actions in question'. This distinction was further expanded by Aristotle to include conjunction (*sýndesmos*) that covered conjunctions, pronouns, articles and prepositions. This was the first definition that contained a morphological component. *Sýndesmoi* according to Aristotle are parts that are not inflected or declined. Aristotle was also the first one to define 'part of speech' (*méros lógou*): the word as component of the sentence having a meaning of its own but not further divisible into meaningful units (Robins, 1969, p. 26).

The Stoics further developed the Aristotelian part-of-speech system. They introduced new categories and defined them more precisely. Aristotle's *ónoma* was split into proper names (*ónoma*) and common nouns (*prosēgoría*). This was another semantic distinction: *ónoma* reflects a peculiar or individual quality (e.g. being Socrates), *prosēgoria* reflects a common quality or an attribute (e.g. being a human). Finally, they introduced pronoun (*árthro*), a nominal part that could stand for proper names but could not exist without them (Luhtala, 2000, p. 84–85).

This semantic distinction between proper names and common nouns was abandoned by the Alexandrians and specifically in the work of Dionysius Thrax *Tékhnē Grammatiké* (*The Art of Grammar*) where we find the first comprehensive account of parts of speech (Robins, 1969, p. 33–34). In the *Tékhnē* Dionysius defines eight parts

of speech (noun, verb, participle, article, pronoun, preposition, adverb, conjunction) based on a mixture of rigorous semantic, syntactic and morphological definitions; for instance a noun is defined as 'a part of the sentence which is subject to case inflection, and signified something corporeal or non corporeal' (Kemp, 1986). The list and the definitions have been the basis of most modern theories of parts of speech[1].

The next stage in the evolution of part-of-speech systems comes with the Latin philosophers Varo and later Priscian. According to Robins (1969, p. 50), Varo was the first to propose a purely morphological classification of words into those with case inflection (nouns), those with tense inflection (verbs), those with both case and tense inflection (particles) and those with neither (adverbs).

Priscian, around 500 A.D. wrote one the most comprehensive grammars of Latin. He was influenced greatly by the work of Dionysius and his part-of-speech system also contains eight categories with the only difference being the omission of the article and the introduction of the interjection.

In the Middle Ages we see the rise of a philosophical exploration of grammar and, in consequence, the first versions of the concept of Universal Grammar that would play a vital role in the Generative tradition of linguistics and the exploration of cross-lingual part-of-speech systems. Up to this point in history, all grammatical systems were trying to describe a specific language (Greek, Latin, French, etc.) instead of 'Language' (in the sense of our universal ability to speak and understand speech). This can be explained by the lack of non-Indo-European (or non-Western) linguistic data, which started to become available with the advent of the great trade routes of the middle ages, as well as a flourishing scientific development in non-Western societies.

Alongside this philosophical tradition, but back in the domain of language-specific grammars, the development of the *modistic* system allowed for a connection of the morphological description of words with the syntax of the sentence (the way certain words interacted) and therefore allowed for a syntactic view of parts of speech based on the notion of governance. For the first time, but without making them a distinct class, Thomas of Erfurt in 1350 distinguishes between adjectives and nouns based on the dependence of the latter to the former since adjectives cannot exist independently of nouns in a sentence (Robins, 1969, p. 85).

This formal morphosyntactic view is fully developed in the grammar of Petrus Ramus. In writing his Latin grammar in 1548 he demanded for purely formal identi-

[1]The lack of adjectives is noticeable. Dionysius classified them as a subcategory of nouns since like common nouns (*prosēgoríe*s) they reflect attributes (e.g. being red is like being a human).

fication criteria; that is, there should be no account of the semantic properties of the various parts of speech in their definitions. So, although he kept Priscian's eight parts of speech, he relied on inflection for number and its absence to distinguish nouns, pronouns, verbs and participles from the rest. This distinction was further aided by the use of syntactic relations like concord and governance (Graves, 1912, p. 130).

There are two trends in the early approaches to part-of-speech definitions: the ones that describe morphologically rich languages (e.g. Greek, Latin) and are based on morphological properties with semantic elements as support (for distinguishing only the major categories—verbs from nouns); and the others based on syntactic/semantic definitions which used the same labels as the Greeks but ascribed semantic properties to them (Rauh, 2010, p. 28–29).

With Lindley Murray's English Grammar of 1795 we have the first account of the modern set of parts of speech: nine categories with adjectives being a distinct class. However, the definitions are less formal and are based on a mixture of semantic/pragmatic properties and syntactic rules:

1. An ARTICLE is a word prefixed to substantives to point them out, and to show how far their signification extends.

2. A SUBSTANTIVE or noun is the name of any thing that exists or of which we have any notion.
   A substantive may, in general, be distinguished by its taking an article before it, or by its making sense of itself.

3. A PRONOUN is a word used instead of a noun to avoid the too frequent repetition of the same word.

4. An ADJECTIVE is a word added to a substantive to express its quality.
   An adjective may be known by its making sense with the addition of the word thing *or of any particular substantive.*

5. A VERB is a word which signifies to BE, to DO, or to SUFFER.
   A verb may be distinguished, by its making sense with any of the personal pronouns, or the words to *before it.*

6. An ADVERB is a part of speech joined to a verb, an adjective, and sometimes to another adverb, to express some quality or circumstance respecting it.
   An adverb is generally known, by its answering to the question, How? How much? When? or Where?

7. PREPOSITIONS serve to connect words with one another, and to show the relation between them.

   A preposition may be known by its admitting after it a personal pronoun, in the objective case.

8. A CONJUNCTION is a part of speech that is chiefly used to connect or join together sentences; so as, out of two, to make one sentence. It sometimes connects only words.

9. INTERJECTIONS are words thrown in between the parts of the sentence, to express the passions or emotions of the speaker.

(Murray, 1798, p. 26–29)

We can see that these definitions lack the rigour of the formal grammarians and rely on what Murray perceived to be common sense (i.e. prototypical) uses as well linguistic tests (e.g. adding *thing* after a word to test for adjectives—itself a common–sense–based process).

These definitions are used more or less unaltered today, and Murray's nine parts of speech constitute what we would call a 'school account' of parts of speech (even though English grammar is no longer being taught as a subject in British or American schools). The English grammar book of Wren & Martin (1995) used in most Indian schools, contains eight parts of speech (article is not defined) with almost identical definitions to those given by Murray. For instance Noun is defined as "a word used as the name of a person, place or thing"; Pronoun is "a word used instead of a noun"; Verb is "a word used to express an action or state" etc. (Wren & Martin, 1995, p. 3–4).

These empirical definitions, although being intuitive and easy to learn provide little help to the linguistic enquiry. This lack of formality turns into a problem when it becomes the basis of corpus annotation and therefore evaluation, which in turn is one of the main points of the present thesis (see sections 2.3.1 and 3.3).

Moving away from the linguistically egocentric approaches of Ramus and Murray that focused on a single language, the philosophical exploration of grammar of the middle ages was taken up by the Port-Royal scholars and their *Grammaire Générale*, first published in 1660. They drew from their knowledge of Latin, Greek and Hebrew, as well as many modern European languages to create an account of a general grammar with pure philosophical reasoning at the heart of it. Accordingly their part-of-speech system re-introduced semantic distinctions of the classical nine categories dividing them into the 'objects' and the 'form' of our thought (Lancelot & Arnauld, 1975).

This need to escape the Indo-European-centric views takes full form with Franz Boas, Leonard Bloomfield and the advent of the American Structuralists. It was Bloomfield that developed for the first time a concept of syntax as a discrete level of language containing a hierarchical and linear arrangement of elements (Rauh, 2010, p. 32). With Zellig Harris and Charles Fries we move to a purely distributional view of syntactic categories. Both Harris and Fries, under the Structuralist tradition, try to capture language 'in the wild', collecting corpora and use the notion of *substitutability* as a means to discover parts of speech. The notion of substitutability is the cornerstone of unsupervised part-of-speech induction systems and so it is worth describing in detail.

> In some cases it is possible to find a set of morphemes such that each of them occurs in precisely the *total* environments in which every other one does.
>
> (Harris, 1951, p. 243, my emphasis)

Here Harris uses the term *morphemes* to refer to word and sub-word units which he treated as one and the same (allowing for a distributional account of morphology as well as syntax). The term *environment* refers to the sum of all the contexts each word occurs in. As we can see under Harris's distributional criteria, substitutability is defined as the idea that if two words share exactly the same context in a corpus of natural language utterances, they can be exchanged for each other, which means that they belong to the same class of words[2]. For instance, let us define a corpus comprised of the following utterances:

(2.1)
a. The black duck was afraid.
b. The grey duck was afraid.
c. The grey cat was afraid.
d. The small cat was afraid.
e. The small duck was afraid.
f. The duck was afraid.
g. The cat was afraid.

Under Harris's *total* environment clause, most of the words which should be members of the same class occur in the different environments: 'duck' occurs in the context of 'The ____was afraid.' and 'The {grey, black, small} ____was afraid', but 'cat' does not occur in 'The black ____was afraid'; 'grey' and 'small' share all their environments but not with 'black'. Harris recognised that it is difficult to find words occurring

[2]This definition was recently formalised by Clark (2010) in the context of grammar learning.

in identical environments so he used two methods to relax this requirement. The first was that the two need to share at least 80% of their contexts[3]. The second method involved replacing word tokens with their part-of-speech labels. Using these methods, and by substituting 'grey' and 'small' with X_1 (since they share all their environments), the words 'cat' and 'duck' now share 85.71% of their environments ('duck' still has the extra 'The black ____was afraid' context) and therefore belong to the same class. If we substitute them with another label X_2, black can now be added to X_1 since it shares all its environments with the other words in that class. The rest of the words now share the same context and can be classified accordingly.

Using this method Harris defines 18 parts of speech for English (of which 11 were major categories and seven were subcategories of verbs). Note here that Harris treated words as morphemes, separating inflectional affixes from stems (he defined 16 morphological *affix classes*) and most of his definitions contain morphological elements that are treated exactly as distributional properties. For example his definition of noun is 'morphemes that occur before plural -s or its alternants, or after *the* or adjectives' (Harris, 1946).

Despite his definitions being purely distributional we can detect elements of semantic distinctions since the notion of substitutability depends on corpora of utterances. This means that syntactically plausible but semantically unsound substitutions will not be present since the speakers of the language in question would never utter those sentences. For instance although 'grey' is an adjective and 'idea' is a noun the following utterance cannot occur[4]:

(2.2) The grey idea was afraid.

Chomsky (1957) brings a new view of syntactic categories under the *phrase-structure* rules of the Generative Grammar. In addition to the 'major' lexical categories (noun, verb, adjective, particle, pronoun and adverb) we find non-lexical, purely syntactic categories such as NP, VP and PP. The categories are not linked to either semantic or morphological properties but instead are introduced by the phrase-structure rules. The same holds for the new categories (and subcategorizations) introduced in the model of Chomsky (1965).

[3] With the advent of empirical methods for part-of-speech induction this condition has been further relaxed to a narrow context window of at most 3-4 words.

[4] Under a *very* large corpus of utterances even this example might occur but still this will have a low probability; a Google search for "the grey idea was" yielded eight results, compared to "the grey cat was" yielding about 198,000 results.

	Subject	Object	Complement	Determiner
verb	+	+	+	
modal	+	+	-	
preposition	-	+	+	
particle	-	+	-	
noun	+	-	+	
article	+	-	-	+
quantifier	+	-	-	-
adjective	-	-	+	
degree	-	-	-	+
adverb	-	-	-	-

Table 2.1: List of lexical categories using the X-bar feature-based categorisation [source: Jackendoff (1977, p. 33)]

The next major step in the Generative Grammar tradition of category identification comes with X-bar theory and the feature-based representation of parts of speech. First Chomsky in his *Amherst Lectures* defined the three major X categories (noun, verb, adjective) in terms of ±N (nominal) and ±V (predicative) features and then Jackendoff (1977, p. 33) introduced a new set of features (±Subject, ±Object, ±Complement, ±Determiner) and applied them to describe 10 categories as shown in table 2.1.

Extensions and refinement to the feature-based classification system of the X-bar theory include functional features starting with Abney (1987) which led to a distinction between lexical and functional categories. However, some of the distinctions of functional features include semantic evaluations, for example Cinque's adverb split by Mood, Aspect and Tense (Cinque, 1999, p. 106).

The period between the late 70s and early 80s marks a major divide between the followers of Chomskyan view of language, with syntax at its core (known as *formalists*) and *functionalist* approaches[5]. The main goal of the functionalists is to restore the semantics as the basis for grammar and therefore describe parts of speech or syntactic categories using semantic criteria. They focus heavily on viewing parts of speech under a *typological* (cross-lingual) perspective and use the notion of prototypical members of categories, similarly to the Port-Royal scholars and their *Grammaire Générale*.

[5]These two camps are also called West Coast (University of California) and East Coast (MIT) linguistics.

Plato SE	**Aristotle** SE+M	**Stoics** SE+M	**Dionysius** SY+SE+M	**Priscian** SY+SE+M	**Varo** M	**Murray** SY+SE	**Harris** D	**Jackendoff** SY	**Croft** SE
noun	noun	common noun	noun	noun	noun	noun	noun (N)	noun	noun
verb	verb	verb	verb	verb	verb	verb	verb (V)	verb	verb
	conjunction	conjunction	conjunction	conjunction		conjunction	conjunction (&)		
		pronoun	pronoun	pronoun		pronoun	pronoun (I)		
		proper name							
			participle	participle	participle				
			preposition	preposition		preposition	preposition (P, I)	preposition	
			adverb	adverb	adverb	adverb	adverb (D)	adverb	
			article			article	article (T)	article	
				interjection		interjection			
						adjective	adjective (A)	adjective	adjective
							particle (B)	particle	
							modal (R)	modal	
							have, *be*		
								degree	
								quantifier	

Legend
SE=semantic
SY=syntactic
M=morphological
D=distributional

Table 2.2: Overview and comparison of major historical part-of-speech systems

Dixon (1977) presents the notion of *prototypes*. He defines using cognitive criteria the notion of typical adjectives, which correspond (not intentionally) with the semantic notion of adjective. This view seems to be validated by language acquisition experiments where young children will classify new instances of actions to the verb category (i.e. use them in verb-like constructions) and new instances of objects to the noun category (Brown, 1958, p. 247–52)[6].

Croft (1991) extends the prototype theory with typological universals in mind, but confines himself to defining only the 'fundamental' grammatical categories (noun, verb, adjective). In his view, parts of speech should be distinguished by their pragmatic role (or discourse function—Reference, Modifications, Predication), as well as their semantic class.

An even stronger case for discourse criteria as a primary source of distinction is made by Hopper & Thompson (1984). They agree to a 'universal correlation' that prototypical 'thing-like entities' tend be coded as nouns while actions will be coded as verbs but they assert that their semantic nature is rooted in discourse functions. They define prototypical nouns as word forms that "serve to introduce a participant to a discourse" and verbs as forms that "assert the occurrence of an event of the discourse".

Susan Schmerling defined syntactic categories by formal semantic (Montagovian)

[6] A similar view of parts of speech has been used in the *prototype-driven learning* system of Haghighi & Klein (2006), presented in section 3.4.1, although their work did not make the connection to cognitively plausible categories.

terms. Under these terms, a category can be thought of as a (first order) logic function that receives inputs and returns outputs. For instance $\langle e,t \rangle$—a category that receives entities (like objects) and returns truth values—defines nouns, adjectives and intransitive verbs (Schmerling, 1983).

Schachter (1985) suggests a semantic heuristic for labelling parts of speech across languages using the *notional* definition of categories:

> (1) Nouns denote persons, places or things.
> (2) Adjectives denote properties/qualities.
> (3) Verbs denote actions/events.
>
> [source: Croft (2000)]

However, he argues that grammatical criteria must be employed for their identification. By grammatical Schacter refers to a mixture of distributional morphological and syntactic criteria:

(2.3) Boys like girls.

In this example 'boys' and 'like' differ distributionally (under Harris's definition). They also differ in that 'boys' is specified for number but not tense but 'like' is specified for both. They finally differ in their syntactic function: 'boys' is the subject of, or controlled by 'like'. On the other hand 'boys' is similar to 'girls' morphologically and distributionally but not syntactically ('boys' is the subject, 'girls' the object).

One common characteristic of both formalist and functionalist approaches is their emphasis on the major parts of speech (noun, verb, adjective) as being truly universal while some of them will describe language-specific minor categories or subcategories of the major ones.

As a conclusion to this section table 2.2 presents a comparison of all the major part-of-speech systems discussed here.

2.3 Part-of-Speech Tagging and Tagsets

We will look now more at the computational approaches to language and discuss the evolution of part-of-speech labels and automatic part-of-speech tagging systems that shaped the field of computational linguistics and set the ground for the unsupervised induction of parts of speech discussed in the next chapter.

2.3.1 Parts of Speech in Corpora

The beginning of *corpus linguistics* marked a new era in the analysis and categorisation of parts of speech as well as the beginning of the area of NLP. The Brown Corpus developed by Henry Kučera and W. Nelson Francis in the early 1960s was the first attempt to collect and compile a corpus of natural language with the intention of being used for the analysis of grammar (Francis, 1964). Despite the early attempts of the Structuralists, the study of English with the use of computational analysis of corpora was very radical. As Kučera (1992, p. 402) describes, they were met with scepticism and sometimes hostility by the adopters of the Chomskyan tradition, where the analysis of actual linguistic data was considered far less useful than the intuitions of a native speaker of English[7].

Part-of-speech *tagging* refers to the annotation of the text with part-of-speech labels (tags). The part-of-speech tagging of a portion of the Brown Corpus by Greene and Rubin finished in 1971 (Greene & Rubin, 1971). They used a set of 77 individual tags but combined them to produce a more fine-grained set of 226 tags (or *tagset*; see table A.1, appendix A). Greene and Rubin also pioneered the semi-automatic annotation of the corpus using the TAGGIT system described in section 2.3.2.

The next big annotation project and development of a new annotation scheme was the Lancaster-Oslo-Bergen (LOB) corpus—the British equivalent (in both size and genres) of the Brown corpus (Marshall, 1983). They used 153 individual tags—a refined version of the Brown tagset—and a probabilistic tagger (CLAWS, described below).

The SUSANNE corpus (Sampson, 1995) expanded the Brown tagset even further to include morphological, semantic and pragmatic distinctions to a total of 356 tags. The SUSANNE tagset, shown for reference in table A.3, contains extremely fine-grained distinctions, like two different types or equations (chemical: FOqc and other: FOqx), a tag specifically for UK or US postcodes (FOp), feminine forenames (NP1f), base forms of transitive (VV0t) and intransitive verbs (VV0i) and a different tag for each gender, number and case of the personal pronouns.

Undoubtedly the most influential corpus in NLP has been the Penn Treebank (PTB, Marcus et al., 1993). The PTB tagset was a coarser version of the Brown tagset and contained 48 tags of which 36 are part-of-speech tags and 12 for handling punctuation

[7] Interestingly we have now come full circle back to this idea with rule-guided semi-supervised NLP systems, for example in Naseem et al. (2010).

and currency symbols.

Other part-of-speech-tagged corpora include the International Corpus of English (ICE) containing 205 tags (Greenbaum, 1993) and the Polytechnic of Wales corpus (POW, Souter, 1989) with 66 tags.

One common thread to all the tagging approaches is that tagging was always viewed as a pre-processing step to syntactic parsing. This was clearly stated by the creators of the Brown corpus tagset:

> Since the purpose of the tagged corpus is to facilitate automatic or semi-automatic syntactic analysis, the rationale of the tagging system is basically syntactic, though some morphological distinctions with little or no syntactic significance have also been recognised.
>
> (Francis & Kučera, 1964)

This led the annotators to employ engineering criteria rather than adhere to a specific linguistic theory and under-/over-specified their part-of-speech labels accordingly. This is especially obvious in the case of the SUSANNE tagset, where the distinctions are so fine-grained that the syntactic structure of the sentence is almost unambiguous after the tagging stage. Also clear from the annotation guidelines is the emphasis on the intuitive (semantic/pragmatic) nature of the labels with the use of examples for exposing difficult cases:

> Since the parts of speech are probably familiar to you from high-school English, you should have little difficulty in assimilating the tags themselves. However, it is often quite difficult to decide which tag is appropriate in a particular context. The two sections 4 and 5 therefore include examples and guidelines on how to tag problematic cases.
>
> (Santorini, 1990)

Actually, if we convert the 36-tagset (excluding symbols) of the PTB to a *logical tagset* (Leech, 1997, p. 27) we can see that it contains 11 main categories (conjunction, numeral, existential, preposition, adjective, noun, determiner, pronoun, adverb, particle, verb)[8] seven of which are in Dionysius Thrax's original set (table 2.2).

These pragmatically-driven annotation approaches have indeed been proven useful for the task of syntactic analysis *parsing*—and in fact for supervised part-of-speech tagging—but leave us with the problem of category sets that are not easily derivable from text alone (i.e. in an unsupervised fashion), using any of the linguistic theories discussed earlier. We will return to this problem when we discuss evaluation methods for unsupervised systems in section 3.3.4.

[8]The other categories are: foreign word, list item marker, genitive marker, and the various symbols.

The latest development in corpus-driven linguistics is the use of *parallel* corpora—collections of texts containing the same utterances translated into multiple languages (for example the proceedings of the European Parliament). Parallel (or comparable[9]) corpora are used not just for machine translation, but for the discovery of linguistic structure, for example see Naseem et al. (2010); Cohen et al. (2011). To facilitate these efforts, large amounts of multilingual corpora had to be annotated with the same labels (e.g. the MULTEXT-East corpus of Erjavec, 2004) or their annotations had to be converted to a 'universal' representation (e.g. the Universal Tagset of Petrov et al. 2011). A similar attempt is the creation of a coarser set of 17 tags for the WSJ portion of the PTB by Smith & Eisner (2005a), using a process similar to the logical tagset of Leech (1997, p. 27) discussed earlier.

The effect of these approaches has been similar to the attempts of the structuralists, namely a reduction in specificity to account for cross-lingual differences. Indeed the MULTEXT-East tagset contains only 14 tags, 11 of which are used in all languages, which is the same number of tags contained in the Universal Tagset of Petrov et al. (2011).

2.3.2 Supervised Part of Speech Tagging Systems

The first attempt to build an automatic part-of-speech tagging system coincided with the creation of the tagged version of the Brown corpus. Greene & Rubin (1971) used the TAGGIT system, a rule-based disambiguation tagger, as a means to automate the annotation process. The system had access to a lexicon and a suffix list which it could use as look-up tables and come up with a number of candidate tags for each word. Then it would proceed to eliminate all but one of the candidate tags by using Context Frame rules. These were manually created by linguists, based on observations of ± 3 context words. TAGGIT, using the Context Frame rules, could successfully disambiguate 77% of the words in the corpus; the rest were manually disambiguated by the linguists.

The introduction of probabilistic systems (first introduced in speech recognition) brought a revolution in part-of-speech tagging, dramatically increasing the performance of the tagging systems. One of the first probabilistic systems was the Constituent Likelihood Automatic Word-tagging System (CLAWS) developed for the LOB corpus (Marshall, 1983). Tagging with CLAWS consisted of three stages[10]: Initial tag

[9] A comparable corpus, while not containing parallel texts, contains texts of similar style and structure across multiple languages.

[10] There is a pre-processing stage of tokenization but this is of little importance in the tagging process.

assignment and tag disambiguation, which were the main probabilistic elements, and idiom-tagging, which was a rule-based step. During initial tag assignment, the tagger would assign each word a list of tags with some probability score (from a lexicon lookup); after that the tag disambiguation stage would choose a 'winning tag' from the list of possible tags. At this point the accuracy of the tagger was about 96%, a major improvement over TAGGIT's automatic stages. Finally, the idiom-tagging stage would use manually created rules to re-tag idiomatic cases such as multi-word-expressions or place name expressions.

For the tag disambiguation stage of CLAWS, Marshall used a probability model that was an approximation to a Hidden Markov Model (HMM, Rabiner, 1989). A full version of the HMM was used by PARTS tagger of Church (1988) in the semi-automatic part-of-speech annotation of the PTB. Since then, the HMM has been used extensively by Merialdo (1994); Weischedel et al. (1993); Schütze & Singer (1994) and Brants (2000), among others.

A notable rule-based tagger from the 90s was that of Brill (1992). It used a very simple probability model to assign the most frequent tag to a word irrespective of the context (a unigram probability model) and then used hand-crafted rules capturing features from the context of the word to correct the tagging. The rules used features from ± 3 context words in a similar style to the TAGGIT system. Brill's tagger challenged the growing notion that probabilistic systems always outperformed rule-based ones (Charniak, 1997) and in fact most probabilistic systems ever since have a 'rule-based component' or heuristics to help with their tagging (for instance see Ratnaparkhi et al., 1996; Daelemans et al., 1996 and Carlberger & Kann, 1999).

More recently linear and log-linear feature-based models have started to be used extensively producing state-of-the-art results: The Stanford Tagger of Manning (2011) (using a model developed originally by Toutanova & Manning, 2000 and Toutanova et al., 2003) and the tagger of Shen et al. (2007) have achieved an accuracy of over 97.3% on the WSJ corpus.

There are two remarks that will put the performance of part-of-speech tagging systems into perspective. The first is that, as Charniak (1997) points out, simply assigning the most common tag to each known word and the tag 'proper noun' to all unknowns will yield a 90% accuracy (compared against the annotation of the WSJ corpus). The second is that the *inter-annotator agreement* for English is about 98% (Baker, 1997, p. 243). This is a not only theoretical upper limit to the performance of any supervised system trained on human annotations but also an upper limit for any evaluation based

on a single gold-standard. This means that, at least when gold-standard annotations are provided as a training corpus, the task of part-of-speech tagging is effectively solved.

However, these numbers, and supervised part-of-speech tagging as a task obscure two problems. First, all of the corpora and systems described above are designed on English; while it is true that there are now part-of-speech annotated corpora (and therefore trained taggers) for most of the major languages, the effort and cost to annotate a new corpus are prohibitive for most resource-poor languages. Second, it is not necessarily true that even the gold-standard annotations provide the best account of what parts of speech are—indeed as we have seen the annotation guidelines are far from rigorous—and there are cases where induced parts of speech will outperform gold-standard tags in downstream tasks (Spitkovsky et al., 2011a).

2.4 Conclusion

The historical evolution of part-of-speech systems has taken us from semantics, to morphology, to syntax and back, and from highly specialised, linguistically egocentric definitions, to cognitively driven universals. While there is no consensus about the definition of parts of speech, most definitions agree on the fact that multiple sources of features (morphological, syntactic, semantic) are required. This was one of the main goals of this chapter: not to find a conclusive definition of parts of speech, but to recognise their multidimensionality and their interdependence with other levels of linguistic description—something that our NLP should try and capture. This is the main emphasis of chapters 5, 6 and 7.

From a computational perspective, parts of speech have been viewed as facilitators of parsing and have only recently started to take their own place in computational linguistic research. This change has been facilitated by the appeal of part-of-speech tagging as a stand-alone, well-defined testbed for machine learning techniques, as well as by the rise of unsupervised methods in general and for part-of-speech induction in particular which will be examined in the next chapter.

CHAPTER 3

Unsupervised Part-of-Speech Induction

The problem of induction is not a problem of demonstration but a problem of defining the difference between valid and invalid predictions.

Goodman (1983, p. 65)

In the last two decades, alongside supervised systems in NLP, there has been an increasing interest in unsupervised methods. Broadly speaking, we describe as unsupervised any type of learning that does not rely on annotated examples of the type of structure to be learnt (see next section for a more detailed definition). This type of learning is appealing for several reasons. Firstly, annotating a corpus is very expensive: as Marcus (2011) reports, the proposed total cost of the Penn Treebank was about $10 million[1]. Secondly, it is not easy to get annotated examples for many languages. This is related to the cost of the annotation, but also to the availability of expert annotators for certain languages. A final reason, related to the annotation process is that you have to know what you are looking for before you start. That is, there might be regularities in the data that can be discovered by unsupervised learning that were overlooked by human annotators.

Part-of-speech induction is particularly attractive to the unsupervised learning community, since it is a straightforward self-contained task with enough gold-standard data to evaluate against (at least in English). However, there is little consensus about

[1] Given that Fred Jelinek's original proposal was submitted to DARPA in 1987, the cost of the project today (adjusted for inflation) is more like $20 million.

evaluation methods, which makes direct comparison of the various unsupervised part-of-speech induction systems very difficult. A discussion about evaluation metrics is presented in section 3.2, and a quantitative comparison is presented in section 3.3.6.

There have been a few simple stochastic unsupervised learning systems for part-of-speech induction in the past decade, but recently many sophisticated machine learning algorithms have been applied to this task. In section 3.4.1, I describe a number of part-of-speech induction systems and present a direct quantitative comparison in section 3.4.4. An interesting thing to note here is that unlike the supervised task of part-of-speech tagging, most of these approaches follow one particular linguistic theory—that of Zellig Harris, presented in section 2.2. This is because the distributional theory proposed by Harris is the most straightforward to translate to computational terms, even though the distributional models examined in this chapter use a very limited notion of *environment* (usually a 1 or 2 word window) and replace the notion of *morphemes* with that of word tokens.

Most of the work described in these sections has been previously published in Christodoulopoulos et al. (2010); however, I will also include some systems that were not covered in the original review.

Some notes on terminology: I will be using the term 'tag' to refer to a gold-standard label and the term 'cluster' or 'class' to refer to the part-of-speech induction system's output. I will also be using the terms 'system', 'model', 'method', and 'technique' interchangeably unless a clear distinction is needed.

3.1 Unsupervised vs. Fully Unsupervised

There is a spectrum of approaches between fully supervised and fully unsupervised which relates to the amount of external knowledge that is required by any given system.

The term *induction*, borrowed from logic and statistical reasoning, is used in NLP to emphasise the unsupervised nature of a task. Here, I use it with a more restrictive sense that covers only *fully unsupervised* systems. Since no external source of knowledge is used, the induced labels are arbitrary symbols (usually numbers) and unless a matching is forced they bear no resemblance to traditional part-of-speech tags. This is a crucial difference between *fully* unsupervised systems and unsupervised systems that use some kind of external knowledge.

Before we go into an overview of fully unsupervised part-of-speech induction systems, I present in table 3.1 a list of techniques that do not require any manual annotation

Technique	External resources used	Example systems
disambiguation	lexicon of allowed tags/word	Merialdo (1994)
disamb. w/ dilution	lexicon for most freq. words	Goldwater & Griffiths (2007)
prototype-driven	list of prototypes for each tag	Haghighi & Klein (2006)
projected	supervision in another language	Yarowsky & Ngai (2001)

Table 3.1: List of non-supervised or resource-light techniques and sources of external knowledge used.

of training examples—and hence are not supervised—but rely on various other sources of external (or prior) knowledge. In the first category of *disambiguation* techniques we have systems that use a lexicon containing a list of tags that each word type can take. The job of the system is to determine which of these tags to assign to a given token (Merialdo, 1994). The degree of external knowledge can change by diluting the lexicon (Smith & Eisner, 2005a; Goldwater & Griffiths, 2007): including only a certain fraction of the most frequent words and allowing all possible tags for the rest of the words. In the extreme case of lexicon dilution (Goldwater & Griffiths, 2007), where every word can take every tag, the amount of external knowledge is reduced to a minimum; however the complete *tagset* needs to be known and therefore this method is not fully unsupervised.

Perhaps a more meaningful distinction could be one between resource-heavy and resource-light approaches, with fully unsupervised systems classified as resource-less. See Hana & Feldman (2012) for a review of morphological analysis and tagging under this distinction.

A noticeable omission from table 3.1 is *semi-supervised* techniques. Under a strict definition (like for instance in Chapelle et al., 2006, p. 2) semi-supervised learning is any technique that is provided with a mixture of unlabelled data and some labelled data but only for a subset of the examples. This is a very active area of NLP with work on both in-domain (e.g. Huang et al., 2010) and domain-adaptation (e.g. Petrov & McDonald, 2012) tasks. Recently, Garrette & Baldridge (2013) and Garrette et al. (2013) have presented a semi-supervised approach where the amount of annotation is quantified, providing a trade-off between the amount of annotation needed and the quality of the part-of-speech tagging system. However, since these techniques rely on at least some amount of annotated data they are not discussed further here[2].

[2]Boonkwan & Steedman (2011) take a similar, resource-light approach for grammar induction that

The last entry of table 3.1 is projected learning. In this approach, supervised data from a resource-rich language is used to guide the unsupervised learning algorithm in a target language. This guidance can be projected directly through parallel corpora (e.g. Yarowsky & Ngai, 2001; Das & Petrov, 2011) or by constraining the learning parameters of the target language via comparable (but not parallel) corpora (e.g. Cohen et al., 2011; Li et al., 2012). Even though under the strict definition given above it can be regarded as a type of semi-supervised learning, projected learning requires no external-knowledge resources in the target language. In this respect it is similar to fully unsupervised methods.

Henceforth I will be using the term 'unsupervised part-of-speech induction' or simply 'part-of-speech induction' to refer to the fully unsupervised kind, which requires no external knowledge and is equivalent to word clustering.

Before closing this section it is worth mentioning that even fully unsupervised systems contain external knowledge in some form. Most of the systems that will be discussed in this and the following chapters will contain some hand-coded learning bias[3] or modelling assumptions. Furthermore, systems that have any manually-set parameters are subject to biases introduced by their development data or language. It is difficult to avoid any form of bias when designing a system, but different systems will use different biases and it is worth examining them during the following discussion.

3.2 Evaluation of Unsupervised Systems

Evaluation is a crucial part of NLP systems. Broadly speaking, there are two types of evaluation: *intrinsic* and *extrinsic*. Intrinsic methods evaluate the output of the system directly, comparing it to some manually annotated version of the test data by an expert, also known as *gold-standard* annotations. This is what Smith & Eisner (2005b) call MATCHLINGUIST. Extrinsic evaluation, on the other hand, refers to methods that evaluate the output of the system by evaluating another system that (at least partially) relies on the first one. The obvious advantage of the intrinsic method is that once the annotation is created it can be reused by different systems (provided they use the same test data), making comparisons between systems straightforward. There are of course

while requiring no explicit annotation, relies on a questionnaire to elicit language-specific syntactic constraints.

[3]The term learning (or inductive) bias has a very general definition in supervised machine learning model: 'any basis for choosing one generalization over another, other than strict consistency with the observed training instances' (Mitchell, 1980). Here I will use the term more loosely to include all engineering, development and tuning decisions that influence the inference of a statistical model.

disadvantages of intrinsic evaluation, but before I discuss them I will briefly present an overview of extrinsic evaluation methods.

3.2.1 Extrinsic Evaluation

In its most general sense, extrinsic evaluation includes any evaluation technique that does not require gold-standard annotated test data for the task under evaluation. Given the plethora of alternative intrinsic evaluation methods and the inherent problems that each of them has (discussed later), many researchers have turned to extrinsic evaluation methods.

One of the most common extrinsic evaluation methods is to evaluate the output of a system in a *downstream* task. The term downstream implies an inherent directionality of the *pipeline* approach in NLP (see figure 1.1), where the output of a system is used as input or part of the input for a system that performs a (usually) more complicated task. For instance, part-of-speech tagging can be used as input to a dependency induction system. We will examine the pipeline approach and its implications later, in chapter 5. Under this evaluation regime the performance of the first system is indirectly measured by evaluating the downstream system most often using intrinsic evaluation methods.

Apart from being more time consuming and therefore less practical, extrinsic evaluation on a downstream task suffers from two main problems. The first comes from the fact that if the downstream system is evaluated intrinsically, we have simply deferred the same problems discussed earlier to this new task. Of course for some downstream tasks these problems will be less prominent, perhaps because of more consistent annotation or better understanding of the linguistic area in question but in any case—and this is only true for unsupervised systems—we should not require our unsupervised computational models to predict the same kind of structure as a human expert.

The second, and most important problem is that even if we can accurately evaluate the performance on a downstream task, that performance might not be correlated with the performance of the first system. Headden et al. (2008) examined various mapping and information-theoretic part-of-speech induction metrics (including many-to-1, 1-to-1 and VI, see section 3.3) and their correlation to dependency parsing scores (directed/undirected accuracy, see section 5.2.5) when the two systems are used in the pipeline approach. The authors showed that none of the standard part-of-speech induction metrics correlates with the performance of the dependency parsing system under Kendall's τ significance test.

A similar situation exists in larger systems with multiple input components such as systems in statistical machine translation. Ganchev et al. (2008) examine an agreement-based word alignment system, both with intrinsic evaluation and as part of a machine translation system. Even though they show that certain configurations yield significantly better performance under every intrinsic metric in word alignment, those performance gains do not translate to equal performance gains in machine translation scores.

This does not necessarily mean that we should not trust the evaluation of the downstream task (even though it too might suffer from the problems of intrinsic evaluation discussed later), since we might be more interested in the downstream performance anyway. However, as a means of evaluating the current task its application seems limited.

A new kind of extrinsic evaluation has been suggested recently by Smith (2012). He proposes an evaluation based on real world data with no external annotation required. I will briefly present here the main points of this evaluation method for part-of-speech induction, even though the same method can be applied to any NLP task given appropriate changes. In Smith's *adversarial* evaluation framework, there are two components[4]: the first is called *The Transformer of Data* which receives a real-world sentence (a blog post, a news report, etc.) and creates a copy with a specific—linguistically motivated—corruption. For instance, it could replace an adjective with a noun. The role of the second system, named *The Chooser* is to identify which of the two copies of the sentence is better according to some internal—supervised, unsupervised or rule-based—model. The evaluation of multiple systems is straightforward: all we need to do is keep the same *Transformer* and evaluate the accuracy of different *Choosers* on the same set of sentences. Note that with this setup it is equally straightforward to evaluate different *Transformers* by keeping the same *Chooser*. This adversarial evaluation is a theoretically interesting idea but since it has not been put into practice yet it is difficult to tell whether it can actually overcome all the problems with our current models of evaluation.

We will return to extrinsic evaluation methods in chapter 5; for the comparison of the part-of-speech induction systems I will only use intrinsic evaluation metrics, since using extrinsic evaluation would complicate the analysis beyond the intended overview of the area.

[4]Smith eventually extends the framework to include three systems, the third being the data selection and meta-data annotation system. For the purposes of this brief exposition I will focus on the first two systems.

3.3 Intrinsic Evaluation

Assuming the existence of gold-standard (hand-annotated) data, intrinsic evaluation for part-of-speech induction can be performed in two ways: either by enforcing a mapping between the output of the induction system and the set of gold-standard tags, or by using *information-theoretic* metrics to compare the clusterings of the inducer output and the gold-standard.

What follows is a short presentation of a number of mapping and information-theoretic metrics that have been proposed in the literature. In square brackets are the shorthands used in the evaluation section (3.4.4).

Throughout this section I will be using T to refer to the set of gold-standard tags, C to the set of induced clusters and $|\cdot|$ to the size of the set.

3.3.1 Mapping metrics

3.3.1.1 [m-1]: Many-to-one mapping accuracy

In many-to-one accuracy (also known as *cluster purity*), each cluster is mapped to the gold standard tag that is most common for the words in that cluster (henceforth, the *preferred tag*), and then the proportion of words tagged correctly is computed. More than one cluster may be mapped to the same gold standard tag. This is the most commonly used metric across the literature as it is intuitive and creates a meaningful part-of-speech sequence out of the cluster identifiers. Many-to-one mapping is also useful for tagging corpora for downstream tasks that depend on specific tagset labels (for example, a parser trained on the Penn Treebank will need to have part-of-speech information based on the Penn Treebank tagset). However, as we will see in section 3.3.6, it tends to yield higher scores as $|C|$ increases (reaching 100% when every word has its own tag), making comparisons difficult when $|C|$ can vary.

3.3.1.2 [crossval]: Cross-validation accuracy

This metric, first proposed by Gao & Johnson (2008), was intended to address the problem with many-to-one accuracy that assigning each word to its own class yields a perfect score. In this measure, the first half of the corpus is used to obtain the many-to-one mapping of clusters to tags, and this mapping is used to compute the accuracy of the clustering on the second half of the corpus. However, this metric suffers from the

same problem as **m-1**, since the mapping created on the first half would be influence by $|C|$.

3.3.1.3 [1-to-1]: One-to-one mapping accuracy

Unlike many-to-one and cross-validation, one-to-one constrains the mapping from clusters to tags, so that at most one cluster can be mapped to any tag. The mapping is performed greedily—that is, each cluster will always be mapped to the first available preferred tag without considering a globally optimal mapping. In general, as the number of clusters increases, fewer clusters will be mapped to their preferred tag and scores will decrease (especially if the number of clusters is larger than the number of tags, so that some clusters are unassigned and receive zero credit). Again, this makes it difficult to compare solutions with different values of $|C|$.

3.3.2 Information-theoretic metrics

Information-theoretic metrics begin with the assumption that *clusterings*[5] are discrete random variables where each word is tagged with a label $x \in T$ where T is the tagset (or set of cluster IDs C in the unsupervised case). They then use the concept of *entropy* $H(X)$ as introduced in information theory by Shannon (1948) to describe the amount of uncertainty within clustering X[6].

$$H(X) = -\sum_{x \in T} \tilde{p}(x) \log \tilde{p}(x) \tag{3.1}$$

Note here that we use the empirical probability $\tilde{p}(x)$ (#words labelled with x/#total words) as an approximation of the true probability. Under this definition, it is easy to see that entropy agrees with an intuitive notion of what an information measure should be: a clustering where all the words belong to the same cluster has the lowest entropy whereas a clustering in which all words belong to different clusters has maximum entropy.

Once entropy is defined we are interested in examining the amount of similarity between two clusterings; that is the amount of overlapping information that is captured by each cluster. For this we need to define *conditional entropy* $H(Y|X)$, the amount of information needed to describe clustering Y given all the information that we have

[5] I use the word clustering to refer to the collection of all the different clusters.

[6] Note that *log* is short for log_2 and not log_{10}. This form of the entropy equation is commonly used, so I will keep it for reasons of consistency.

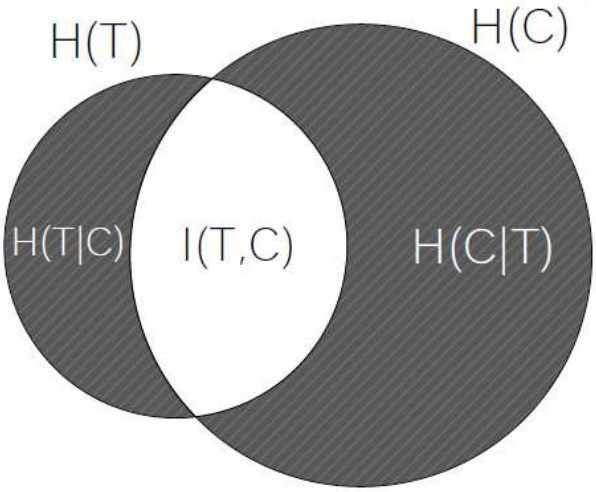


Figure 3.1: Diagram of cluster entropy (circles), conditional entropy (shaded parts), and mutual information (intersection)

about X and *mutual information* $I(X,Y)$ which is the measure of the amount of information that each clustering contains about the other. These are defined as follows:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} \tilde{p}(x,y) \log \frac{\tilde{p}(x,y)}{\tilde{p}(x)\tilde{p}(y)} \tag{3.2}$$

$$H(Y|X) = H(Y) - I(X,Y) \tag{3.3}$$

where $\tilde{p}(x,y)$ is the co-occurrence of x and y (#word tagged with x in X and y in Y/#total words). Figure 3.1 shows the relationship between entropy, conditional entropy and mutual information.

Importantly, information-theoretic metrics, by abstracting away the cluster labels and instead comparing the relative amount of information captured by the clusterings, provide an excellent solution to the problem of having to map the cluster IDs onto part-of-speech tags and also allow for direct comparison of differently sized clusterings (i.e. different number of labels).

3.3.2.1 [vi]: Variation of Information

This is an information-theoretic metric proposed by Meilă (2003). Variation of Information (VI) regards the output of the unsupervised model and the gold-standard part-of-speech tags as two separate clusterings. The quality of the unsupervised clustering is then evaluated by summing the conditional entropy (equation 3.3) of the tag

clustering given the unsupervised clusters and conditional entropy of the clusters given the tags. More formally

$$VI = H(T|C) + H(C|T) \tag{3.4}$$

3.3.2.2 [vm]: V-Measure

Proposed by Rosenberg & Hirschberg (2007), V-Measure (VM) is another entropy-based measure that is designed to be analogous to F-measure (or F-score, used in Information Retrieval), in that it is defined as the weighted harmonic mean of two values, *homogeneity* (h, the precision analogue) and *completeness* (c, the recall analogue):

$$h = 1 - \frac{H(T|C)}{H(T)} \tag{3.5}$$

$$c = 1 - \frac{H(C|T)}{H(C)} \tag{3.6}$$

$$VM = \frac{(1+\beta)hc}{(\beta h)+c} \tag{3.7}$$

As with F-measure, β is normally set to 1.

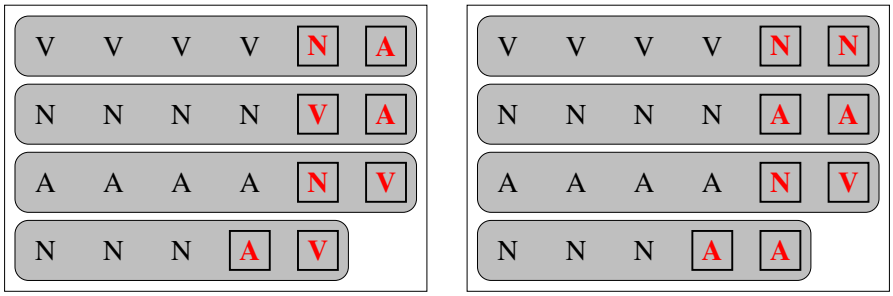
Intuitively, homogeneity means that each cluster should contain as few different tags as possible and completeness that each tag should be contained in only a few clusters.

3.3.2.3 [vmb]: V-beta

V_β is an extension to V-Measure, proposed by Vlachos et al. (2009). They noted that VM favours clusterings where the number of clusters $|C|$ is larger than the number of part-of-speech tags $|T|$. To address this issue the parameter β in equation 3.7 is set to $|C|/|T|$ in order adjust the balance between homogeneity and completeness.

3.3.3 Comparison of mapping and information-theoretic metrics

To get an idea on how these metrics differ consider the example illustrated in figure 3.2. We have a corpus of 23 word tokens that are labelled with their gold-standard tags (**V**erb, **N**oun or **A**djective) and we want to compare the output of three different clustering systems. For the mapping metrics the clusters are going to assume the label of the most frequently occurring tag (so cluster 1 in 3.2a will be labelled as **V**, cluster 2 as **N** and so on). Under **m-1** there is no way to distinguish between the systems of 3.2a



(a) (b)

Figure 3.2: Two example clustering outputs. Each row represents an induced cluster with every token being labelled with its gold-standard tag. Incorrectly tagged tokens are highlighted.

and 3.2b as they both have $15/23$ correctly tagged tokens (a score of 65.2%)[7].

Both **vi** and **vm** give better scores to the 3.2b system, since the clusters are more homogeneous. However the relative improvement is easier to interpret using **vm** where the first system scores 16.6% and the second 31.2%; the **vi** scores are 2.98 and 2.45 for the two systems respectively and although there is an improvement (lower scores are better) it is hard to quantify as a proportion.

3.3.4 Problems of gold-standard based metrics

Intrinsic evaluation methods suffer from two major problems. The first is endemic to unsupervised methods and has to do with the lack of any meaningful output labels. In the case of part-of-speech induction the 'tags' are arbitrary numbers (corresponding to cluster IDs) with no correspondence to gold-standard labels. This is particularly problematic for the map-based metrics in cases where the number of induced clusters is greater than that of the gold-standard tagset (see section 3.3.6). In other NLP tasks such as morphology segmentation or dependency parsing this not a problem since the output of the inducers is comparable with the gold standard even without labels. For instance, the accuracy of a segmentation system can be calculated without labelling the segments as STEM, PREFIX or SUFFIX.

The second problem is more general and applies to both supervised and unsupervised systems. It is not necessarily true that by maximising the agreement with the expert's annotation (MATCHLINGUIST) the output of the system in question will be

[7]The same applies to the **1-to-1** metric, although since we cannot use the **N** label twice the score for both cases is 52.2%

optimal for any downstream application. This could be because the annotators are following a specific linguistic convention that does not convey all the necessary distinctions required by another linguistic task[8]. As we saw in section 2.3.1, in the case of part-of-speech tagging there are a number of annotation schemes, using different tagsets that encode different levels of morphological, syntactic and semantic information.

For instance the SUSANNE tagset (Sampson, 1995) contains 356 distinct tags that are meant to allow the "retrieval of all important grammatical distinctions in language" (Sampson, 1995, p. 29); in other words, to convey the syntactic roles of each word to a parser without relying on context or morphology. The CLAWS2 tagset (Garside et al., 1987) has fewer distinctions, containing 166 tags, but again it is geared towards helping the downstream parser with ambiguous words (e.g. nouns using punctuation as a marker of abbreviation, like 'Mrs.', have a designated tag). The Penn Treebank tagset uses 45 tags[9] and was specifically designed to reduce redundancy at the cost of ambiguity. In this case lexical and syntactic information from the context of the word need to be used to reduce the ambiguity when parsing. Also, in the PTB tagset some of the tags tend to reflect morphological properties instead of syntactic function (e.g. the VBG tag). While this annotation is helpful in English—providing a form of subcategorisation—it could lead to extremely fine-grained tagsets in morphologically rich languages where the majority of the tags encode morphological variations of a single syntactic tag[10]. It is therefore important to keep in mind that that different tagsets were designed to capture different properties of the words and that MATCHLINGUIST will not always provide the most representative results. As Roger Garside puts it:

> ... there can be no claim that the annotation scheme represent 'God's truth'. [...] No one annotation scheme should claim authority as an absolute standard. [...] The purpose for which the annotations are primarily intended may give priority to certain kinds of information...
>
> (Garside et al., 1997, p. 6–7)

One way of avoiding this issue would be to compare the output of the unsupervised

[8]There is also the problem that certain linguistic phenomena might not be well understood or that there are several competing theories that attempt to explain them. This type of problem is systemic to linguistic theory and can be partially addressed by having more than one annotator. However, these problems are beyond the scope of this analysis.

[9]Note that is the number of tags in PTB-2. The original PTB, as mentioned in section 2.3.1, had 48 tags; the tags for '(opening single quote), '(closing single quote) and "(double quotes) were omitted from later versions, and the pound sign tag (#) was replaced with the dash tag (–).

[10]For instance in Spanish there could be as many as 475 tags given the richness of the language's inflectional morphology (Sánchez-León & Nieto-Serrano, 1997, p. 157) and in Turkish the morphosyntactic tagset can contain more than 6,000 tags (Oflazer et al., 2003).

system against a multi-tagged corpus, annotating the same corpus with multiple tagsets and comparing the distance of the unsupervised system's output against all the different gold standards. This, however, is a laborious process and has only been done for a very small fraction of a corpus (see the AMALGAM project of Atwell et al., 1994).

Another solution would be to evaluate the quality of the tagsets themselves, either with respect to their usefulness in parsing as the immediate downstream task (see Déjean 2000 for a proposed method of evaluation) or with respect to any other formal or practical linguistic property we are interested in.

A final approach to solving this problem (and the one I examine further) is to generate a 'surrogate' gold-standard annotation from the raw data and use that to calculate standard evaluation scores. We will now examine one such method.

3.3.5 Non-gold-standard based metrics

3.3.5.1 [s-fscore]: Substitutable F-score

This is a novel evaluation metric proposed by Frank et al. (2009) that requires no gold standard, instead using the concept of substitutability (as described in section 2.2) to evaluate performance. Instead of comparing the system's clusters C to gold-standard clusters T, they are compared to a set of clusters S created from *substitutable frames*, i.e., clusters of words that occur in the same syntactic environment. Ideally, like in Harris's definition, a substitutable frame would be created by sentences differing in only one word (e.g. "I want the blue ball." and "I want the red ball.") and the resulting cluster would contain the words that change (i.e. [blue, red]). However, since it is almost impossible to find these types of sentences in real-world corpora, the authors use frames created by two words appearing in the corpus with exactly one word between (e.g. the ____ball). Once the substitutable clusters have been created, they can be used to calculate the Substitutable Precision (*SP*), Recall (*SR*) and F-score (*SF*) of the system's clustering:

$$SP = \frac{\sum_{s \in S} \sum_{c \in C} |s \cap c|(|s \cap c| - 1)}{\sum_{c \in C} |c|(|c| - 1)} \tag{3.8}$$

$$SR = \frac{\sum_{s \in S} \sum_{c \in C} |s \cap c|(|s \cap c| - 1)}{\sum_{s \in S} |s|(|s| - 1)} \tag{3.9}$$

$$SF = \frac{2 \cdot SP \cdot SR}{SP + SR} \tag{3.10}$$

Note that in order to account for syntactic ambiguity in the frames (as in the following examples), cluster identifiers are appended to each word of the frame.

(3.1) a. I want [to$_1$ eat cake$_2$] today.
b. Put it next [to$_2$ her cake$_1$].

3.3.5.2 [s-vm, s-vmb]

Substitutable V-Measure and V-beta are an addition to the substitutable metrics that incorporate the entropy-based evaluation approach and therefore are not subject to the pairwise nature of *SP* and *SR*. They are calculated like V-Measure and V-beta (equation 3.7) except that instead of the gold-standard tags T we use the substitutable clusters S.

3.3.6 Qualitative Comparison of Intrinsic Evaluation Metrics

Our ultimate goal is to evaluate the performance of various part-of-speech induction systems. However, given the theoretical problems discussed in the previous section, it is imperative to perform a qualitative comparison of the intrinsic evaluation metrics. It is necessary to find a metric that can describe as well as possible the correlation of the induced part-of-speech-tags and the gold-annotated tags. That metric needs to be invariant to the size of the induced tagset and the size of the corpus and also provide intuitive interpretations.

Section 3.3 presented a theoretical overview of the different evaluation methods used in part-of-speech induction along with a small comparison between the strengths and weaknesses of each method. This section presents some empirical results to expand on these claims.

To examine the properties of the various metrics empirically, I performed a series of tests, using a range of different systems and different sizes of the induced tagset. Results were obtained by training and evaluating each system on the full WSJ [**wsj**] portion of the PTB corpus, which (as I have mentioned in section 2.3.1) is one of the most commonly used corpora in the literature.

I also included a 7k sentence version of the WSJ corpus [**wsj-s**] to examine the effects of corpus size. For the WSJ corpora I experimented with two commonly used tagsets: the original PTB 45-tag gold standard and a coarser set of 17 tags previously used by several researchers working on unsupervised part-of-speech induction (Smith & Eisner, 2005a; Goldwater & Griffiths, 2007).

The main system used for these comparison is the Brown clustering algorithm (Brown et al., 1992) as it is one of the most simple and robust part-of-speech induction

	super	all	single
m-1	97.8	14.0	100
crossval	97.6	14.0	0.0
1-to-1	97.9	14.0	0.01
vi	0.3	4.3	15.8
vm	96.0	0.0	35.4
vmb	96.0	0.0	100
s-fscore	7.5	0	0
s-vm	5.8	2.7	0.01
s-vmb	43.6	38.0	95.2

Table 3.2: Supervised (Stanford Tagger) and baseline systems results on the PTB WSJ corpus. The baselines are **all**: every word in the same cluster; **single**: each word to its own cluster.

systems available. A full description of the system is presented in section 3.4.1. I will be referring to this system as **brown**.

Another set of systems which will be used as baselines is comprised of a system that assigns every word in the same cluster [**all**], a system that assigns each word to its own cluster [**single**] and finally a supervised part-of-speech tagging system [**super**]. I will be using the Stanford Tagger[11] (presented in section 2.3.2) trained on the WSJ corpus.

First, we examine the effects of varying $|C|$ on the behaviour of the evaluation measures, while keeping the number of gold-standard tags the same ($|T| = 45$). Figure 3.3 shows the results from **brown** for $|C|$ ranging from 20 to 200. In addition, table 3.2 provides results for the two extremes of $|C| = 1$ (**all**) and $|C|$ equal to the size of the corpus (**single**), as well as the **super** baseline.

These empirical results confirm that certain measures favour solutions with many clusters, while others prefer fewer clusters. As expected, **m-1** correlates positively with $|C|$, rising to almost 85% with $|C| = 200$ and reaching 100% when the number of clusters is maximal (i.e., **single**). Recall that **crossval** was proposed as a possible solution to this problem, and it does solve the extreme case of **single**, yielding 0% accuracy rather than 100%. However, its performance is just like **m-1** for up to 200 clusters, sug-

[11] `http://nlp.stanford.edu/software/tagger.shtml`, accessed 10/05/13.

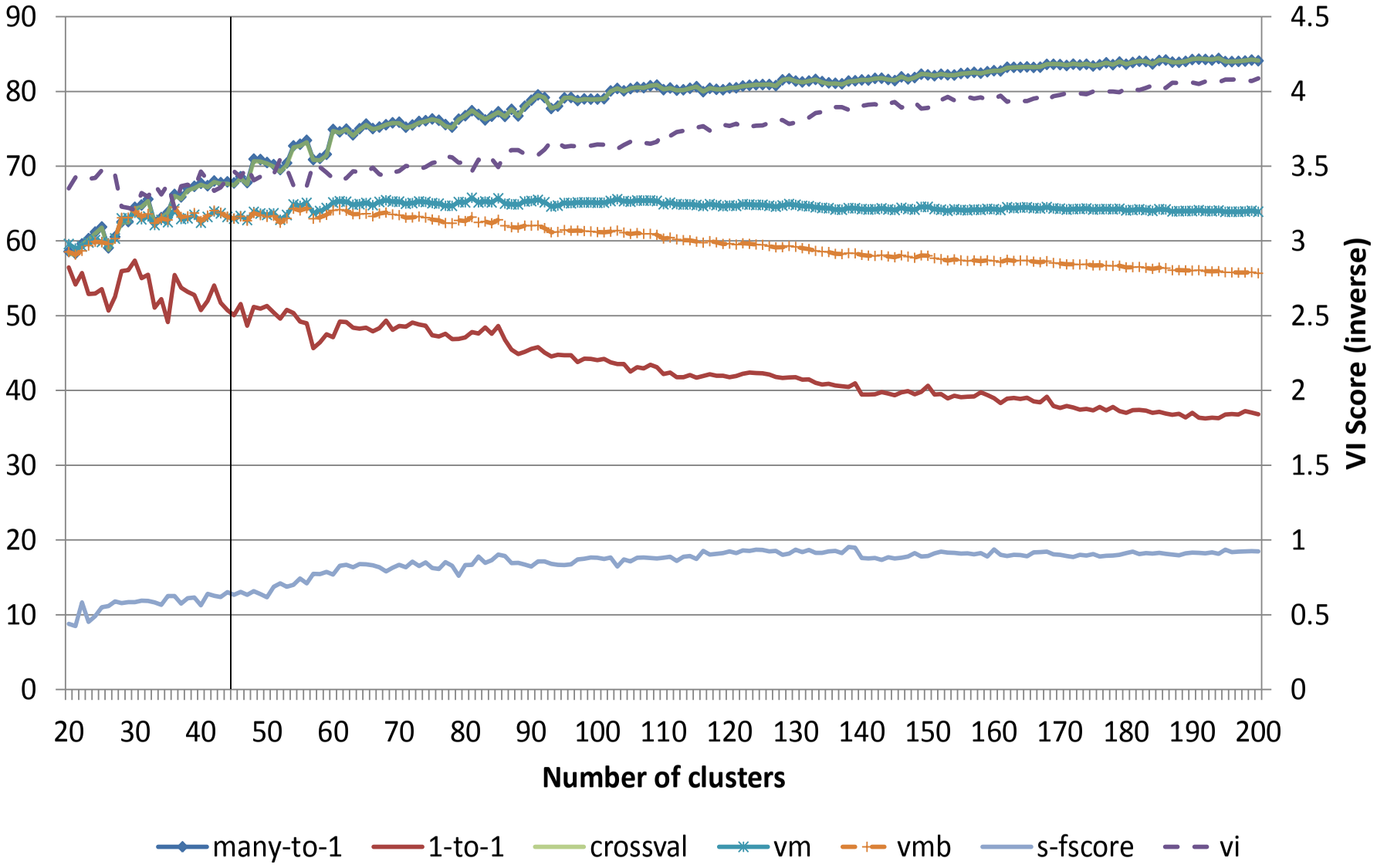


Figure 3.3: Metric evaluation results using the **brown** system (see section 3.4.1.1 below) and the PTB WSJ corpus and varying the number of unsupervised clusters ($|C|$:{20-200}). **vi** results are plotted on the right axis (lower is better). A horizontal line is drawn to represent the gold-standard number of tags ($|T|$:45).

gesting that there is very little difference between the two for any reasonable number of clusters and we should be wary of using either one when $|C|$ may vary.

In contrast to these measures are **1-to-1** and **vi**: for the most part, they yield worse performance (lower **1-to-1**, higher **vi**) as $|C|$ increases. However, in this case the trend is not monotonic: there is an initial improvement in performance before the decrease begins. One might hope that the peak in performance would occur when the number of clusters is approximately equal to the number of gold-standard tags; however, the best performance for both **1-to-1** and **vi** occurs with approximately 25–30 clusters, many fewer than the gold-standard 45. Nevertheless, if the goal is to select the optimal number of clusters to produce using a particular system (rather than to compare different systems producing different numbers of clusters), then these measures may be more appropriate than the others.

Next we consider **vm** and **vmb**. Interestingly, although **vmb** was proposed as a way to correct for the supposed tendency of **vm** to increase with increasing $|C|$, we find that **vm** is actually more stable than **vmb** over different values of $|C|$. Thus, if the goal is to compare systems producing different numbers of clusters (especially important for systems that induce the number of clusters), then **vm** seems more appropriate than any of the above measures, which are more standard in the literature. Note that **vm** was not included in the trials of Headden et al. (2008) and therefore a correlation between its performance as a part-of-speech induction score and the score of a downstream task has not been shown. We will return to this point in chapter 5.

Finally, we analyse the behaviour of the gold-standard-independent measures, **s-fscore**, **s-vm** and **s-vmb**. On the positive side, these measure assign scores of 0% to the two extreme cases of **all** and **single** and are relatively stable across different values of $|C|$ after an initial increase.

Although the actual number of substitutable clusters differs in every system run (since cluster membership information is taken into account) the difference in $|T|$ and $|C|$ is often more than three orders of magnitude (e.g. for **brown** $|T| \approx$77,000 and $|C| = 45$). Since neither **s-fscore** nor **vm** account for such differences in size, they are ineffective in capturing the performance of the system. On the contrary, substitutable V-beta that normalises for $|T|$ proves to be a better indicator of the system's performance. Under **s-vmb** the models behave similarly to the gold-standard metrics, with the exception of **super** (table 3.2). This may seem alarming at first but we should take into consideration that since the new "gold-standard" clusters are not dependent on the PTB tagset (that the supervised tagger is trained on) the gold-standard annotation has

no significant advantage.

Furthermore, **s-fscore** assigns a lower score to the supervised system than to **brown**, indicating that words in the supervised clusters (which are very close to the gold standard) are actually less substitutable than words in the unsupervised clusters. This is probably due to the fact that the gold standard encodes "pure" syntactic classes, while substitutability also depends on semantic characteristics (which tend to be picked up by unsupervised clustering systems as well). Another potential problem with this measure is that it has a very small dynamic range – while scores as high as 100% are theoretically possible, in practice they will never be achieved, and we see that the actual range of scores observed are all under 20%.

It is worth noting that most researchers have not explored solutions with different numbers of clusters while holding the number of gold standard tags fixed, as I described in the experiments above. However, several papers have presented experiments in which both $|T|$ and $|C|$ are varied together, and usually performance is higher for the smaller values of $|T|$ and $|C|$ (for instance see Goldwater & Griffiths, 2007; Naseem et al., 2009; Das & Petrov, 2011). This is intuitive, since there are fewer distinctions to be made, so the choice should be easier.

Figure 3.4 presents the results for all the metrics where the size of the corpus and the granularity of the tagset are varied. Again we fix all other parameters using the **brown** system with $|C|$:45. As expected, the mapping metrics are mostly influenced by the size of the corpus and the number of gold-standard tags $|T|$ with the exception of **1-to-1**, which is more dependent on $|T|$. In the case of **vm** and **vmb** $|T|$ has no effect, while the influence of the size of the corpus is minimal, proving that they are the metrics least affected by any of the parameters varied. The substitutable scores (except **s-vmb**) have no dependencies on the tagset and are affected inversely by the size of the corpus. This is to be expected as the number of s-clusters is proportional to the size of the corpus.

These results raise an important issue. If we take into account the performance of the system in terms of accuracy, we would assume that with more clusters produced, the resulting clusterings should be "cleaner", that is, each cluster will contain almost exclusively members of only one part-of-speech tag. However, both **vm** and **s-vmb** scores seem to suggest that the number of clusters is (or should be) irrelevant to the performance of the system.

One possible conclusion of these experiments is that there is probably no single evaluation measure that is best for all purposes. If a gold standard is available, then

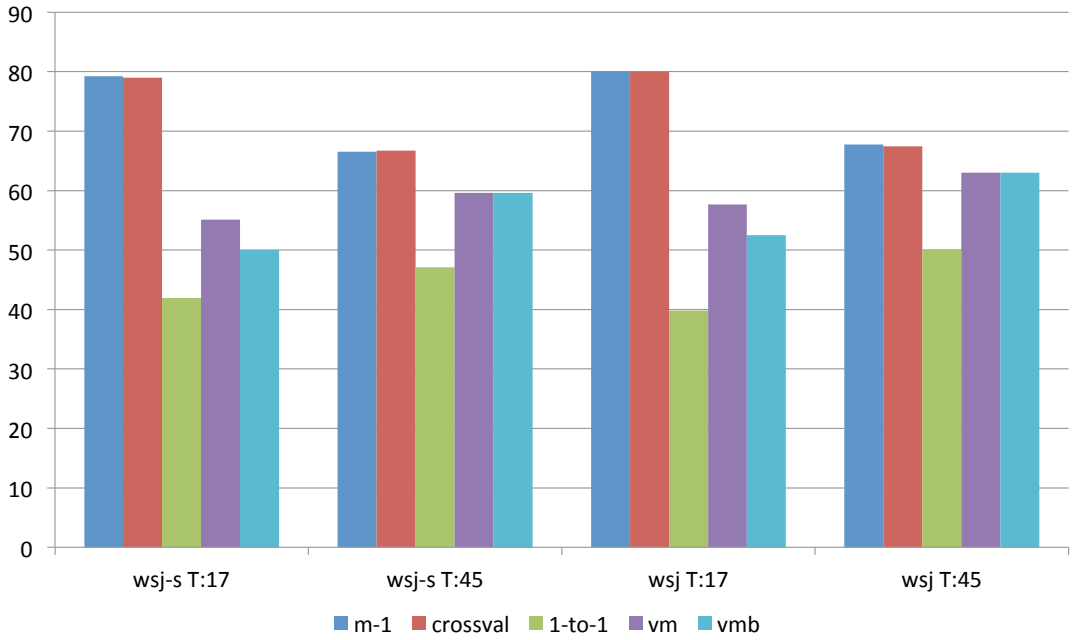


Figure 3.4: Metric evaluation results using the **brown** system (section 3.4.1.1) on corpora of different size ($\{$wsj,wsj-s$\}$) and gold-standard tagsets of different granularity ($|T|$:$\{$17,45$\}$). Number of unsupervised clusters was kept constant at $|C|$:45.

m-1 is the most intuitive measure, but should not be used when $|C|$ is variable, and does not account for differences in the errors made. While **vi** has been popular as an entropy-based alternative to address the latter problem, its scores are not easy to interpret (being on a scale of bits) and it still has the problem of incompatibility across different $|C|$. Overall, **vm** seems to be the best general-purpose measure that combines an entropy-based score that distinguishes between the different types of errors with stability over a wide range of $|C|$. However, despite having a 0–100% scale, like **vi**, it does not provide an intuitive understanding of the underlying clusters, since entropy is measured at the level of the entire clustering.

In conclusion, for all subsequent experiments, I will be using both **m-1** and **vm**, making sure that the number of induced clusters $|C|$ is fixed.

Having provided some data about the behaviour of different evaluation methods, I will move to the presentation and evaluation of unsupervised part-of-speech induction systems. But before I do, and while on the subject of evaluation procedures, I will briefly talk about statistical significance tests, their underlying assumptions and their use for determining performance differences between systems.

3.3.7 Significance testing for part-of-speech induction

Tests for statistical significance of results are one of the cornerstones of scientific discovery. While they cannot be used to *prove* a hypothesis or a theory, they can show that a particular hypothesis is unsatisfactory because the distribution of the data can be explained by a more parsimonious model. The most common form of testing is called *Null Hypothesis Significance Testing* (NHST) and was introduced by Fisher (1925). It replaces the scientific hypothesis with a *statistical* one (A) which we can either accept or reject within some margin of error, based on our observations or measurements (B) of a sample of a population. It uses the *modus tollens* logical argument which is stated like this:

$$\frac{\mathrm{A} \rightarrow \mathrm{B} \qquad -\mathrm{B}}{-\mathrm{A}}$$

We start with the premise that if A is true then B is true (i.e. we believe that our hypothesis will lead to some particular observations); if B is found to be false (i.e. the observations are pointing to the opposite direction), then we can conclude that A is also false[12]. This means that while we cannot accept our original hypothesis even if the data support it, we can reject the opposite hypothesis (the Null Hypothesis or H_0) if the observations do not support it. The amount of 'support' the data provide is called the p-value of the statistical test. If we formulate the null hypothesis as one that contrasts directly with our original theory[13] (now called the Alternative Hypothesis or H_1), by rejecting H_0, it is plausible (within the margin of error, and the fact that other alternative hypotheses might be true) that H_1 is correct. Conversely, if we fail to reject H_0, H_1 becomes less likely as a true explanation. It should be noted however, that failure to reject a (null) hypothesis is not synonymous with accepting that hypothesis (see footnote 12); it only means that the hypothesis is more credible under our current observations. Under this light, as Ramon Henkel puts it, "scientific truths are simply those statements which we consider to have a low probability of being proven incorrect in the future" (Henkel, 1976, p. 35).

Given the probabilistic nature of these tests, there are two types of errors that might occur. The first (type I error) is to incorrectly reject a true H_0. The probability of

[12] The opposite statement—that is, $\frac{\mathrm{A} \rightarrow \mathrm{B} \qquad \mathrm{B}}{\mathrm{A}}$, is a logical fallacy called *affirming the consequent*.

[13] This formulation of the null hypothesis follows the hybrid tradition of statistical hypothesis testing. In the paradigms of *significance testing* of Ronald Fisher and *hypothesis testing* of Jerzy Neyman and Egon Pearson, the definition of the null hypothesis is slightly different.

making this error is denoted as α. This probability is called the *significance level* of the test and is the critical value at which we choose to accept or reject the null hypothesis (if p-value$< \alpha$). The second type of error (type II) is a failure to reject a false H_0 with probability β. The inverse of this probability $(1-\beta)$ is called the *power* of the test which (following from the definition of β) is the probability of confirming the alternative hypothesis when the alternative hypothesis is true.

Significance tests are commonly used in (supervised) NLP in order to show that differences in performance between two different systems are significant (Gillick & Cox, 1989; Och & Ney, 2003; Koehn, 2004). The null hypothesis here is that the performance of the two competing systems (or a system and a baseline) is the same; more explicitly H_0 states that the average performances of the two systems are equal $\mu_1 = \mu_2$. The alternative hypothesis can either be directional ($\mu_1 > \mu_2$ or $\mu_1 < \mu_2$) or non-directional ($\mu_1 \neq \mu_2$). The population which the samples for the test statistics are taken from is usually the full set of test data, and the samples are scores of either individual words or sentences depending on the task.

The idea of the population being only the test-set in a particular language (instead of being the set of all the possible utterances in that language) is problematic, since it narrows the usefulness of the significance test. When a system significantly outperforms another under this assumption, all we can deduce is that the first system is more likely to outperform the second in that test-set and that set alone. When the notion of population is broadened to cover more than the main training/testing corpus (which is closer to what we actually want to measure—i.e. the ability of NLP systems to generalise), the power of the significance tests decreases dramatically. Specifically, Berg-Kirkpatrick et al. (2012) showed that to reasonably predict true performance differences of systems based on observations in a different corpus, a p-value of less than 0.00125 is needed (which is much lower than the commonly used α value of 0.05). This observation brings us to the next problem with significance testing in general which is that the decision as to what is an appropriate significance level α is totally arbitrary (Henkel, 1976, p. 40). As Kanji (2006, p. 3) says 'we usually set α to between 1 and 10 percent, depending on the severity of making such an error' but this value should be (and usually is) domain-specific: one would expect very high significance levels in medicine or engineering, where very precise measurements are needed and the consequences of an error could be dire, and less strict values for psychology or cognitive science where the nature of the experiments is more unpredictable and the consequences of errors are limited.

In NLP, the majority of significance tests use the standard $\alpha = 0.05$ level, inherited from the social sciences, even in the empirical investigation by Berg-Kirkpatrick et al. (2012). Indeed the only part where the researchers look at different significance levels is the cross-corpus testing mentioned earlier, only to discover that the standard 0.05 value on one corpus provides no information about the performance on another. This does not mean that we need to abandon the $\alpha = 0.05$ critical level, but that we need more empirical evidence as to whether it is acceptable for the NLP tasks it is used for. Another problem with the kind of significance tests for NLP is that the β value (and hence the power of the test) cannot be calculated since it requires an exact alternative hypothesis (e.g. $\mu_1 - \mu_2 = 1.38$) which cannot be formulated in advance. This means that we have no way of knowing the probability of type II errors; this is problematic since minimising the probability of type I errors (small α) increases the chance of making a type II error (Henkel, 1976, p. 44).

Despite these problems (and many other theoretical arguments against them e.g. Cohen, 1994; Gigerenzer, 2004; Kline, 2013), statistical tests of significance are still one of the best ways to provide us with some evidence about the validity of our hypotheses, and thus a cornerstone of the scientific method. I will therefore proceed in using them for all subsequent experiments, when their underlying conditions are met. These conditions are different from those of supervised NLP tasks due to the nature of the evaluation. As described in the previous sections, an unsupervised part-of-speech induction system cannot produce gold-standard labels and therefore we cannot use metrics based on the performance of the system on individual words; instead we have to evaluate the entire clustering as a whole, either by first mapping clusters to gold-standard labels (based on the frequency statistics of the entire clustering) or by measuring the entropies of the induced and gold-standard clusterings. For this reason, one option is to treat the entire output of single run of a system as a sample from a population of all possible runs. However, for some of the systems of this and subsequent chapters it is very time consuming to run a single system multiple times on a single language/corpus. Instead, I will treat a system running on a particular language as a sample from the population of all possible system runs in all possible languages. This means that I will not be able to present significance scores for tests ran on a single language/corpus. While this assumption conflates the randomness which is internal to the system (e.g. the random initialisation) with the randomness of drawing a particular language from all possible languages, it is both more practical (only one run per language needed), and more interesting theoretically, since the ultimate goal of my

experiments is to show the ability of unsupervised systems to generalise over different languages.

The test that I will be using is the independent one sample t-test in which the null hypothesis is that the mean of a population is equal to a specific value μ_0. Specifically, I will be testing whether the mean of the differences in the scores of two systems (μ) is significantly different from 0 (so H_0 is $\mu = \mu_0 = 0$ and H_1 is $\mu \neq \mu_0$). This test is equivalent to the *paired* t-test in which two means are checked against each other (as formulated earlier). The basic assumption of the t-test is that the samples are drawn from a normal distribution, which has been met in all the cases where I report significance values. The assumption of normality of distribution of differences in scores was tested using the Shapiro-Wilk test (Shapiro & Wilk, 1965). One related issue is whether the sample of languages was normally distributed across all languages under any quantifiable (continuous) measure of difference. Since most of the languages used in this thesis are Indo-European there is a strong possibility that this is not the case; however, this is an empirical question which lies beyond the scope of the present work.

3.4 Comparison of part-of-speech Induction Systems

The following is an overview of part-of-speech induction systems. It is comprised of a theoretical description (section 3.4.1) and an empirical evaluation (section 3.4.4). The evaluation is a combination of the results presented by Christodoulopoulos et al. (2010), which contains a detailed comparison of multiple systems, over many corpora and $|C|$, $|T|$ configurations; and Christodoulopoulos et al. (2011), where I used published scores on multiple languages, but did not run the experiments myself. I will also present some systems that were not presented in those papers for a more comprehensive exposition of the part-of-speech induction area.

3.4.1 Description of Systems

I describe each system only briefly; for details, see the respective papers, cited below[14]. Each system outputs a set of syntactic clusters C; except where noted, the target number

[14]Implementations were obtained from:
brown: `https://github.com/percyliang/brown-cluster` (Percy Liang),
clark: `www.cs.rhul.ac.uk/home/alexc/pos2.tar.gz` (Alex Clark),
cw: `wortschatz.informatik.uni-leipzig.de/~cbiemann/software/CW.html` (Chris Biemann),
bhmm, **vbhmm**, **pr**, **feat**: by request from the authors of the respective papers.

of clusters $|C|$ must be specified as an input parameter. Since I am interested in out-of-the-box performance, I use the default parameter settings for each system, except for $|C|$, which is varied in some of my experiments.

3.4.1.1 [brown]: Class-based n-grams

Proposed by Brown et al. (1992), this is the oldest and one of the simplest part-of-speech induction systems examined here. It uses a bigram model to assign every instance of a word type to a latent class—also known as a *hard* assignment, as opposed to the assignment of a class to each word token (allowing for part-of-speech ambiguity). The probability of the corpus $w_1 \ldots w_n$ is computed as:

$$P(w_1|c_1)\prod_{i=2}^{n} P(w_i|c_i)P(c_i|c_{i-1}) \tag{3.11}$$

where c_i is the class of w_i. The goal is to optimise the probability of the corpus under this model. The authors use an approximate search procedure: it starts by assigning each word to a distinct cluster and computes the mutual information (see equation 3.2 in section 3.3) between two adjacent clusters using the bigram model above for the cluster probabilities. It then proceeds by merging pairs of clusters that result in the least loss of mutual information.

Apart from the *Markov assumption* (where the probability of the assigned class is only conditioned on the previous class), and the *hard* assignment of clusters, the **brown** system has no explicit learning biases.

3.4.1.2 [clark]: Class-based n-grams with morphology

This system introduced by Clark (2003) is based on a model described by Clark (2000). The system is similar to Brown et al. (1992)—again, a *hard* assignment—the only difference being the use of a slightly different approximate search procedure (an agglomerative clustering algorithm instead of a hierarchical one). In Clark (2003) the original model is augmented with a prior that prefers clusterings where morphologically similar words are clustered together. Each cluster now has a distribution over Σ^*, where Σ is the set of all characters used in the vocabulary. So the probability of the cluster $P(c)$ of equation 3.11 becomes:

$$P(c) = \prod_{i=1}^{n} \prod_{c(w)=i} P_i(w)$$

This morphology component is implemented as a single-order letter HMM. This allows the **clark** system to capture various kinds of morphological phenomena, even though the component is somewhat limited by independence assumptions of the HMM.

3.4.1.3 [cw]: Chinese Whispers graph clustering

Unlike the other systems considered here, this one induces the value of $|C|$ rather than taking it as an input parameter. The system of Biemann (2006) uses a graph clustering algorithm called *Chinese Whispers* that is based on contextual similarity. The algorithm works in two stages. The first clusters the most frequent 10,000 words (*target words*) based on their context statistics, with contexts formed from the most frequent 150–250 words (*feature words*) that appear either to the left or right of a target word. The second stage deals with medium- and low-frequency words and uses pairwise similarity scores calculated by the number of shared neighbours between two words in a four-word context window. The final clustering is a combination of the clusters obtained in the two stages. While the number of target words, feature words, and window size are in principle parameters of the algorithm, they are hard-coded in the implementation used here. As discussed in section 3.1, this makes the system vulnerable to biases introduced during development.

3.4.1.4 [bhmm]: Bayesian HMM with Gibbs sampling

Goldwater & Griffiths (2007) presented a system that is based on a standard HMM for part-of-speech tagging. HMMs have been used extensively in supervised tagging systems (see section 2.3.2) and allow for word-token-based tagging. The main difference from the basic model is the use of *Dirichlet distributions* as priors over the transition and emission probabilities. These priors are used to introduce external knowledge about the distributions of tag-to-tag (state-to-state) transitions and the distributions of the tag-to-word emissions. The shape of the Dirichlet prior distributions is controlled by the transition and emission *hyperparameters* α and β which can be fixed or inferred from the data. In both cases sparse distributions are desirable: only a few tags should follow more than one kind of tag (most of them should only follow a particular tag) and only a few tags should emit most words (i.e. the *open class* words: usually nouns, verbs, adjectives and adverbs). The system uses a *Gibbs sampler* to infer the tags and a *Metropolis-Hastings sampler* to infer the hyperparameters. Since these are the same algorithms used in my part-of-speech induction systems, I will describe them in

detail in section 4.2.2. In the comparison that follows the HMM is a bigram and the hyperparameters are inferred.

3.4.1.5 [vbhmm]: Bayesian HMM with variational Bayes

The system proposed by Johnson (2007) uses the same bigram model as **bhmm**, but instead of a Gibbs sampler, it uses *Variational Bayesian* methods for inference (Attias, 2000). These methods provide an exact analytical solution to an approximation of the posterior distribution, instead of an approximate solution of the exact posterior (which is what Gibbs sampling provides). Like the previous Bayesian systems discussed in this section, **vbhmm** follows the Markov assumption, but unlike **bhmmm** the hyperparameters α and β are both fixed to 0.1, values that appeared to be reasonable based on Johnson's grid search, and which are also used by Graça et al. (2009).

It is interesting to note here that, **bhmm** and **vbhmm** use the same underlying model (an HMM with Bayesian priors) and differ only in their inference method. Their comparison could provide us with a way of separating the effects of the statistical model versus the inference which is not always easy to achieve.

3.4.1.6 [pr]: Sparsity posterior-regularisation HMM

The Bayesian approaches described above encourage sparse state-to-state and state-to-emission distributions only indirectly through the Dirichlet priors. The posterior-regularisation HMM of Graça et al. (2009), while utilising the same bigram HMM, encourages sparsity directly by constraining the *posterior* distributions using the posterior regularisation framework (Ganchev et al., 2009). A parameter σ controls the strengths of the constraints (default = 25). Following Graça et al. (2009) and Johnson (2007), the hyperparameters α and β are again set to 0.1.

3.4.1.7 [feat]: Feature-based HMM

This system by Berg-Kirkpatrick et al. (2010) uses a model that has the structure of a standard HMM, but assumes that the state-state and state-emission distributions are logistic, rather than multinomial. The logistic distributions allow the model to incorporate local features of the sort often used in discriminative models. The default features are morphological, but unlike Clark (2003), this system uses manually-selected morphology features such as character trigrams and capitalisation. This is an obvious

introduction of external knowledge which biases the model; however, as we will see in section 3.4.4 this leads to high performances in most of the languages tested here.

3.4.1.8 [proto]: Learning from Induced Prototypes

One final approach examined here is the induced-prototype learning introduced by Christodoulopoulos et al. (2010). It is based on the prototype-driven learning model of Haghighi & Klein (2006) where a few *prototypes* or canonical examples of each part of speech are introduced as prior knowledge to an otherwise unsupervised system. The system then uses a log-linear model to incorporate various features including morphological information (similar to the ones used by Berg-Kirkpatrick et al., 2010) and the similarity of each token to the prototype words (which is calculated by using SVD on word context matrices and cosine distance between the principal components). To turn the system into a fully unsupervised one (see discussion in section 3.1) we implemented a simple heuristic method for inducing prototypes from the output C of a part-of-speech induction system by selecting a few frequent words in each cluster that are the most similar to other words in the cluster and also the most dissimilar to the words in other clusters. For each cluster $c_i \in C$, we retain as candidate prototypes the words whose frequency in c_i is at least 90% as high as the word with the highest frequency (in c_i). This yields about 20–30 candidates from each cluster. For each of these, we compute its average similarity $\mathcal{S}_{intra}$ to the other candidates in its cluster, and the average dissimilarity $\mathcal{D}_{extra}$ to the candidates in other clusters. Similarity between a pair of words is computed using cosine distance (Haghighi & Klein, 2006) and the dissimilarity is simply one minus the similarity. Finally, we compute the average $\mathcal{M} = 0.5(\mathcal{S}_{intra} + \mathcal{D}_{extra})$, sort the words by their $\mathcal{M}$ scores, and keep as prototypes the top ten words with $\mathcal{M} > 0.25 * \max_{c_i}(\mathcal{M})$.

3.4.2 Systems not included in the review

To provide a better, more up-to-date coverage of the literature, I will now describe some part-of-speech induction systems that were not included in the original review paper (Christodoulopoulos et al., 2010). Unless otherwise stated, the results of these systems—presented in section 3.4.4.1—are drawn from published papers and therefore are not directly comparable to the in-depth results obtained for the systems above.

3.4.2.1 [k-means]: k-means clustering algorithm

k-means (MacQueen et al., 1967) is a well-known clustering algorithm that uses an iterative refinement technique to divide n points (words) to k clusters based on their mean distance in Euclidean space. Each word-type is assigned to one cluster (i.e. a hard assignment) meaning that every instance (token) of that type will be labelled with the same tag. To provide a representation for each word we used the context and morphology feature vectors described in section 4.3.

3.4.2.2 [ihmm]: Infinite HMM

Like **cw**, the Infinite HMM (Van Gael et al., 2009) is another model that induces $|C|$. It uses the same Bayesian approach as the **bhmm** and **vbhmm** by introducing priors to the parameters of a classical trigram HMM, but unlike the previous two approaches, the **ihmm** uses a *non-parametric* framework (Beal et al., 2002) to include the number of hidden states (i.e. the number of parts of speech) as another parameter of the model which can be inferred. To achieve that, the authors used a Dirichlet process (DP, Antoniak, 1974) which is an infinite dimensional version of the Dirichlet distribution, and they experiment with inferring and fixing the hyperparameters. Van Gael et al. also present a preliminary extension of the DP to its generalised form, the Pitman-Yor process (Pitman & Yor, 1997), with limited success.

3.4.2.3 [pyphmm]: Pitman-Yor Process HMM

The system presented by Blunsom & Cohn (2011) is another HMM-based model, which draws from a number of previous contributions to improve on the basic HMM architecture. Like the **ihmm**, this is a non-parametric model which uses Bayesian priors to smooth a standard trigram HMM (like the **bhmm** and **vbhmm**) and a lower-level character HMM to model morphology information (similar to **clark**). Also, similarly to **clark** the **pyphmm** produces a *hard* assignment; it assigns every instance of a word type to the same latent class. The main power of this model comes from the advanced non-parametric priors used for the smoothing. In particular Blunsom & Cohn use a hierarchical Pitman-Yor process (Teh, 2006) to back-off both the transition and emission probabilities of the HMM. The Pitman-Yor process better describes the power-law distribution of natural language categories (Goldwater et al., 2006a), and the hierarchical version allows for a better integration into the HMM model. The inference is performed by a Gibbs sampler and the hyperparameters are sampled using a slice sampler

(Neal, 2003). An extension of this model as well as an examination of different sampling methods is presented by Dubbin & Blunsom (2012).

3.4.2.4 [hcd]: Hierarchy over Class Distributions

Chrupała (2012) describes a simple modular system that induces a hierarchical clustering over word class distributions. His system contains three components: a generative Bayesian word-class induction model (presented by Chrupała, 2011) based on Latent Dirichlet Allocation (LDA, Blei et al., 2003); a hierarchical clustering algorithm that uses the Jensen-Shannon divergence (Lin, 1991) between the class distributions as a distance function and builds a cluster tree of the 1,000 most frequent words; and finally a deterministic system that labels each word type by recording the path down the hierarchical tree until the word is reached in a leaf node (if the word type is one of the 1,000 most frequent words) or by following the path that minimises the Jensen-Shannon distance of that word type and each of the nodes of the tree.

3.4.2.5 [svd]: SVD clustering

Singular Value Decomposition (SVD) is an algebraic method of matrix factorisation and was introduced as a part-of-speech induction tool by Schütze (1995). SVD was used to reduce the dimensionality of left- and right-context matrices of word tokens from thousands of different words to just a few principal components which could then be used to identify the clusters. Lamar et al. (2010) refined this idea and used a two-stage SVD. They applied the original SVD to the context matrices and clustered the principal components into 500 fine-grained clusters (using k-means). They applied the same process to the new clusters, further reducing the number of clusters to the number of gold-standard tags of the corpus.

3.4.3 Datasets

We now move on to evaluate the various systems presented in section 3.4.1. I first present results for the same WSJ corpus used above. However, because most of the systems were initially developed on this corpus, and often evaluated only on it, there is a question of whether their methods and/or hyperparameters are overly specific to the domain or to the English language. This is a particularly pertinent question since a primary argument in favour of unsupervised systems is that they are easier to port to a new language or domain than supervised systems.

To address this question, I will evaluate all the systems without changing any of the parameter settings on the multilingual MULTEXT-East corpus (Erjavec, 2004). Specifically, I will use the 1984 portion of the MULTEXT-East corpus (~7k sentences), which contains parallel translations of Orwell's *1984* novel in eight different languages: Bulgarian, Czech, Estonian, Hungarian, Romanian, Slovene, Serbian and the original English. For this corpus only a coarse 14-tag tagset is available.[15]

As mentioned in section 3.3.6, I will also use **wsj-s**, a 7k sentence version of the WSJ corpus to help differentiate effects of corpus size from those of domain/language. To facilitate direct comparisons of genre while controlling for the size of both the corpus and the tagset, apart from the original 45- and the coarse-grained 17-tagsets, a further collapsed 13-tag set for WSJ was also created.[16]

$|C|$ was set to 45 for all of the experiments reported in this section. Based on the assessment of evaluation measures above, I report **vm** scores as the most reliable measure across different systems and cluster set sizes.

More recently in the published literature, researchers have used the corpora of the CoNNL-X (Buchholz & Marsi, 2006) shared task. This dataset was compiled for the dependency induction task and contains dependency (and part-of-speech) annotated data in 13 languages, 4 of which are freely available (Danish, Dutch, Portuguese and Swedish) and 9 that are used with permission from the creators of the corpora (Arabic[17], Bulgarian[18], Czech[19], German[20], Chinese[21], Japanese[22], Slovene[23], Spanish[24] and Turkish[25]). Following Lee et al. (2010) I used only the training sections for each language. Some of the results of the systems in section 3.4.2 are on this corpus as well the MULTEXT-East. I will also be using this dataset in the experiments described in chapters 4 and 5.

[15] Out of the 14 tags only 11 are shared across all languages. For details see Appendix B in Naseem et al. (2009).

[16] I tried to make the meanings of the tags as similar as possible between the two corpora; I had to create 13 rather than 14 WSJ tags for this reason. The 13-tag set can be found at `http://homepages.inf.ed.ac.uk/s0787820/pos/`.

[17] Part of the Prague Arabic Treebank (Hajič et al., 2003; Smrž & Pajas, 2004)

[18] Part of the BulTreeBank (Simov et al., 2004).

[19] Part of the Prague Dep. Treebank (Böhmová et al., 2001)

[20] Part of the TIGER Treebank (Brants et al., 2002)

[21] Part of the Sinica Treebank (Chen et al., 2003)

[22] Part of the Tübingen Treebank of Spoken Japanese (FKA VERMOBIL - Kawata & Bartels, 2000).

[23] Part of the Slovene Dep. Treebank (Džeroski et al., 2006)

[24] Part of the Cast3LB Treebank (Civit et al., 2006)

[25] Part of the METU-Sabanci Treebank (Oflazer et al., 2003).

system	runtime ($\|C\|$:45)
brown	$\sim$ 10 min.
clark	$\sim$ 40 min.
cw	$\sim$ 10 min.
bhmm	$\sim$ 4 hrs.
vbhmm	$\sim$ 10 hrs.
pr	$\sim$ 10 hrs.*
feat	$\sim$ 40 hrs.*

Table 3.3: Runtimes for the different systems on WSJ [$|C|$:45]. All the systems except **pr** and **feat** were tested on a single 3GHz Xeon processor. *__pr__ and **feat** have multi-threading implementations and were tested on 16 1.8GHz Opteron cores.

3.4.4 Results

I will now present the empirical results of all the systems presented in the sections above. For ease of reference I will use figures instead of numerical scores and when appropriate I will present average results across the different corpora or the different systems. For the full list of results see appendix B.

Figure 3.5 presents results for the seven systems presented in the original review (section 3.4.1), with approximate run-times shown in table 3.3. While these algorithms have not necessarily been optimised for speed, there is a fairly clear distinction between the older type-clustering models (**brown**, **clark**) and the graph-based algorithm (**cw**) on the one hand, and the newer machine-learning approaches (**bhmm**, **vbhmm**, **pr**, **feat**) on the other, with the former being much faster to run. Despite their faster run-times and less sophisticated methods, however, these systems perform surprisingly well in comparison to the latter group. Even the oldest and perhaps simplest method (**brown**) outperforms the two BHMMs and posterior regularisation on all measures. Only the very latest approach (**feat**) rivals **clark**, showing slightly better performance (2.7% improvement on **m-1** and 2.2% on **vm**). The **cw** system returns a total of 568 clusters on this data set, so its **m-1** score is not strictly comparable to the other systems; on **vm** this system achieves middling performance.

Note that the two best-performing systems, **clark** and **feat**, are also the only two to use morphological information. Since the clustering algorithms used by **brown** and **clark** are quite similar, the difference in performance between the two can probably

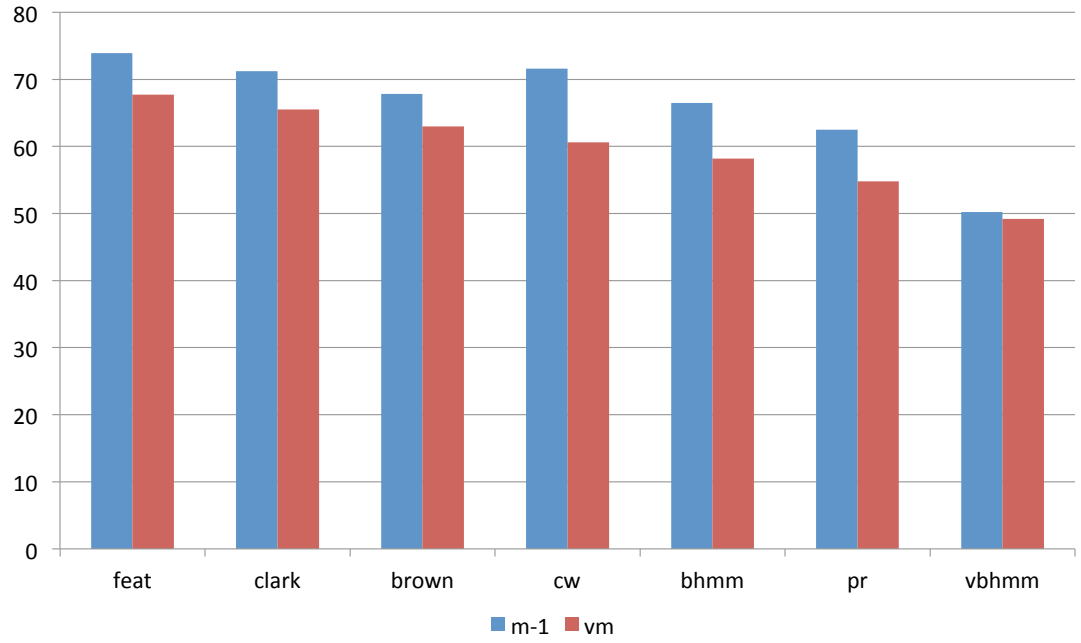


Figure 3.5: V-Measure (**vm**) and many-to-1 (**m-1**) scores for the different systems on the full WSJ corpus [$|C|$:45, $|T|$:45]. Systems are sorted in decreasing performance. For numeric results see table B.1 in appendix B.

be attributed to the extra information provided by the morphology. This suggests that (rather unsurprisingly) incorporating morphological features is generally helpful for part-of-speech induction.

We now examine whether either the relative or absolute performance of the different systems holds up when tested on a variety of different languages. Figure 3.6 illustrates the abilities of the different systems to generalise across different genres of English text. Comparing the results for the MULTEXT-East English corpus and the small WSJ corpus with 13 tags (i.e., controlling as much as possible for corpus size and number of gold standard tags), we see that despite being developed on WSJ, the systems actually perform better on MULTEXT-East. This is encouraging, since it suggests that the methods and hyperparameters of the algorithms are not strongly tied to WSJ.

Another possible explanation is that MULTEXT-East is in some sense an easier corpus than WSJ. Indeed, the distribution of vocabulary items supports this view: the 100 most frequent words account for 48% of the WSJ corpus, compared to 57% of the 1984 novel. It is also worth pointing out that, although previous researchers have reduced the 45-tag WSJ set to 17 tags in order to create an easier task for unsupervised

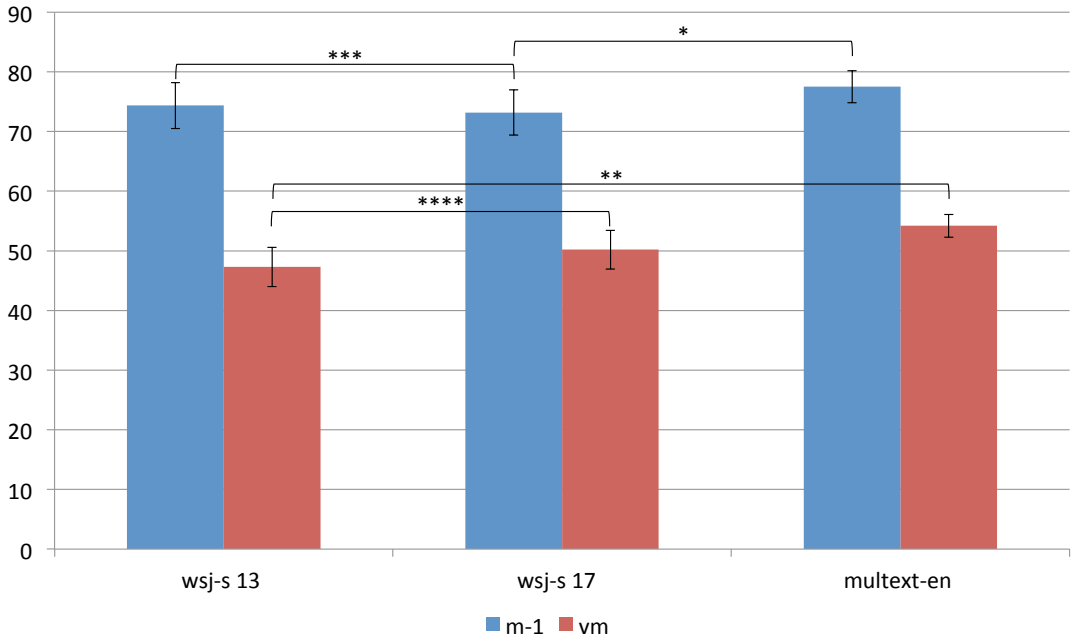


Figure 3.6: Average V-Measure (**vm**) and many-to-1 (**m-1**) scores for the different systems on the 7k version of WSJ (**wsj-s**) and the English MULTEXT-East (**multext-en**) corpora [$|C|$:45, $|T|$:{13,17}]. Significance levels: $* = 0.05$, $** = 0.01$, $*** = 0.001$, $**** = 0.0001$

learning (and to decrease training time), reducing the tag set further to 13 tags actually decreases performance, since some distinctions found by the systems (e.g., between different types of punctuation) are collapsed in the gold standard.

Table 3.4 gives the results of the different systems on the various languages[26]. Not surprisingly, all the algorithms perform best on English, often by a wide margin, suggesting that they are indeed tuned better towards English syntax and/or morphology. One might expect that the two systems with morphological features (**clark** and **feat**) would show less difference between English and some of the other languages (all of which have complex morphology) than the other systems. However, although **clark** and **feat** (along with **brown**) are the best performing systems overall (see figure 3.7), they do not show any particular benefit for the morphologically complex languages.[27]

One difference between the MULTEXT-East results and the WSJ results is that on

[26]Some results are missing because not all of the corpora were successfully processed by all of the systems either due to memory restrictions or character encoding problems.

[27]It can be argued that lemmatisation would have given a significant gain to the performance of the systems in these languages. Although lemmatisation information was included in the corpus, I chose not to use it, maintaining the fully unsupervised nature of this task.

	clark	brown	cw	bhmm	vbhmm	pr	feat
	vm / m-1	vm / m-1	vm / m-1	vm / m-1	vm / m-1	vm / m-1	vm / m-1
Bulgarian	57.3 / 77.4	51.4 / 73.7	42.0 / 59.8	48.1 / 69.1	26.9 / 34.9	35.5 / 56.4	52.5 / 73.4
Czech	51.9 / 72.8	45.0 / 68.4	-	43.1 / 65.1	27.3 / 38.4	28.0 / 49.1	45.4 / 65.2
English	61.3 / 84.3	56.9 / 81.0	53.3 / 80.5	56.9 / 82.0	46.4 / 62.2	47.6 / 72.5	56.9 / 80.0
Estonian	46.4 / 69.8	40.9 / 66.0	38.7 / 66.9	40.0 / 65.3	24.8 / 38.9	27.5 / 52.1	40.6 / 64.8
Hungarian	52.7 / 73.7	45.6 / 67.4	-	44.2 / 68.0	27.7 / 38.7	28.8 / 49.2	53.0 / 74.1
Romanian	56.0 / 75.4	52.4 / 72.3	45.9 / 65.0	49.8 / 69.1	3.2 / 22.8	35.5 / 55.0	-
Slovene	56.3 / 78.3	48.3 / 71.9	39.6 / 62.5	-	27.5 / 42.7	-	46.0 / 70.2
Serbian	51.3 / 72.9	45.2 / 69.3	39.2 / 63.6	-	23.9 / 35.1	-	43.7 / 64.6

Table 3.4: **m-1** and **vm** scores for the different systems on the eight MULTEXT-East corpora [$|C|$:45, $|T|$:variable (13–16)]

MULTEXT-East, **clark** clearly outperforms all the other systems. This is true for both the English and non-English corpora, despite the similar performance of **clark** and **feat** on (English) WSJ. This suggests that **feat** benefits more from the larger corpus size of WSJ. For the other languages **clark** may be benefiting from somewhat more general morphological features; **feat** currently contains suffix features but no prefix features (although these could be added).

Next, I examine the performance of the automatic prototype inducing method on all the systems examined above. Figure 3.8) shows the average performance of the original systems (**base**) and their prototype-based versions (**+proto**) using the prototype extraction method described in section 3.4.1.8, as well as the performance of the Haghighi & Klein (2006) system (**h&k**) wich uses hand-annotated prototypes. We can see that, on average, the performance of the systems is improving, suggesting that the prototype extraction method is indeed effective; however, the average performance is still around 8 points lower in both metrics than the hand-annotated **h&k** system.

Finally, I evaluated the two best-performing **+proto** systems on MULTEXT-East, as shown in figure 3.9. We see that **brown** again yields the best prototypes, and again yields improvements when used as **brown+proto**. Although the improvements are not as large as those on WSJ, they are statistically significant (t = 4.09, p-value = .005). Interestingly, **clark+proto** actually performs significantly worse than **clark** on the multilingual data ($t = -7.66$, p-value = .000), showing that although induced prototypes can in principle improve the performance of a system, not all systems will benefit in all situations. This suggests a need for additional investigation to determine what properties of an existing induction system allow it to produce useful prototypes

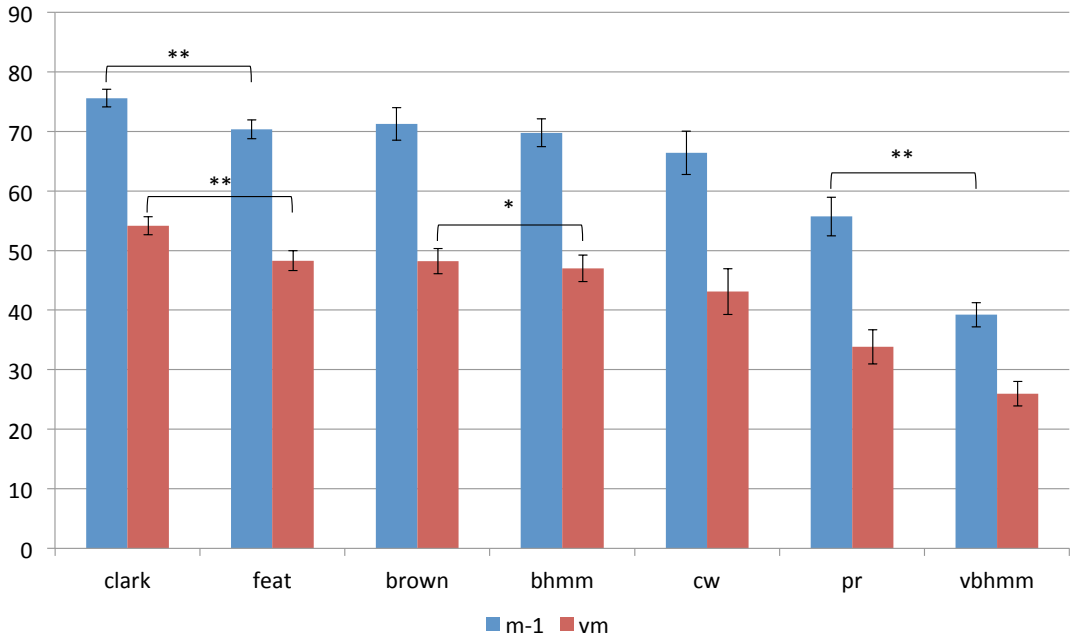


Figure 3.7: Average V-Measure (**vm**) and many-to-1 (**m-1**) scores for the different systems on the eight MULTEXT-East corpora [$|C|$:45, $|T|$:variable (13–16)]. Systems are presented in decreasing **vm** order. Significance levels: $* = 0.05$, $** = 0.01$.

with the current method and/or to develop a specialised system specifically targeted towards inducing useful prototypes. Nevertheless, this is an encouraging first step towards fully-automated prototype-learning systems.

Overall, the experiments on multiple languages support the view that many of the newer part-of-speech induction systems are not as successful as the older methods. Moreover, these experiments underscore the importance of testing unsupervised systems on multiple languages and domains, since both the absolute and relative performance of systems may change on different data sets. Ideally, some of the corpora should be held out as unseen test data if an effective argument is to be made regarding the language- or domain-generality of the system.

3.4.4.1 Systems not included in the review

Figure 3.10 presents the **m-1** results for the systems not included in the original review, as well as the results for **clark**, which had the best overall performance of all the systems in section 3.4.4, for comparison. For the **k-means** and **svd** systems I used

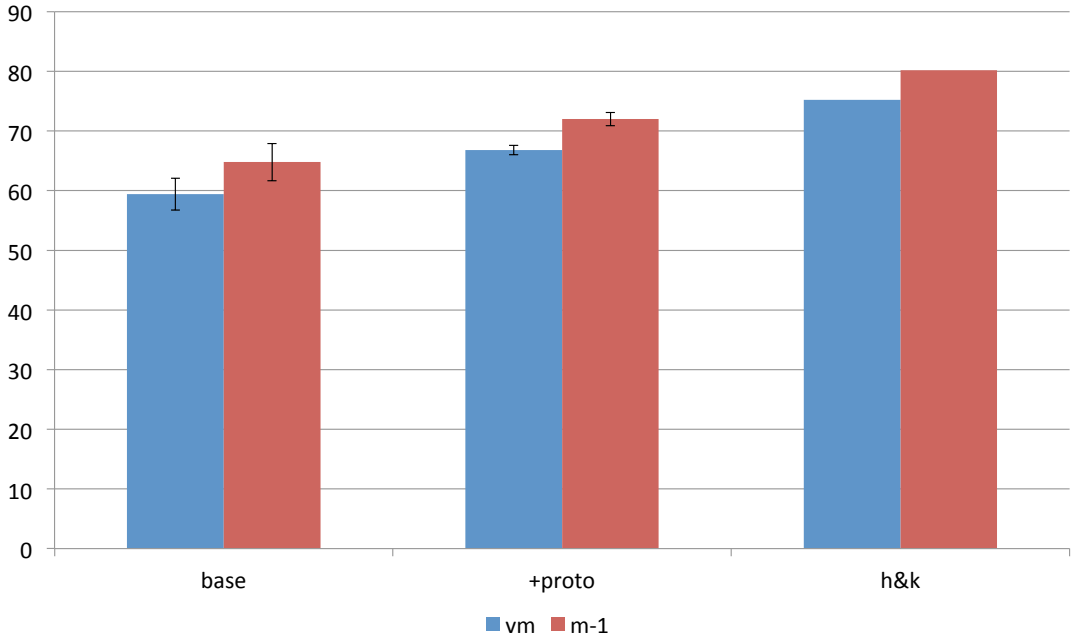


Figure 3.8: Average V-Measure (**vm**) and many-to-1 (**m-1**) scores on WSJ for the original systems examined in section 3.4.4 (**base**) and the prototype-based part-of-speech induction systems (**+proto**), with prototypes extracted from each of the existing systems [$|C|$:45,$|T|$:45]. **h&k** uses hand-annotated prototypes.

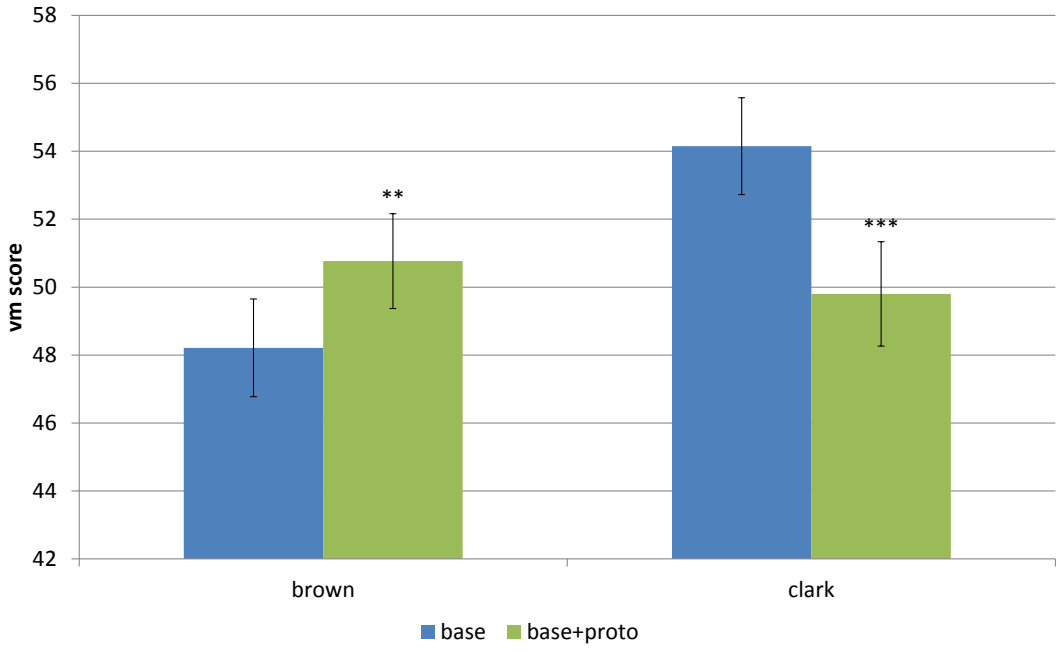


Figure 3.9: Average V-Measure (**vm**) scores for **brown+proto** and **clark+proto** on the MULTEXT-East corpora [$|C|$:45, $|T|$:variable (13–16)]

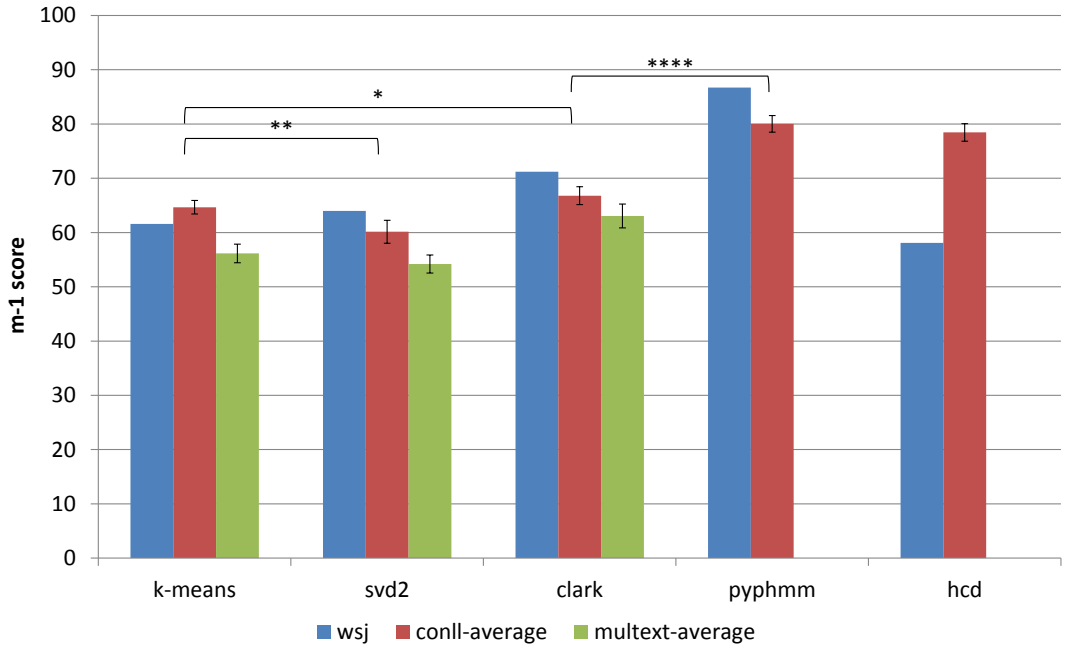


Figure 3.10: Many-to-one (**m-1**) results of systems not included in the review of section 3.4.4. The results for **pyphmm** and **hcd** are taken from the PASCAL challenge Gelling et al. (2012).

publicly available implementations[28] to run the tests using the gold-standard number of clusters.

The results for **pyphmm** and **hcd** were taken from the PASCAL challenge (Gelling et al., 2012). Van Gael et al. (2009) do not report results for languages other than English (WSJ) where their best **vm** score is $\sim$59% (results are in figures, so exact numbers are difficult to obtain).

One of the first observations we can make is that **k-means** is a very strong baseline, beating both the **svd** (on CoNLL, $t = 2.93$, p-value = .022) and **hcd** (on WSJ) and most of the systems tested in my review (see the results in section 3.4.4). On the other hand, the newer systems of Chrupała (2012) (**hcd**) and Blunsom & Cohn (2011) (**pyphmm**) are the clear winners for all the CoNLL languages. The performance of **pyphmm** is markedly better than the next best system (**clark**, $t = 6.26$, p-value = 6.206×10^{-5}) while the difference between **hcd** and **pyphmm** is not significant ($t = -0.72$, p-value = .503). The success of these systems should be attributed both to the complexity of the models and—in the case of the **pyphmm**—to the use of morphological features

[28] I used MATLAB's implementation of k-means. The code for **svd** is available at `http://faculty.biu.ac.il/~marony/code/SVD2/SVD2.m`

(something that is missing from the **ihmm** which also uses a non-parametric Pitman-Yor-based model).

Note that **hcd** produces surprisingly high **m-1** scores (even compared to every other system in the PASCAL challenge, as shown in Gelling et al., 2012); while Chrupała (2012) reportedly uses the gold-standard number of tags, this improvement is not reflected in the **vm** performance (not shown here—see table B.5), suggesting that development attempts were geared towards optimising the **m-1** metric.

3.5 Conclusion

In this chapter I presented an overview of the state of part-of-speech induction, both in terms of available methods and in terms of the inherent difficulties of evaluation.

Concerning the evaluation, looking back to chapter 2, we can see that part-of-speech induction is linked to the nature of parts of speech themselves, and in order to be able to better evaluate our systems, we need to broaden the discussion on what parts of speech are, or what they should be representing[29]. This discussion links back to the systems evaluated in this chapter, with morphologically-aware systems like **clark**, **feat** and **pyhmm** performing much better than their counterparts.

One conclusion that we can take from this chapter is that there are certain properties that lead to better part-of-speech induction models. Despite the underlying statistical models or inference methods, systems that used morphological features (**clark**, **feat** and **pyphmm**) performed significantly better than the rest. Another useful property is that of *hard-clustering*, that is all instances of the same word are assigned to the same cluster. These properties will influence the design of my own induction system described in the next chapter.

Another conclusion that seems to be emerging after this chapter is that we should try to view the problem of part-of-speech induction inside the context of a broader attempt to induce linguistic structure. We should try to think beyond the compartmentalised NLP pipeline and into a more holistic view of computational grammar induction that better reflects the approaches of traditional linguists[30].

However, we must not forget that we are bounded by the restrictions of our com-

[29] Another possibility of course is to move away from intrinsic evaluation of part-of-speech induction systems altogether, using the methods discussed in section 3.2.1.

[30] At least in the case of part-of-speech systems like the ones seen in section 2.2. General linguistic theories of grammar tend to be equally compartmentalised, with the exception of cognitively-motivated linguistic analyses.

putational tools, which is why the more advanced systems presented in section 3.4 (**ihmm**, **pyphmm**, **hcd**), despite their success, were more expensive to train and more difficult to expand[31]. This means that every new method proposed should adhere to these limitations.

It is for these reasons that I have designed a new system for part-of-speech induction; a system that can be extended to incorporate a variety of linguistic features, but that at the same time would have a manageable complexity and would be easy to train. This system, presented in the following chapter, will form the basis for connecting systems across multiple levels of NLP and provide a more holistic approach to linguistic structure induction.

[31] According to my experiments, reports in the respective papers and personal communication with the authors.

CHAPTER 4

The Bayesian Multinomial Mixture Model

Mathematical and other methods of arranging data are not a game but essential parts of the activity of science.

Harris (1954, fn. 5_a)

The results of the previous chapter indicate that although there has been an increasing number of machine-learning-heavy approaches to unsupervised part-of-speech induction, only a few can outperform the much simpler and faster methods such as *k*-means or the systems of Brown et al. (1992) and Clark (2003). Furthermore, the results showed that there are certain features that make the older models (and some of the newer ones) more successful. In this chapter, I will consider which features are more useful and present a system based on the Bayesian machine-learning framework that provides an easy and intuitive way of combining those features. This framework will eventually allow the use of non-local features such as word alignments (see section 4.3.2) and syntactic dependencies (chapter 5). The model I will be using is a generative *Bayesian Multinomial Mixture Model* (BMMM) presented in section 4.2. Most of the work described in this chapter has been previously published in Christodoulopoulos et al. (2011).

4.1 Properties of the BMMM

There are three major properties found in the literature (see previous chapter) that are used by the model presented here: it uses *type-based* instead of *token-based* inference; it is a *clustering* instead of a *sequence* model; and finally it uses additional (non-distributional) features.

The most important property of this model is that it is type-based, meaning that all tokens of a given word type are assigned to the same cluster. This property is not strictly true of linguistic data, but is a good approximation: as Lee et al. (2010) note, assigning each word type to its most frequent part of speech yields an upper bound accuracy of 93% or more for most languages. Since this is much better than the performance of current unsupervised part-of-speech induction systems, constraining the model in this way seems likely to improve performance by reducing the number of parameters in the model and incorporating useful linguistic knowledge. Both of the older systems discussed in section 3.4.1 (Brown et al., 1992 and Clark, 2003), included this constraint and achieved very good performance relative to token-based systems. More recently, Lee et al. (2010) presented a new type-based model, and also reported very good results. Note that implied here is the fact that we are using the systems within a very specific genre. It makes sense that, for example, the word 'bank' will not be used as a verb in the WSJ corpus (nor as geographical formation). However, if the goal is to process text 'in the wild' this assumption no longer holds and we will have to relax the one-tag-per-type constraint. Spitkovsky et al. (2011a) present a principled way of doing this, by using a type-based clustering model as an input to a sequence model (HMM) that allows for ambiguous tagging.

The second property of the model, which distinguishes it from the type-based Bayesian model of Lee et al. (2010), is that the underlying probabilistic model is a clustering model (specifically, a multinomial mixture model), rather than a sequence model (HMM). In this sense, this model is more closely related to several non-probabilistic systems that cluster context vectors or lower-dimensional representations of them (e.g. see Schütze, 1995; Redington et al., 1998; Lamar et al., 2010). As discussed in section 2.3.2, sequence models are by far the most common method of supervised part-of-speech tagging, and have also been widely used in unsupervised part-of-speech tagging approaches both with and without a dictionary[1] (e.g. see Smith & Eisner, 2005a;

[1] In the unsupervised part-of-speech induction literature, dictionary means a list of all the parts of speech seen with a particular word. The use of a dictionary makes these approaches 'resource light'

Haghighi & Klein, 2006; Goldwater & Griffiths, 2007; Johnson, 2007; Ravi & Knight, 2009; Lee et al., 2010). However, systems based on context vectors have also performed well in the unsupervised setting (Schütze, 1995; Lamar et al., 2010; Toutanova & Johnson, 2007) and present a viable alternative to sequence models.

The final property is an advantage that stems from using a clustering model rather than a sequence one. This is that the features used for clustering need not be restricted to context words. Additional kinds of features[2] can easily be incorporated into the model and inference procedure using the same general framework as in the basic model that uses only context word features.

After an overview of the basic model in the following section, I will present a number of extensions. The first extension is a model that decomposes the left and right context of each word token (section 4.3.1). The second extension, described in section 4.3.2, will incorporate the use of *alignment features* gathered from parallel corpora. Previous work suggests that using parallel text can improve performance on various unsupervised NLP tasks such as part-of-speech disambiguation (Naseem et al., 2009), morphological segmentation (Snyder & Barzilay, 2008) and grammar induction (Yarowsky & Ngai, 2001; Cohen et al., 2011). Finally a model with type-level *morphological features*, which serve as cues to syntactic class and seemed to partly explain the success of two best-performing systems analysed in the previous chapter will be described in section 4.3.3.

4.2 The Basic Model

The model I will be using is a generative Bayesian Multinomial Mixture Model. A *mixture model* means that the underlying probability distribution that generated the observed data can be described as a mixture of distributions, each one with a mixing weight. In this case the distributions are assumed to be multinomial[3], that is, each variable can only be one of a set of k possible values (either parts of speech, or word features). Finally, the Bayesian framework is used to infer the underlying structure h (the parts of speech/syntactic categories) from the observed data d. This can be expressed in terms of the *posterior* probability of the structure ($P(h|d)$) which is derived

instead of fully unsupervised (see the discussion in section 3.1).

[2] I will use the word *kind* here to avoid confusion with *type*, which I will reserve for the type-token distinction, a distinction that can apply to features as well as words.

[3] The distributions are, strictly speaking, *categorical* since there is only one observation; however, the term multinomial is much more common in the NLP literature when describing these distributions.

from the *likelihood* of the data ($P(d|h)$) and the *prior* probability of the structure ($P(h)$) using Bayes' rule:

$$\begin{aligned} P(h|d) &= \frac{P(d|h)P(h)}{P(d)} \\ &\propto P(d|h)P(h) \end{aligned}$$

This is a very intuitive framework and has been used to simulate human learning (Goldwater, 2006, p. 9). One of the key features of the Bayesian framework is that we can use the prior distributions to encode external knowledge about the domain. In this case we want to encode the belief that only a few categories can generate most of the words (i.e. open class words are most likely either verbs, nouns or adjectives) and that most words can have only a few features (e.g. exist in very specific contexts). The way to achieve this is by enforcing the mixing weights of both multinomials to be *sparse* by using a Dirichlet distribution with a small concentration parameter.

In the basic model (referred to as **base** henceforth), the observed data are *token-level* features; that is each word token is represented by a single feature (such as which word appears to its left—this feature will be referred to as 'the left context word'). It is straightforward to extend the model to include more than one kinds of features, such as the right context word or the suffix of the current word; I will discuss these extensions in section 4.3. These different kinds of features are assumed to be independent which leads to a deficient model from a generative perspective. Furthermore, modelling explicitly the dependencies between the features could potentially lead to performance gains. However, since we are not interested in the generative capacity of the model but rather in producing the latent structure (i.e. the parts of speech), these assumptions provide a useful and efficient approximation.

The **base** model explains the data by assuming that it has been generated from some set of latent syntactic classes. The ith class is associated with a multinomial parameter vector ϕ_i (the output distribution) that defines the distribution over features generated from that class, and with a mixing weight θ_i that defines the prior probability of that class. The vectors θ and ϕ_i are drawn from symmetric Dirichlet distributions with concentration parameters α and β respectively. These parameters are also called *hyperparameters*. The model is defined so that all observations associated with a single word type are generated from the same mixing component (syntactic class).

As is customary with generative models, the presentation is divided between the *generative story* (section 4.2.1) which explains how the observed data were produced from the underlying latent distributions and the *inference* (section 4.2.2) which de-

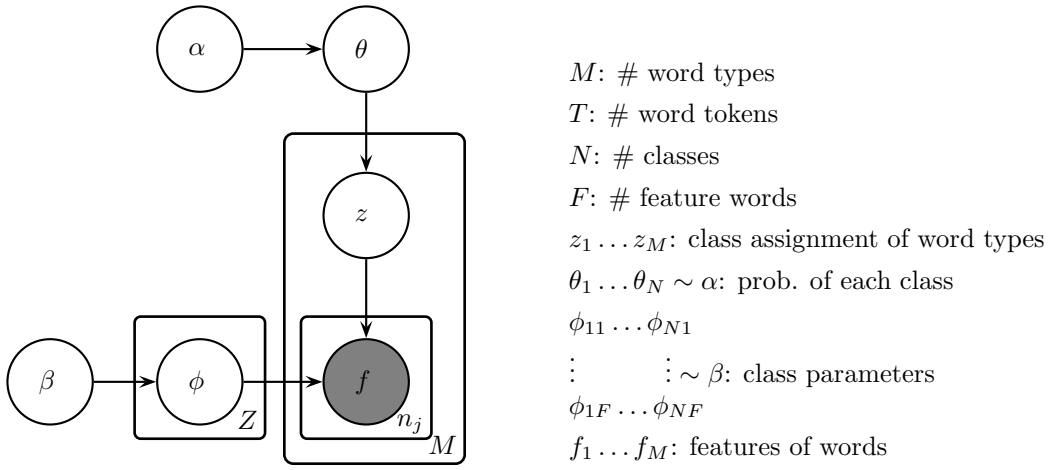


Figure 4.1: Plate diagram of the basic model with a single feature per token. This is the observed variable f as represented by the shaded circle.

scribes the empirical way of getting the latent structure out of the observed data. To better understand these two processes as well as the nature of the model itself, figure 4.1 presents a *plate diagram* of the *Bayes network* that represents the model. A Bayes network is a directed acyclic graph (DAG) where nodes are the random variables (observed and unobserved) and arcs are the dependence assumptions, that is the information required to produce the distribution of each variable (Charniak, 1991). Plate notation used on Bayes networks (Buntine, 1994) is a handy way of representing variables that repeat across the network (e.g. parts of speech should be drawn for every word type).

4.2.1 Generative Story

The observed data can be generated as follows:

1. Generate the prior class probabilities (class mixing weights) from a Dirichlet distribution θ with hyperparameter α. This is formalized as:

$$\theta|\alpha \sim \text{Dirichlet}(\alpha)$$

2. For each word type $j = 1 \dots M$, choose a class assignment z_j from a multinomial distribution (of N elements) with a mixing weight of θ:

$$z_j|\theta \sim \text{Multinomial}(\theta)$$

3. For each class $i = 1 \ldots Z$, generate the class parameters (feature distribution mixing weights) from a Dirichlet distribution ϕ_i with hyperparameter β:

$$\phi_i|\beta \sim \text{Dirichlet}(\beta)$$

4. For each word token $k = 1 \ldots n_j$ of word type j, generate a feature f_{jk} from a multinomial distribution (of F elements) with a mixing weight of ϕ_{z_j}:

$$f_{jk}|\phi_{z_j} \sim \text{Multinomial}(\phi_{z_j})$$

The full joint probability for this generative story can be found by traversing the plate diagram and multiplying the independent components of the DAG over the number of times that each one is generated, as indicated by the different plates. According to conventional notation, the hyperparameters are separated by ';'.

$$P(\mathbf{z}, \mathbf{f}, \theta, \phi; \alpha, \beta) = \prod_{i=1}^{N} P(\phi_\mathbf{i}; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) P(z_j|\theta_j;) \prod_{t=1}^{F} P(f_{jt}|\phi_{z_j}) \tag{4.1}$$

Since F is the number of different possible values a feature can take, ϕ is a $Z \times F$ matrix. Thus, one way to think of the model is as a vector-based clustering system, where word type j is associated with a $1 \times F$ vector of feature counts representing the features of all n_j tokens of j, and these vectors are clustered into similar classes. The difference from other vector-based syntactic class induction systems we saw in section 3.4.1 is in the way the dimensions of the feature vector are reduced and also the method of clustering. Schütze (1995) and Lamar et al. (2010) used dimensionality reduction to reduce the size of the context vectors and then simple k-means clustering to induce the clusters; I will use the F most common words as context features and define a Gibbs sampler that samples from the posterior distribution of the clusters given the observed features. The inference process is described below.

4.2.2 Inference

Like in most applications of generative models in NLP, even though in principle the model can generate the observed data from the underlying distributions, we are more interested in discovering the latent structure itself, which in this case is the parts-of-speech categories. Statistical inference is the tool for this task and can be thought of conceptually as traversing the plate diagram of figure 4.1 in reverse.

Given the nature of the probabilistic distributions used, an exact solution to the inference process is intractable. An alternative is to approximate the posterior distribution by sampling from a distribution that progressively approaches the true posterior. A powerful framework of statistical sampling which has become very popular in NLP research is *Markov chain Monte Carlo* (MCMC). Originally developed for approximating complex physics systems (Metropolis & Ulam, 1949), it has gained wide acceptance in the field of statistics and eventually in machine learning. One of most popular MCMC algorithms for Bayesian inference is *Gibbs sampling* (Geman & Geman, 1984). The main intuition is that using the Markov chain assumption, we can sample one parameter at a time from a conditional distribution of all other parameters which are fixed at that time step. After that we update the parameter and sample the next one. The basic algorithm is described in algorithm 1 (adapted from Bishop, 2006, p. 543). A very good tutorial for deriving a Gibbs sampler including a practical example in topic modelling can be found in Resnik & Hardisty (2010).

Algorithm 1 Basic Gibbs sampling algorithm with k parameters over T iterations.

Initialize $\{z_i : i = 1, \ldots, k\}$

for $t = 1, \ldots, T$ **do**

Sample $z_1^{(t+1)} \sim p(z_1 | z_2^{(t)}, z_3^{(t)}, \ldots, z_k^{(t)})$

Sample $z_2^{(t+1)} \sim p(z_2 | z_1^{(t+1)}, z_3^{(t)}, \ldots, z_k^{(t)})$

$\vdots$

Sample $z_j^{(t+1)} \sim p(z_j | z_1^{(t+1)}, \ldots, z_{j-1}^{(t+1)}, z_{j+1}^{(t)}, \ldots, z_k^{(t)})$

$\vdots$

Sample $z_k^{(t+1)} \sim p(z_k | z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{k-1}^{(t+1)})$

Note that not all model parameters are equally interesting. More specifically, we are interested in the values if the part-of-speech classes but not the class distributions (the mixing weights) or the feature distribution of a particular class. For this reason, rather than using the class and feature selection parameters (θ and ϕ) to predict the assignment in the conditional distribution, these parameters are integrated out and the following posterior distribution is used for the inference using a *collapsed* Gibbs sampler:

$$P(\mathbf{z}|\mathbf{f};\alpha,\beta) \propto P(\mathbf{z};\alpha)P(\mathbf{f}|\mathbf{z};\beta). \tag{4.2}$$

Equation 4.2 decomposes the posterior into the syntactic class prior and the feature data likelihood, which can be calculated separately to yield a multinomial probability vector $\mathbf{z}$ from which we can draw a sample for each word type.

Rather than sampling the joint class assignment $P(\mathbf{z}|\mathbf{f};\alpha,\beta)$ directly, the sampler iterates over each word type j, resampling its class assignment z_j given the current assignments $\mathbf{z}_{-j}$ of all other word types, as illustrated by algorithm 1. This effectively means that we are discarding all our current knowledge about j's class assignment, and (re)evaluating all possible class assignments. The posterior over z_j can be decomposed into the class prior probability and the feature likelihood as:

$$P(z_j|\mathbf{z}_{-j},\vec{f};\alpha,\beta) \propto P(z_j|\mathbf{z}_{-j};\alpha,\beta)P(\vec{f}_j|\mathbf{f}_{-j},\vec{z};\alpha,\beta) \tag{4.3}$$

where $\vec{f}_j$ are the features associated with word type j (one feature for each token of j).

The first factor (the class prior probability) is easy to compute due to the conjugacy between the Dirichlet and multinomial distributions, and is equal to:

$$P(z_j = z|\mathbf{z}_{-j};\alpha) = \frac{n_z + \alpha}{n_\cdot + Z\alpha} \tag{4.4}$$

where n_z is the number of types in class z and $n_\cdot$ is the total number of word types in all classes. For a proof of this derivation see Resnik & Hardisty (2010, p. 15–16). Note that in their derivation Resnik & Hardisty have an extra -1 term in both the numerator and denominator to account for the fact that we have one less word to count, since we are discarding the information about word j. For clarity reasons all counts in this and the following equations are computed with respect to $\mathbf{z}_{-j}$ (e.g., $n_\cdot = M - 1$).

Computing the second (likelihood) factor is slightly more complex due to the dependencies between the different variables in $\vec{f}_j$ that are induced by integrating out the ϕ parameters. Consider first a simple case where word type j occurs exactly twice in the corpus, so $\vec{f}_j$ contains two features (the two left context words of j). The probability of the first feature f_{j1} is:

$$P(f_{j1}|z_j = z,\mathbf{z}_{-j},\mathbf{f}_{-j};\beta) = \frac{m_{j1,z} + \beta}{m_{\cdot,z} + F\beta} \tag{4.5}$$

where $m_{j1,z}$ is the number of times feature f_{j1} has been seen in class z, $m_{\cdot,z}$ is the total number of feature tokens in the class, and F is the number of different possible features.

The probability of the second feature f_{j2} can be calculated similarly, except that it is conditioned on f_{j1} in addition to the other variables, so the counts for previously observed features must include the counts due to f_{j1} as well as those due to $\mathbf{f}_{-j}$. We also need to regularise for the total number of features seen so far. Thus, the probability is:

$$P(f_{j2}|f_{j1},z_j = z,\mathbf{z}_{-j},\mathbf{f}_{-j};\beta) = \frac{m_{j2,z} + \delta(f_{j1},f_{j2}) + \beta}{m_{\cdot,z} + 1 + F\beta} \tag{4.6}$$

where:

$$\delta = \begin{cases} 1 & \text{if } f_{j1} = f_{j2} \\ 0 & \text{otherwise} \end{cases}$$

What this example shows is that, since the feature token emissions are not independent, we should take into account the number of times each feature type has been seen before. Extending this example to the general case, the probability of a sequence of features $\vec{f}_j$ is computed using the chain rule, where the counts used in each factor are incremented as necessary for each additional conditioning feature, yielding the following expression:

$$P(\vec{f}_j|\mathbf{f}_{-j}, z_j = z, \mathbf{z}_{-j}; \beta) = \frac{\prod_{k=1}^{F}\prod_{i=0}^{m_{jk}-1}(m_{jk,z}+i+\beta)}{\prod_{i=0}^{m_{j\cdot}-1}(m_{\cdot,z}+i+F\beta)} = \prod_{k=1}^{F}\frac{\Gamma(m_{jk,z}+\beta)}{\Gamma(m_{\cdot,z+F\beta})} \quad (4.7)$$

where m_{jk} is the number of instances of feature k in word type j, $m_{j\cdot}$ is the total number of features emitted by the type[4] and Γ is the generalised factorial function ($\Gamma(k) = (k-1)!$).

Finally, a simple heuristic is used to avoid the stochastic nature of the sampling, especially since the algorithm is not guaranteed to reach convergence before the maximum number of iterations. At the end of the sampling process, the algorithm will return the part-of-speech sequence with the best posterior probability which is not necessarily the sequence of the last iteration. Algorithm 2 shows the full sampling process.

4.2.2.1 Annealing

In order to improve convergence of the sampler, following Johnson (2007), I used *annealing*. This practically means that instead of sampling from the posterior $P(z_j|\cdot)$, I will be using $P(z_j|\cdot)^{1/\tau}$, where τ is a "temperature" that, while high, allows for wider "jumps" in the search space, and then by cooling down lets the sampler settle closer to the true posterior. The cooling schedule is sigmoid-shaped and drops with every iteration ($iter = 1, \ldots, num_{iter}$) from an initial temperature of 2 down to 1 following:

$$\tau_{iter} = (t_{start} - t_{end}) \times (s_{iter} - s_1)/(s_0 - s_1) + t_{end} \quad (4.8)$$

[4]One could approximate this likelihood term by assuming independence between all $m_{j\cdot}$ feature tokens of word type j. This is the approach taken by Lee et al. (2010).

Algorithm 2 BMMM sampling algoritm

1: INITRANDOM $\{z_i : i = 1,\ldots,M \sim \mathcal{U}[0,N]\}$ OR INITFREQ
2: **FOR** $iter = 1 \rightarrow num_{iter}$ **DO**
3: **FOR** $w = 1 \rightarrow M$ **DO**
4: DISCOUNT(num_Z, z_w)
5: DISCOUNT(num_{F_z}, z_w)
6: **FOR** $c = 1 \rightarrow N$ **DO**
7: $P(c) = P(\mathbf{c};\alpha)P(\mathbf{f}|\mathbf{c};\beta)$
8: $P(c) = P(c)^{1/\tau_{iter}}$ ▷ USING EQ. 4.8
9: z_w =MULTSAMPLE(P)
10: ADDCOUNTS(num_Z, z_w)
11: ADDCOUNT(num_{F_z}, z_w)
12: **IF** $P(\mathbf{z}) > P_{best}$ **THEN**
13: $\mathbf{z}_{best} = \mathbf{z}$
14: $P_{best} = P(\mathbf{z})$
15: α_{new} = SAMPLEHYPERPARAMS$(\alpha_{old}, P(\mathbf{z}), \tau_{iter})$ ▷ USING EQ. 4.9
16: β_{new} = SAMPLEHYPERPARAMS$(\beta_{old}, P(\mathbf{z}), \tau_{iter})$ ▷ USING EQ. 4.9

17: **FUNCTION** INITFREQ
18: SORTBYFREQ$(w_1,\ldots,w_M)$
19: **FOR** $i = 1 \rightarrow N$ **DO** $z_i = i$
20: **FOR** $i = N+1 \rightarrow M$ **DO** $z_i \sim \mathcal{U}[0,N]$
21: **RETURN** z

22: **FUNCTION** MULTSAMPLE(P)
23: **FOR** $i = 1 \rightarrow P_{end}$ **DO**
24: $P_i = P_i + P_{i-1}$
25: $u \sim \mathcal{U}[0, P_{end}]$ ▷ SCALE BECAUSE P IS UNNORMALIZED
26: **FOR** $i = 1 \rightarrow P_{end}$ **DO**
27: **IF** $P_i > u$ **THEN**
28: **RETURN** i

where

$$\begin{aligned} s_{iter} &= 1/(1+e^{\chi(x_i-\psi)}) \\ x_{iter} &= iter/num_{iter} \\ s_0 &= 1/(1+e^{\chi(0-\psi)}) \\ s_1 &= 1/(1e^{\chi(1-\psi)}) \end{aligned}$$

and

$$t_{start} = 2,\ t_{end} = 1,\ \chi = 10,\ \psi = 0.2$$

The values for the parameters of this temperature schedule were taken from the implementation of Johnson et al. (2007) which uses the same inference methods as Johnson (2007).

4.2.2.2 Hyperparameter inference

Instead of fixing the hyperparameters α and β, following Goldwater & Griffiths (2007) I used the *Metropolis-Hastings sampler* to get updated values based on the likelihood of the data with respect to those hyperparameters. The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) is another member of the MCMC framework and more specifically a generalisation of the Gibbs algorithm used for the sampling of the main model. It relies on the use of a proposal (or jumping) distribution $Q(x)$ that is used to generate new samples from and calculate the acceptance ratio. It is common to use the normal (Gaussian) distribution as the proposal.

I will illustrate the use of the algorithm for the inference of α. First a new hyperparameter α_{new} is generated by drawing from the normal distribution centred around the old value with a fixed standard deviation:

$$\alpha_{new} \sim \mathcal{N}(\alpha_{old}, \alpha_{old} \times 0.1)$$

Here I use the *Box-Muller* transform (Box & Muller, 1958) to approximate the draw:

$$\alpha_{new} = \alpha_{old} + \alpha_{old} \times 0.1 \times \sqrt{-2\log X} \times \cos(2\pi Y)$$

where $X, Y \sim \mathcal{U}(0,1]$. Next, the new posterior probability is computed under the new hyperparameter (the features hyperparameter β remains fixed):

$$P_{new} = prior(\alpha_{new}) * likelihood(\beta)$$

Using the new posterior and the proposal distribution I can calculate the *acceptance ratio r*:

$$r = \frac{P_{new}Q(old|new)}{P_{old}Q(new|old)} \tag{4.9}$$

where

$$Q(x,y) = \frac{1}{y \times 0.1\sqrt{2\pi}e^{-\frac{1}{2}(\frac{x-y}{y\times 0.1})^2}}$$

Finally the new hyperparameter is accepted if $r \geq \mathcal{U}(0,1]$. The same method is used for inferring β, the only difference being the calculation of the posterior probability (this time α remains fixed).

Although in principle each kind of feature can have its own β hyperparameter, for simplicity I will use *tied* β parameters for all features[5].

4.2.2.3 Initialisation

According to Clark & Lappin (2010, p. 211) the initialisation method used has little impact on the final results of an unsupervised system. However, it might have an effect on the speed of convergence.

Here, I investigated two different initialisation techniques. The first is the random initialisation (INITRANDOM), commonly used in unsupervised NLP systems. This simply involves drawing a class assignment for each word type from a uniform distribution over the total number of classes. The second (INITFREQ), works by assigning each of the *N* most frequent word types to a separate class and then randomly distributing the rest of the word types to the classes.

As with every stochastic system where the optimisation function is non-convex, doing multiple trials with different initialisations is a crucial step for avoiding local minima. Unless otherwise stated, for most of the results reported below the performance of each system was averaged over three trials.

4.3 Extended Models

The **base** model above can be extended in two different ways: by adding more features at the word token level, or by adding features at the type level. Token-level features can capture detailed things about word behaviour in context, like distributional characteristics, syntactic dependencies or semantic roles; type-level features represent context-

[5]Early development experiments showed no significant difference between using tied and untied βs.

independent properties of words such morphological segmentation, word frequency, open/closed-class membership, etc.

To add more token-level features, I simply assume that each word token generates multiple features, one feature from each of several different kinds. For example, in the model presented in section 4.3.1 the left context word might be one kind of feature and the right context word another. Conditional independence between the generated features given the syntactic class is assumed, so each kind of feature t has its own output parameters $\phi^{(t)}$ and hyperparameters $\beta^{(t)}$.

Due to the independence assumption between the different kinds of features, the basic Gibbs sampler is easy to extend to this case by simply multiplying in extra factors for the additional kinds of features, with the prior (equation 4.4) unchanged. The likelihood becomes:

$$P(\vec{f}_j^{(1)},\ldots,\vec{f}_j^{(T)}|\mathbf{f}_{-j}^{(1\ldots T)},z_j=z,\mathbf{z}_{-j};\beta)=\prod_{t=1}^{T}P(\vec{f}_j^{(t)}|\mathbf{f}_{-j}^{(t)},z_j=z,\mathbf{z}_{-j};\beta) \tag{4.10}$$

where each factor in the product is computed using equation 4.7.

4.3.1 L+R model:

In the **base** model the feature vectors used consisted of the concatenation of the left and right context vectors of each type. However, it is a straightforward extension of the model to introduce a conditional independence on the left and right features and model them separately. This way I can use different hyperparameters on the Dirichlet priors of each vector to achieve different levels of sparsity. Figure 4.2 shows the extended model **l+r**.

Inference is performed in a similar way to the **base** model assuming the independence between the different context vectors. The syntactic class prior remains as in equation 4.4 whereas the feature likelihood for a particular word type j becomes:

$$\begin{aligned}P(\mathbf{l_j},\mathbf{r_j}|z_j;\beta) &= P(\mathbf{l_j}|z_j;\beta)P(\mathbf{r_j}|z_j;\beta)\\ &= \prod_{k=1}^{F_l}\frac{\Gamma(m_{jk,z}+\beta_l)}{\Gamma(m_{\cdot,z}+F_l\beta_l)}\times\prod_{k=1}^{F_r}\frac{\Gamma(m_{jk,z}+\beta_r)}{\Gamma(m_{\cdot,z}+F_r\beta_r)}\end{aligned}$$

Note that this model with multiple context features and the independence assumption between the different kinds of features is deficient. Intuitively, this means that the model can generate data that are inconsistent with any actual corpus, because there is no mechanism to constrain the left context word of token e_i to be the same as the right context word of token e_{i-1}.

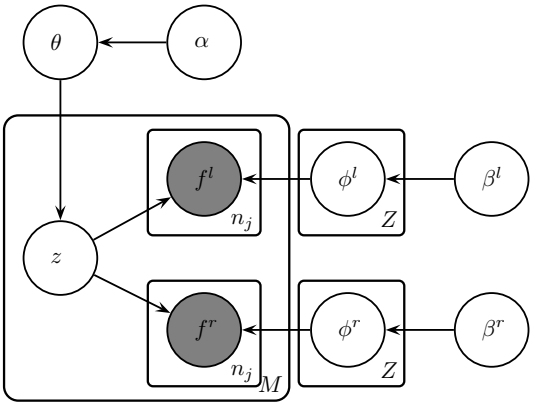


Figure 4.2: Plate diagram of the **l+r** model using two separate vectors for the left and right context.

However, deficient models have proven useful in other unsupervised NLP tasks (Klein & Manning, 2002; Toutanova & Johnson, 2007). In particular, Toutanova & Johnson (2007) demonstrate good performance on unsupervised part-of-speech tagging (using a dictionary) with a Bayesian model similar to the one described here[6]. If we remove the part of their model that relies on the dictionary (the morphological ambiguity classes), their model is equivalent to **base**, without the restriction of one class per type. I use this token-based version of the model as a baseline in my experiments.

4.3.2 Alignments model

Although not directly supported by any linguistic theory of parts of speech, word alignment information can be proven extremely useful for unsupervised part-of-speech induction. As Naseem et al. (2009) explains, the benefit of having multiple languages is that 'the patterns of ambiguity inherent in part-of-speech tag assignments differ across languages'. The means that a part-of-speech assignment (or a set of features that lead to that assignment) in one language can help disambiguate the part of speech in the other.

The **aligns** model of figure 4.3 uses another set of token-level features, this time taking advantage of parallel multilingual corpora. For any 'target' language alignment features can be extracted out of several 'source' languages using the following process.

[6]Although Toutanova & Johnson (2007) refer to their model as "LDA-based", note that Latent Dirichlet Allocation (LDA, Blei et al., 2003) would generate a separate θ vector for each word type (types are analogous to documents, and tokens to words).

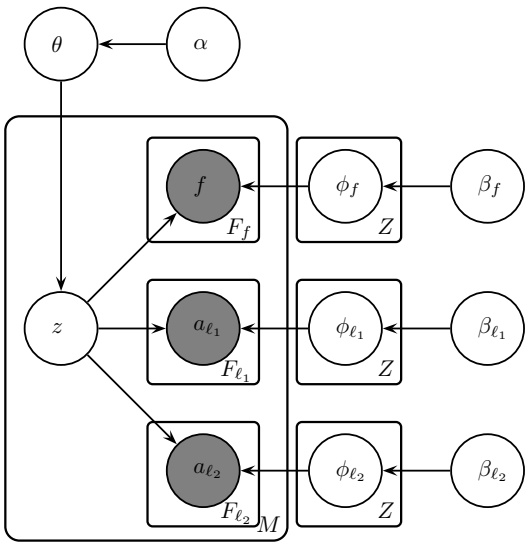


Figure 4.3: Plate diagram of the **aligns** model. The model shown here uses a single context feature vector (left and right concatenated) and two token-level alignment features corresponding to two different languages (ℓ_1 and ℓ_2).

For each of the other languages ℓ, I extract the most frequent word types F^ℓ; using an unsupervised word alignment system I extract a set of bi-directional word alignments[7] between word tokens of ℓ and the original language ℓ_0; finally, for each token t_i in our ℓ_0 I add the left and right context word types of its alignment f_{ij}^ℓ as features if $f_{ij}^\ell \in F^\ell$. Note here that another, more obvious, solution would be to add the aligned word token itself as feature, but preliminary experiments showed that the l+r context yields much better results. Moreover this setup is conceptually closer to the original monolingual model.

Note here that I can concentrate the left and right context features, generating them by a single distribution ϕ, controlled in turn by a single hyperparameter β, or keep them separated as in the **l+r** model. Early development experiments showed that there is no significant difference between the two versions of the model. Therefore, as shown in the plate diagram of figure 4.3, where each language is represented by a single feature vector, the left and right contexts were concatenated.

As mentioned earlier, there is no limit in theory to the number of languages that can be used here; however, there are two limiting factors. The first is that for each of the other languages a new set of $2 \times F^\ell$ features is added which could easily lead to sparsity problems. The second problem is that by adding n languages the alignments

[7]The use of bidirectional vs. unidirectional alignments is discussed in section 4.3.

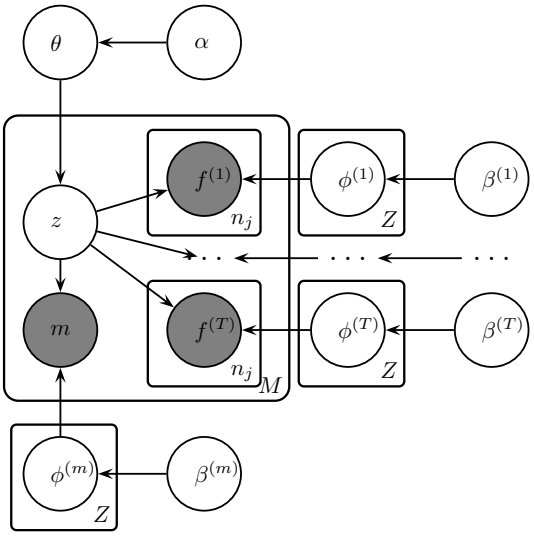


Figure 4.4: Plate diagram of the **morph** model with T kinds of token-level features ($f^{(t)}$ variables) and a single kind of type-level feature (morphology, m).

need to become n-directional, meaning that every word (token) needs to be aligned across all languages. This leads to a rapid decrease of the available aligned words and, again, sparser feature distributions. For both of these reasons the experiments reported in this chapter only involve two languages.

4.3.3 Morphology model

A final extension to this model introduces type-level features, specifically morphology features. The **morph** model, illustrated in figure 4.4, additionally uses T kinds of token-level features (context or alignment). Conditional independence is assumed between the morphology features and other features, so again we can simply multiply another factor into the likelihood during inference. There is only one morphological feature per type, so this factor has the form of equation 4.5. Since frequent words will have many token-level features contributing to the likelihood and only one morphology feature, the morphology features will have a greater effect for infrequent words (as appropriate, since there is less evidence from context and alignments). As with the other kinds of features, I use only a limited number F_m of morphology features, as described in section 4.4.1 below.

The morphological features are extracted from Morfessor (Creutz & Lagus, 2005), an unsupervised morphological segmentation system (described in more detail in sec-

tion 6.2), as a pre-processing step. The type-level restriction means that for each word type we sample only once for its morphological features given the syntactic class. Inference is performed in the same way as before, assuming independence at every level.

4.4 Experiments

4.4.1 Experimental setup

The models were evaluated using an increasing level of complexity, starting with a model that uses only monolingual context features. In order to set the parameters of the models I used a set of development corpora (in English and Bulgarian; see section 4.4.2). Starting with the simplest model (**base**), I use the $F = 100$ most frequent words as features[8], and consider two versions of this model: one with two kinds of features (one left and one right context word) and one with four (two context words on each side). The context features are then concatenated to form a single context vector (with one distribution/hyperparameter controlling it).

Based on preliminary results in English, the **l+r** version of the **base** model is not presented in the results, since it did not yield significantly different results from the concatenated version (the maximum difference in scores was 0.4 in **vm**—lower for the **l+r** model).

For the extended **morph** model, I keep the same setup of the contextual features. To add the morphological features, I ran the unsupervised morphological segmentation system Morfessor (Creutz & Lagus, 2005) to get a segmentation for each word type in the corpus. The suffix of each word type[9] was extracted and used as a feature type. This process yielded on average $F_m = 110$ morphological feature types[10]. Each word type generates at most one of these possible features. If there are overlapping possibilities (e.g. -ingly and -y), the longest possible match was used. Since the goal was to model the morphology at the word type level, I set the value of each feature to 1 if that word type had been observed with that suffix and 0 otherwise.

[8] In practice $F = 101$ since I introduce one extra NULL feature for words that do not have any of the 100 words in their context.

[9] Since Morfessor yields multiple suffixes for each word we concatenated all the suffixes into a single suffix. While Morfessor also produces prefixes, I chose to use only suffixes since they most often carry inflectional information which is relevant to part-of-speech categorisation (see section 6.2 for a discussion).

[10] There was large variance in the number of feature types for each language, ranging from 11 in Chinese to more than 350 in German and Czech.

Another idea that was interesting to explore was to extend the morphology feature space beyond suffixes, by including features like capitalisation and punctuation. Specifically I used the features described by Haghighi & Klein (2006), namely *initial-capital*, *contains-hyphen*, *contains-digit* and added an extra feature *contains-punctuation*. As before, each one is a binary feature generated independently at the word type level.

For the model with alignment features, I followed Naseem et al. (2009) in using only bidirectional alignments: using Giza++ (Och & Ney, 2003), I got the word alignments in both directions between all possible language pairs in our parallel corpora (i.e., alternating the source and target languages within each pair). I then used only those alignments that are found in both directions. As discussed above, I used two kinds of alignment features: the left and right context words of the aligned token in the other language, which were again concatenated into a single feature vector. The feature space is thus set to the $2 \times F = 200$ most frequent words in that language.

Preliminary experiments on the development corpora indicated that better results could be achieved by cooling even further (approximating the maximum-a-posteriori solution rather than a sample from the posterior), so for all experiments reported here, I ran the sampler for a total of 2,000 iterations, with the last 400 of these decreasing the temperature from 1 to 0.66. Finally, all the results shown here are the average scores over three runs of the systems.

4.4.2 Datasets

Although unsupervised systems should in principle be language- and corpus-independent, most part-of-speech induction systems (especially in the early literature) have been developed on English. Whether because English is simply an easier language, or because of bias introduced during development, these systems' performance is considerably worse in other languages, as I have shown in section 3.4.4.

Since the aim is to use this system mostly on non-English corpora, and ones that are significantly smaller than the large English treebank corpora, I developed the BMMM models using one of the languages of the MULTEXT-East corpus, namely Bulgarian. The other languages in the corpus were used during development as a source of word alignments, but otherwise were only used for testing final versions of our models. To have a more intuitive understanding of the results, I also used a smaller version of the WSJ corpus (referred to as **wsj-s**) to approximate the size of the corpora in MULTEXT-East.

System	±1 words vm / m-1	±2 words vm / m-1
base	58.1 / 70.8	55.4 / 67.6
base(tokens)	48.3 / 62.5	37.0 / 54.4
base(INITFREQ)	57.6 / 70.1	56.1 / 68.6
base + morph	58.3 / 74.9	57.4 / 71.9
base + morph(ext)	57.8 / 73.7	57.8 / 70.1
base(INITFREQ) + morph	57.8 / 74.3	57.3 / 69.5
base(INITFREQ) + morph(ext)	58.1 / 74.3	57.2 / 71.3
base + aligns(EN)	58.1 / 72.6	56.7 / 71.1
base + aligns(EN) + morph	**59.0 / 75.4**	57.5 / 69.7

Table 4.1: V-measure (**vm**) and many-to-one (**m-1**) results on the MULTEXT-East Bulgarian corpus for various models using either ±1 or ±2 context words as features. base: context features only; (tokens): token-based model; (INITFREQ): frequency-based initialisation method—other results use INITRAND; (ext): extended morphological features.

For testing, I used the datasets presented in section 3.4.3, namely the remaining seven languages of the MULTEXT-East corpus, as well as the 13 languages of the CONNL-X shared task.

4.4.3 Development results

Tables 4.1 and 4.2 present the results from development runs, which were used to decide which features to incorporate in the final system. Following the discussion in chapter 3, I used V-Measure (**vm**) as the primary evaluation score, because it is less sensitive to the number of classes induced by the model, allowing for the development of my models without using the number of classes as a parameter.

I fixed the number of classes in all systems to 45 during development. However, note that the Bulgarian corpus was tagged using a coarse set of 12 tags which means that the results in Table 4.1 (especially the **m-1** scores) are not comparable to previous results. For results using the number of gold-standard tags refer to table 4.4.

The first conclusion that can be drawn from these results is the large difference between the token- and type-based versions of the system, which confirms that the

System	±1 words vm / m-1	±2 words vm / m-1
base	63.3 / 64.3	62.4 / 63.3
base(tokens)	48.6 / 57.8	49.3 / 38.3
base(INITFREQ)	62.7 / 62.9	62.2 / 62.4
base+morph	66.4 / 66.7	65.1 / 67.2
base+morph(ext)	**67.7** / **72.0**	65.6 / 67.0
base(INITFREQ) + morph	64.8 / 66.9	64.2 / 66.0
base(INITFREQ) + morph(ext)	67.4 / 71.3	65.7 / 67.1

Table 4.2: V-measure and many-to-one results on the **wsj-s** corpus for various models, as described in table 4.1.

.

one-class-per-type restriction is helpful for unsupervised syntactic class induction. We also see that for both languages, the performance of the model using 4 context words (±2 on each side) is worse than the 2 context words model. I therefore used only two context words for all of the additional test languages (below).

We can clearly see that morphological features are helpful in both languages; however the extended features of Haghighi & Klein (2006) seem to help only on the English data. This could be due to the fact that Bulgarian has a much richer morphology and thus the extra features contribute little to the overall performance of the model.

The contribution of the alignment features on the Bulgarian corpus (aligned with English) is less significant than that of morphology but when combined, the two sets of features yield the best performance. This provides evidence in favour of using multiple features.

The frequency-based initialisation method seems less effective than the standard random initialisation. In both corpora it yields worse results than the base system. Moreover, even though the use of this initialisation scheme with the extended morphology features improves the performance relative to the non-extended case, the combined result is still worse than the system with the random initialisation.

Finally, a note on performance: for the two smaller corpora (**wsj-s** and Bulgarian), BMMM takes around 50 minutes on average on a single 2.6GHz AMD Opteron core; however, for the full WSJ corpus (7x bigger) it takes more than 13 hours for a single run (15x longer).

4.4.4 Overall results

Lang.	BASE		ALIGNMENTS		
	base **vm/m-1**	**+morph** **vm/m-1**	**Avg.** **vm/m-1**	**Best** **vm/m-1**	**+morph** **vm/m-1**
Bulgarian	54.4 / 61.5	54.5 / 64.3	53.1 / 60.5	55.2 / 64.5(EN)	**55.7** / **66.0**
Czech	54.2 / 58.9	53.9 / 64.2	52.6 / 58.4	53.8 / 59.7(EN)	**55.4** / **66.4**
English	62.9 / 72.4	63.3 / 73.3	62.5 / 72.0	63.2 / 71.9(HU)	**63.5** / **73.7**
Estonian	52.8 / 63.5	53.3 / **67.4**	52.8 / 63.9	53.5 / 65.0(EN)	**54.3** / 66.9
Hungarian	53.3 / 60.4	54.8 / **68.2**	53.3 / 60.8	53.9 / 61.1(RO)	**55.9** / 67.1
Romanian	53.9 / 62.4	52.3 / 61.1	56.2 / 63.7	**57.5** / **64.6**(ES)	54.5 / 63.4
Slovene	**57.2** / 65.9	56.7 / 67.9	54.7 / 64.1	55.9 / 64.4(HU)	56.7 / **67.9**
Serbian	**49.1** / 56.6	49.0 / **62.0**	47.3 / 55.6	48.9 / 59.4(CZ)	48.3 / 60.8
average	54.7 / 62.7	54.7 / **66.1***	54.1 / 62.4	55.2 / 63.8	**55.5** / **66.5***

Table 4.3: V-measure (**vm**) and many-to-one (**m-1**) results on the languages in the MULTEXT-East corpus using the gold standard number of classes. BASE results use ± 1-word context features alone or with morphology (+MORPH). ALIGNMENTS adds alignment features, reporting the average score across all possible choices of paired language and the scores under the best performing paired language (in parentheses), alone or with morphology features. Significance tests are between **base**–**base+morph**, **base**–**avg. Al.**, **base**–**Best Al.** and **Best Al.**–**Best Al.+morph** and * signifies a p-value of < 0.05.

Table 4.3 presents the average results on the parallel corpora. I tested all possible combinations of two languages to align, and present both the average score over all alignments, and the score under the best choice of aligned language[11]. Also shown are the results of adding morphology features to the basic model (context features only) and to the best alignment model for each language. In accordance with my development results, adding morphology to both the **base** model, and to the best alignment model leads to significant increase in **m-1** scores (t = 3.3, p-value = .013 for **base** and t = 2.94, p-value = .022 for **Best Al.**). There is no change in V-Measure scores for **base+morph** and the increase in **Best Al.+morph** is not statistically significant (t =

[11]The choice of language was based on the same test data, so the 'best-language' results should be viewed as oracle scores.

0.55, p-value = .601).

The alignment results are mixed: on the one hand, choosing the best possible language to align yields improvements, which can be improved further by adding morphological features, resulting in the best scores of all models for most languages; on the other hand, without knowing which language to choose, alignment features do not help on average. Note, however, that three out of the seven languages have English as their best-aligned pair (perhaps due to its better overall scores), which suggests that in the absence of other knowledge, aligning with English may be a good choice[12].

The low average performance of the alignment features is disappointing, but there are many possible variations on this method for extracting these features that we have not yet tested. For example, I used only bidirectional alignments in an effort to improve alignment precision, but these alignments typically cover less than 40% of tokens. It is possible that a higher-recall set of alignments could be more useful.

We turn now to the results on all 21 corpora (18 unique languages), shown in table 4.4 along with corpus statistics, and the best systems from chapter 3, namely the systems of Clark (2003) and Chrupała (2012) which had the best overall performance in the MULTEXT-East and CoNLL corpora respectively. The BMMM system includes morphology features in all cases. Alignment features are not included since these features only yielded improvements for the oracle case where we know which aligned language to choose. We can see that BMMM produces very competitive results. Indeed, as table 4.4 shows, on six out of eight languages of the MULTEXT-East corpus the BMMM outperforms **clark**, and on the CoNLL corpus, BMMM performs better than **hcd** on five out seven languages[13]. On average BMMM scores for the CoNLL languages are better than both **clark** and **hcd** systems, but not significantly (t = 1.76, p-value = .104 for **clark** and t = 1.61, p-value = .158 for **hcd**). Similarly for the MULTEXT-East corpus, the average performance of the BMMM is better than that of **clark** but the difference is marginally not significant (t = 2.17, p-value = .067).

These results show that the BMMM is on par with the state of the art in part-of-speech induction, while at the same time remaining fairly simple and easily expandable. It performs well across multiple languages and is a robust baseline system that can be further extended to handle more linguistic features.

[12]While it is possible to extend the **aligns** model to include alignment features from more than one language (see section 4.3), initial experiments during development suggested that it does not provide further improvement in the performance of the model.

[13]Not all languages of the original CoNLL dataset were used for the Pascal Challenge.

	Lang.	clark	hcd	BMMM	Types	Tags
WSJ	wsj	65.6 / 71.2	53.1 / 58.1	**66.1** / **72.8**	49,190	45
	wsj-s	63.8 / 68.8	-	**67.7** / **72.0**	16,850	45
MULTEXT-East	Bulgarian	**55.6** / **66.5**	-	54.5 / 64.4	16,352	14
	Czech	52.6 / 64.1	-	**53.9** / **64.2**	19,115	14
	English	60.5 / 70.6	-	**63.3** / **73.3**	9,773	13
	Estonian	44.4 / 58.4	-	**53.3** / **64.4**	17,845	13
	Hungarian	48.9 / 61.4	-	**54.8** / **68.2**	20,321	14
	Romanian	40.9 / 49.9	-	**52.3** / **61.1**	15,189	16
	Slovene	54.9 / **69.4**	-	**56.7** / 67.9	17,871	14
	Serbian	**51.0** / **64.1**	-	49.0 / 62.0	18,095	14
	average	51.1 / 63.1	-	**54.7** / **65.7**		
CoNLL06 Shared Task	Arabic	40.6 / 59.8	**51.3** / **83.3**	42.4 / 61.5	12,915	20
	Bulgarian	59.6 / 70.4	-	58.8 / 68.9	32,439	54
	Chinese	31.8 / 56.7	-	42.6 / 69.4	40,562	15
	Czech	47.1 / 65.5	40.2 / **72.3**	**48.4** / 65.7	130,208	12
	Danish	52.7 / 65.3	52.5 / **84.1**	**59.0** / 71.1	18,356	25
	Dutch	52.2 / 67.9	**54.9** / **74.0**	54.7 / 71.1	28,393	13
	German	**63.0** / 73.9	-	61.9 / **74.4**	72,326	54
	Japanese	**78.6** / 77.4	-	77.4 / **78.5**	3,231	80
	Portuguese	57.4 / 69.2	52.5 / **80.4**	**63.9** / 76.8	28,931	22
	Slovene	**53.9** / 63.5	46.6 / **75.5**	49.4 / 56.2	7,128	29
	Spanish	61.6 / **71.9**	-	**63.2** / 71.7	16,458	47
	Swedish	**58.9** / 68.7	47.1 / **79.6**	58.0 / 68.2	20,057	41
	Turkish	36.8 / 58.1	-	**40.2** / **58.7**	17,563	30
	average	53.4 / 66.8	49.3 / **78.5**	**55.4** / 68.6		

Table 4.4: Final results on 21 corpora in 18 languages, with the number of induced classes equal to the number of gold standard tags in all cases. The best systems from chapter 3 (**clark** on the MULTEXT-East corpus and **hcd** on the CoNLL corpus) are included here for reference. BMMM is the **+morph** system (without alignments).

4.5 Conclusion

In this chapter, I have presented a Bayesian model for part-of-speech induction that has two important properties. First, it is type-based, assigning the same class to every token of a word type. I have shown by comparison with a token-based version of the model that this restriction is very helpful. This is as a necessary step in decreasing the complexity of the system described here and related unsupervised systems. At the moment this restriction is relatively harmless since the performance of unsupervised systems is considerably lower than the theoretical upper bound (which, for English, is about 93% according to Lee et al., 2010). However, we should keep in mind that at some point we should have to relax this restriction in order to account for the ambiguity in natural language.

The second property of the BMMM is that it is a clustering model rather than a sequence model. This property makes it easy to incorporate multiple kinds of features (other than distributional) into the model at either the token or the type level. Here, I experimented with token-level context features and alignment features and type-level morphology features, showing that morphology features are helpful in nearly all cases, and alignment features can be helpful if the aligned language is properly chosen.

At the same time, the BMMM (with the morphology features) proves to be a very competitive induction system, achieving performances comparable to, or better than state-of-the-art systems. This provides me with a strong baseline system that I will further extend in the following chapters.

There are two main drawbacks of this model. The first has to do with the independence assumption between the different kinds of features. It is clear that there is an interdependency between the different context features as well as the aligned words and the morphology features. A system that took advantage of these types of dependencies could produce even better results. The other problem comes from the the morphological feature extraction. By using only suffixes as features, the system was biased towards suffixing languages. Some of the languages I tested have a prefixing, or reduplicative morphology, which is hard to capture using just the suffix features extracted from Morfessor. To capture more complex morphological phenomena I would need to use all the information provided by the morphology analyser.

The current system can also be extended in a number of ways. During the alignment feature extraction, I used the frequency of the context words to prune the feature space. However, another possible way of pruning the features would be to use the num-

ber of alignments each word token has instead of the frequency of its context. Another option altogether would be, for each aligned word, to use the previous/next aligned words or even the aligned index jump width as features, instead of the previous/next word of the given word. These are the kinds of features used by the alignment models described in section 6.3.

Furthermore, following Naseem et al. (2009) I chose the bidirectional alignments in an effort to maximise the precision of the output, but also to generate one-to-one alignments at the word token level. However, since the bidirectional alignments cover only a small portion of the total word tokens ($< 40\%$), this process discards a lot of potentially correct unidirectional alignments that could in turn be used as features. The trade-off between alignment precision and feature coverage is an open-ended question that should be further investigated.

One of the most interesting ways to extend the system, not explored in this dissertation, is to replace the standard mixture model with an *infinite mixture model* (Rasmussen, 2000). The main difference would be the use of a non-parametric distribution (instead of the Dirichlet) in the generative story of section 4.2.1. This would enable the model to infer the number of induced parts of speech automatically rather than it being fixed to the number of gold-standard tags. Some possibilities are the *Dirichlet process* and the *Pitman-Yor process*, which are gaining popularity with unsupervised NLP systems, like the **ihmm** and **pyhmm** presented in section 3.4.1. One problem with this approach, which links back to my discussion on unsupervised part-of-speech induction evaluation, is that since we are trying to match the gold-standard tags as closely as possible (MATCHLINGUIST), it makes sense to keep the number of the induced tags the same as the gold-standard tags. The real benefit of a non-parametric model is the ability to discover patterns in the data that might not be captured by the manual evaluation. On the other hand, non-parametric models are sensitive to the amount of data available: they will make really fine-grained distinctions in the presence of big corpora but will be limited to coarser-grained categories with smaller texts. This introduces another dimension to the induction of cross-lingual categories that can be explored further.

As a final remark, this chapter has shown that is possible to add extra features to the BMMM in a easy, intuitive way. Using this system, I can now incorporate multiple levels of NLP analysis with parts of speech at the centre. The next two chapters present a way that this dynamic multilevel induction can be implemented.

CHAPTER 5

The Iterated Learning Framework and Dependency Induction

La nature n'a rien fait d'égal; la loi souveraine est
la subordination & la dépendence[1]

Vauvenargues (1747, p.310)

5.1 Introduction

By now it should be clear that language is a complex, modular phenomenon and that computational models of linguistic structure induction should be equally interconnected, rather than following the traditional pipeline approach.

We now turn to heart of the problem this thesis is examining. Figure 5.1 presents an overview of the interactions between different linguistic levels (morphology, lexicon, syntax, typology) in unsupervised NLP induction systems as documented in the literature. In the diagram, dependency induction corresponds to the linguistic level of syntax, part-of-speech (PoS) induction to the lexicon and (word) alignment induction (roughly) corresponds to typology. We notice that for most of the levels there is no system with bi-directional interaction and certainly little effort has been put into incorporating more than two levels at a time. With the notable exceptions of the two jointly

[1] Nature didn't make anything equal; the sovereign law is subordination and dependence

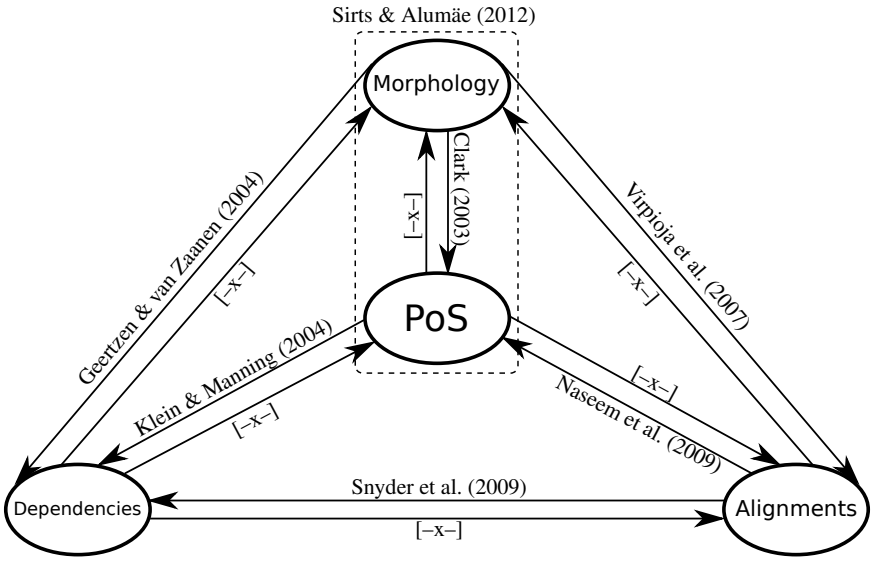


Figure 5.1: The interaction between the various unsupervised NLP areas as documented in the literature. [–x–] denotes the lack of interaction. The dashed line denotes a joint model. Note that this diagram only reports the first published work to show an interaction between any two areas—subsequent studies (even if they were more systematic) are omitted for clarity.

trained models of Sirts & Alumäe (2012) and Lee et al. (2011)[2], in each of these areas of unsupervised NLP, the systems/methods are developed in isolation from the rest of the NLP levels. At best, they use some other level as input but the interaction remains uni-directional. This is the traditional view of the *pipeline* approach where a hierarchy is imposed over these levels:

$$Morphology \rightarrow Parts\text{-}of\text{-}speech \rightarrow Dependencies$$
$$Morphology \rightarrow Parts\text{-}of\text{-}speech \rightarrow Alignments$$

The main reason for the lack of joint models in most cases is the computational complexity of the combined search space. The fact that these interactions are indeed useful (and theoretically motivated) is evident from work in supervised NLP. In the intersection between grammatical structure and syntactic categories, Finkel (2010); Auli & Lopez (2011) and Li et al. (2011) *inter alia* demonstrate the power of joint models over the traditional pipeline approaches.

[2]This model, as will be discussed later, does not actually induce parts of speech; rather it induces a small (5) fixed set of morphosyntactic categories.

5.1.1 Putting the Syntax in Syntactic Categories

In the previous chapter I introduced the Bayesian multinomial mixture model, a flexible method that supports a variety of features, both local and non-local. Until now, in the monolingual setting, the choices of features were limited to the distributional and the morphological characteristics of each word. Given the same raw text, however, there is one more level of information that is inducible and that is syntactic structure.

By allowing the model access to syntactic information, the scope of the definition of syntactic clusters recovered can now cover the majority of definitions of parts-of-speech we saw in section 2.2—given that some of the semantic information can be captured by the distributional properties of the words.

One added advantage syntactic information brings is the possibility of better cross-lingual learning. A long-held belief in linguistics (and recently supported empirically by Moscoso del Prado Martín, in press) has been the principle of *invariance of language complexity*. As Charles Hockett puts it:

> [. . .]impressionistically it would seem that the total grammatical complexity of any language, counting both morphology and syntax, is about the same as that of any other. This is not surprising, since all languages have about equally complex jobs to do, and what is not done morphologically has to be done syntactically. (Hockett, 1958, p. 180-1)

The invariance of language complexity means that the information carried by parts of speech might be in different levels in different languages. For instance, noun case is marked using inflectional morphology in languages like Greek and Japanese, whereas in English it is sometimes marked by a separate part of speech (preposition) and word order (syntax). Another example is passive voice, which in Malay/Indonesian is marked with morphological segments instead of the syntactic constructions used in English.

This means that when identifying cross-lingual parts of speech we should allow for different linguistic levels to align—not just a word-by-word alignment; that is, a sub-word unit might have the same part of speech as a full word, or a syntactic structure containing multiple parts of speech in one language might correspond to a single one in another. A multilingual part-of-speech induction system that is aware of the complexity of either the morphology, or the syntactic structures should be able to push the alignments towards the appropriate levels. Using syntactic information as part of the part-of-speech induction system brings us one step closer to this goal.

5.1.2 The Proposed Approach

To access the syntactic information I will be using *dependency induction* methods, the unsupervised equivalent of dependency parsing. Usually the dependency relations are drawn between parts of speech instead of words, to avoid data sparsity, especially in the unsupervised case. There are some lexicalised dependency induction systems (e.g. Headden et al., 2009) but even they rely on part of speech tags for back-off.

The reliance of dependency induction systems (as well as most syntactic parsers) on parts of speech lends itself naturally to the creation of a joint part-of-speech and dependency induction system. As we saw in section 2.3.1, parts of speech are defined as a way to help the parser. This means that if a part-of-speech induction system was relying (partly) on syntactic information then it is natural to link the two processes to a feedback loop, thereby creating a proxy to a fully joint learning model. This is the *iterated learning* approach presented in section 5.5, where I use the BMMM system of the previous chapter with the *dependency model with valence* (DMV) of Klein & Manning (2004). Sections 5.6 and 5.7 extend the iterated learning experiments by introducing a state-of-the-art dependency parser, instead of the DMV, and by using sentences of full-length (longer sentences are a known limitation of dependency models). Most of the work covered in these sections has been presented in Christodoulopoulos et al. (2012).

Creating a fully joint part-of-speech and dependency induction system is much more challenging, since the combined search space of the distributional, morphological and syntactical features together with all the possible hierarchical dependency trees is prohibitively large. A preliminary approach to this problem will be presented in section 5.8.

5.2 Background

5.2.1 Dependency Grammars

Dependencies are one way of describing hierarchical linguistic structure that can be encoded in 1-to-1 relations between words. Another type of linguistic structure description is constituency relations of phrase-structure grammars. They encode 1-to-N relations between parts of the sentences, either individual words or phrases. In other words, dependency grammars tell us how words relate to each other, while con-

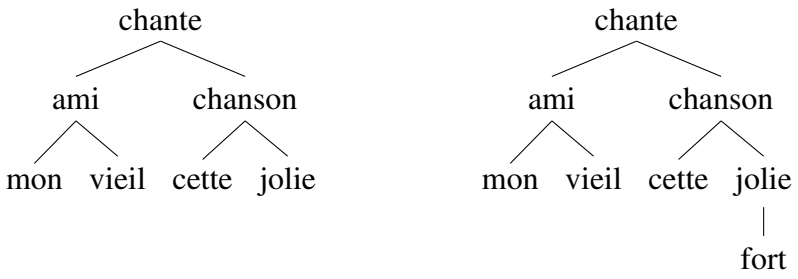


Figure 5.2: Dependency tree structures of the sentences *mon vieil ami chante cette jolie chanson* (my old friend sings this beautiful song), and the same sentence with the addition of the modifier *fort* (very) [source: Tesnière (1959, p. 14–15)]

stituency grammars tell us how words combine into phrases[3].

Dependency grammars were introduced by Tesnière (1959). In his *Éléments de syntaxe structurale* Tesnière draws trees (*stemmas*) between words that represent hierarchical dependency relations (*connexions*) between a governing word (*régissant*) and a dependent (*subodonné*). An example tree is shown in figure 5.2. The notion of linguistic *headedness* (or headship), also used in constituency grammars, plays a crucial role in dependency grammars. Here, head information is not merely complementary to the structure of the sentence (i.e. node labels); rather it gives rise to the structure of the sentence itself. Nevertheless, there does not seem to be an agreement on a formal (consistent) definition of headedness in either constituency or dependency theories.

Although Tesnière does not offer a formal account of headedness, he implies that there are two broad planes on which headedness can be defined (Tesnière, 1959, p. 43): the semantic and the morphosyntactic. This broad distinction, although not followed by Tesnière himself, has led to the development of three distinct notions of dependencies (semantic, syntactic, morphological; see Polguáere & Melčuk (2009, 8–57) for an overview). This chapter mainly deals with syntactic dependencies, but it would be interesting to compare the structures produced by unsupervised dependency systems and semantic dependency corpora such as FrameNet (Ruppenhofer et al., 2006) or the corpus of Mingqin et al. (2003).

In his thorough discussion on headedness, Zwicky (1985) provides five different formal definitions of *head* which he then compresses down to a basic two (semantic and morphosyntactic). In the semantic definition Zwicky starts with the notion of *head*

[3]This makes constituency grammars not suitable to use a word-type-level feature. However, given the equivalency of dependency and constituency trees (Robinson, 1970) one could use any grammatical formalism to produce the type-level features used here.

as follows: 'in a combination X+Y, X is the semantic head, if, speaking very crudely, X+Y describes a kind of the thing described by X'. He then extends this definition to include every X that can be a semantic argument to a functor Y (under a Montagovian-style semantics). According to Zwicky the exact complement to this definition is the notion of *governor* found is some theories of syntax (e.g. in Chomsky, 1965) which is the word that licences all the morphological and syntactic aspects of the modifier.

According to the morphosyntactic definition the head of phrase is 'the bearer of the morphosyntactic marks of syntactic relations between the construct and other syntactic units' (Zwicky, 1985), where the marks can either be overt inflectional properties (in heavily inflectional languages) or abstract 'potential' inflections in languages such as English.

Zwicky also recognises a distributional definition of headedness which is of particular interest to this thesis. Under this operational definition a *head* is a word that belongs roughly to the same distribution as the phrase as a whole. This is equivalent (under the distributional definition of syntactic categories) with the notion of *head* in the X-bar theory (see section 2.2) where the head of a phrase is the word that has the same category as the phrase (or, in other words, when the whole phrase can be replaced by a single word of the same category).

Finally Zwicky acknowledges that in dependency grammars the notion of *head* has no clear definition but that there is a consensus among linguists that for *endocentric constructions*[4] the head is the distributional equivalent, whereas for the *exocentric* ones, the head is the governor.

To review the various definitions of headedness, table 5.1 presents six example constructions and their respective heads according to each definition.

Unfortunately as with parts of speech (see section 2.3.1), the corpus-based dependency parsing approaches suffer from a lack of rigour in their annotation of headedness. For instance one of the most commonly used dependency annotation schemes is the CoNLL constituency-to-dependency conversion scheme of Johansson & Nugues (2007). They follow a series of previous annotation schemes leading back to Magerman (1994) via Yamada & Matsumoto (2003) and Collins (1999). As Magerman (1994, p. 66) admits: 'the lexical representative from a constituent loosely (*very* loosely) corresponds to the linguistic notion of a head word' and that 'the set of deterministic

[4] An endocentric construction is a phrase where one of its parts carries the bulk of the semantic content, making the whole phrase fulfil the same linguistic function as that part (e.g. a *Det + N* phrase). Inversely, an exocentric phrase is one where there the semantic load is spread over more than one part (e.g. an *NP + VP* phrase)

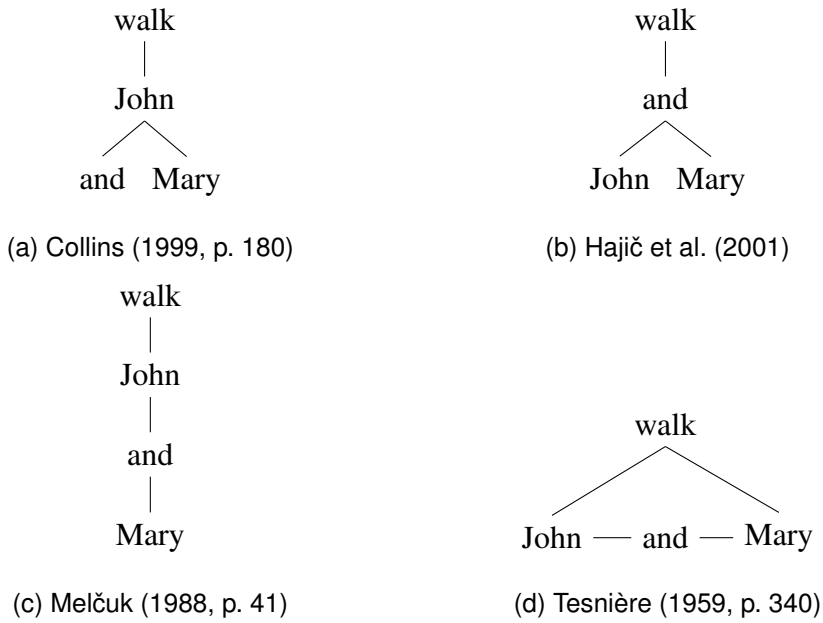
Construction	Example	Semantic	Morphosyntactic	Governors	Distributional	Dependency Grammars
Det + N	the beer	N	N	Det	N	N
V + NP	drink the beer	NP	V	V	V	V
Aux + VP	must drink the beer	VP	Aux	Aux	VP	VP
P + NP	about the beer	NP	P	P	(Adv.)	P
NP + VP	we drink the beer	NP	VP	VP	(S)	VP
Comp. + S	that we drink the beer	S	S	Comp.	S	Comp.

Table 5.1: Comparison of different definitions of grammatical headedness based on the analysis of Zwicky (1985).

rules which select the representative word from each constituent, [...] was developed in the better part of an hour, in keeping with the philosophy of avoiding excessive dependence on rule-based methods'.

This lack of rigour (as well as the theoretical disagreement) often leads to difference of opinion in corpus annotation, which in turn leads to problems in unsupervised dependency systems and their evaluation. In addition to the constructions examined in table 5.1, one especially problematic case is coordination. For a construction such as *John and Mary walk* there are at least three ways of representing the coordination, all of which have been proposed at some point in the literature (figure 5.3a to 5.3c). Interestingly, Tesnière (1959, p. 340) had proposed a much more intuitive *horizontal* dependency relation between the two constituents of the coordination (figure 5.3d) but this was impractical to use in computational representations of dependencies since they need to be acyclic graphs (DAGs) for easier processing.

In the cases when dependency treebanks were created from scratch, such as the Prague Dependency Treebank (Böhmová et al., 2001), the annotators tried to provide a clear definition of headedness. In Hajičová (2002) we read: 'the dependent node is the member of the pair that is syntactically omissible, if not in a lexically specified pair of words (as is the case with endocentric syntagms) then at a level of word classes'. This is reminiscent of the semantic (and distributional) definition given above; however, this is only used in a deep dependency structure called *tectogrammatical tree struc-*



(a) Collins (1999, p. 180)

(b) Hajič et al. (2001)

(c) Melčuk (1988, p. 41)

(d) Tesnière (1959, p. 340)

Figure 5.3: A comparison of dependency representation of coordination structures in literature.

tures (TGTSs) in which only content words appear and function words are clustered together with those content words[5]. The creators also introduce a surface-level layer of annotation called *analytic tree structures* (ATSs) mostly for the convenience of the annotators which they admit 'does not immediately correspond to a level substantiated by linguistic theory'.

5.2.2 Unsupervised Dependency Induction

The task of unsupervised dependency induction deals with the automatic discovery of a hierarchical syntactic structure of a sentence comprising of 1-to-1 dependency relations between words (word x is parent of word y) given just raw text. The main difference from supervised dependency parsing (apart from the lack of annotated training examples) is that in the supervised case the relations (arcs) are labelled with their syntactic role. The notation used in the dependency parsing literature is a flattened variation of the original dependency trees. An example is shown in figure 5.4.

Unsupervised dependency induction has been a very difficult problem to crack, mostly due to the complexity of the search space involved. To get a sense of the size of the space we can calculate the total number of possible binary trees a sentence with

[5]This is similar to the idea of *nœds* in (Tesnière, 1959, p. 55) where he regards function words as morphemes of categorical (content) words and clusters them together.

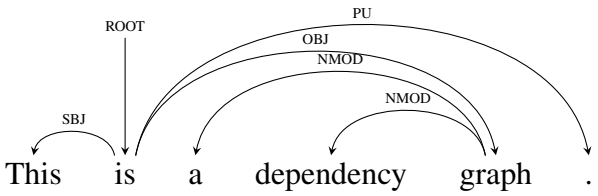


Figure 5.4: Dependency graph with nodes corresponding to words and arcs representing dependency relations. In the unsupervised case these dependencies are unlabelled.

n words has, using the Catalan number C_{n-1}:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

While a sentence with 10 words can have 4,862 full binary trees (not counting trees with only one child), a sentence with 20 words has more than 1.7 billion possible trees.

As recently as 2004 no system had outperformed the right-branching baseline on English, i.e. attaching every word to the word immediately to its right. This simple baseline is extremely powerful since English is a predominately right-branching language with the subject of each sentence being put first and followed by a series of modifiers or subordinate clauses.

The right-branching baseline was beaten when Klein & Manning (2004) introduced the *dependency model with valence* (DMV), which used the concept of valence to compute the probability of a given node attaching to a parent node. All the non-terminal nodes are lexicalised in the sense that their labels are derived from the leaf nodes; however, for sparsity reasons the model uses part-of-speech tags as terminal symbols.

The DMV model is equivalent to a Context-Free Grammar (CFG) with only a few rules for head nodes to generate children (for a description of the grammar see Klein, 2005, p. 106).

5.2.2.1 Description of the DMV model

What follows is a brief description of the DMV model and dependency notation I will be using later.

The DMV model uses the concept of *valence* to compute the probability of a given node attaching to a parent node. It generates dependency trees based on three decisions (represented by three probability distributions) for a given head node h: whether to attach children in the left or right direction, $P_{ORDER}(dir|h), dir \in \{l, r\}$; whether or

not to stop attaching more children in the specific direction given the adjacency of the child in that direction, $P_{STOP}(h, dir, adj), adj \in \{\text{T,F}\}$[6]; and finally whether to attach a specific child node α, $P_{ATTACH}(\alpha|h, dir)$.

The likelihood of the dependency tree $P(h)$ rooted at h can be derived by recursively calculating the following probability for all the dependents of h in any direction:

$$\begin{aligned} P(D(h)) = & \prod_{dir} \prod_{\alpha} (1 - P_{STOP}(h, dir, adj)) \\ & P_{ATTACH}(\alpha|h, dir) P(D(a)) \\ & P_{STOP}(h, dir, adj) \end{aligned} \tag{5.1}$$

This can be seen intuitively as the probability of the node h generating all its child nodes in one direction until it stops and then generating all its child nodes in the other direction. The likelihood of an entire sentence is the sum of the likelihoods of all the possible derivations headed by ROOT ($\diamondsuit$).

The model is trained using the expectation-maximization (EM) algorithm (Dempster et al., 1977) and the parameters are estimated using inside-outside training (Baker, 1979).

A very important aspect of the DMV model is initialisation. Klein & Manning (2004) use what is called the *harmonic initialiser*. They initialise the stopping (P_{STOP}) and direction (P_{ORDER}) probabilities to a fixed value, and set the attachment probability to $1/(1 + distance(h, a))$; intuitively this means that the further away a child is from its head the less likely it is to be attached to that head—a preference for short-distance dependencies.

Other approaches that have been suggested include a uniform initialiser (Spitkovsky et al., 2010b), and a modified version of the harmonic initialiser by Spitkovsky et al. (2011c) where P_{ATTACH} is initially set to $1 + (1/log_2(1 + distance(h, a)))$.

5.2.2.2 Overview of Related Dependency Induction Systems

In recent years the DMV model has been the basis for most state-of-the-art dependency induction systems, including *inter alia* the extended valence grammar (EVG) of Headden et al. (2009), an extension of the DMV grammar which included a Variational

[6] In Klein (2005, appendix A.2) and in most implementations the adjacency is between the head and the current child, but in Klein & Manning (2004) and in Spitkovsky et al.'s descriptions of the model *adj* is true iff the **first** child is adjacent to node h.

Bayes re-estimation for the model parameters and also a fully lexicalised version of the extended model (L-EVG).

Cohen & Smith (2009) used a Bayesian learning approach with logistic normal priors on the model parameters to encode prior beliefs about which parameters should co-vary, effectively tying several prior parameters, and showed that their model can be used to induce a bilingual grammar with no parallel text.

Another extension of the DMV grammar was made by Blunsom & Cohn (2010) where the basic CFG-style grammar of the DMV was replaced by a *Tree Substitution Grammar* and the sparsity was enforced by a hierarchical non-parametric Pitman-Yor Process. This is one of the best performing parsers to date (Gelling et al., 2012) and will be used as a replacement of the basic DMV model. A more detailed description is given in section 5.6.

Gillenwater et al. (2010) also looked at the idea of sparsity, but instead of enforcing it through Bayesian priors, they used the Posterior Regularization framework of Ganchev et al. (2009), presented briefly in section 3.4.1.

Recently, Valentin Spitkovsky and colleagues, have presented numerous extensions of the DMV model: Spitkovsky et al. (2010a) trained multiple versions of the model in increasingly larger sentences. Spitkovsky et al. (2010b) and Spitkovsky et al. (2011b) presented alternative versions of the EM re-estimation scheme. Spitkovsky et al. (2011c) explored the use of punctuation in dependency parsing, something that had not been addressed explicitly in the literature[7].

Finally, Spitkovsky et al. (2011a) showed that it is possible to achieve competitive (and even state-of-the-art) performance with automatically induced part-of-speech tags. Prior to that study, both Klein & Manning (2004) and Headden et al. (2008) had shown that the performance of the DMV model drops significantly with induced part-of-speech tags. To achieve their results Spitkovsky et al. (2011a) used a larger number of induced classes ($\sim$200 rather than 45 which is the number of part-of-speech tags in the WSJ) and a larger training corpus (100M words rather than the 1M words of the full WSJ and the 36K of WSJ-10).

5.2.3 Influence of Parts of Speech on Dependency Induction

Most unsupervised dependency systems following the DMV model rely on gold-standard part-of-speech tags, either directly, using the part-of-speech tags instead of words, or

[7] As discussed later, I will also be using punctuation in my system since it is an important contextual feature for part-of-speech induction.

indirectly, in the back-off mechanism of fully lexicalised models (Blunsom & Cohn, 2010; Headden et al., 2009).

Klein & Manning (2004) showed that when using induced part-of-speech tags their system achieved worse results, but they did not investigate whether better induced part-of-speech tags provided better dependency parsing results. Instead, this idea was explored a few years later by Headden et al. (2008), where dependency parsing was used as an extrinsic evaluation task for part-of-speech induction systems. That study showed that different part-of-speech induction systems lead to quite different performance on dependency induction; however, the scores of the best performing systems were still significantly worse than those with gold-standard part-of-speech tags. Importantly, Headden et al. (2008) also showed that the various unsupervised part-of-speech induction metrics correlate only weakly with the systems' performance on dependency induction, emphasising the importance of extrinsic evaluation in any unsupervised task.

More recently, Spitkovsky et al. (2011a) demonstrated the first positive results of using unsupervised part-of-speech tags for dependency induction. The authors first showed that by using a much larger training corpus and a large number of induced tags, a relatively simple part-of-speech inducing system (Clark, 2003) was able to produce dependency parsing results competitive with those produced with gold-standard tags. Furthermore, by using an HMM on top of a hierarchical clustering system (Brown et al., 1992) to relax the one-cluster-per-type constraint[8] they achieved state of the art dependency parsing results.

5.2.4 Influence of Dependencies on Part-of-speech Induction

It has been shown in supervised systems that using a hierarchical syntactic structure model can produce very competitive sequence models; in other words, that a parser can be a good tagger (Li et al., 2011; Auli & Lopez, 2011; Cohen et al., 2011). This seems sensible, as the parser uses a rich set of hierarchical features that enable it to look at a more global environment than a part-of-speech tagger, which in most cases relies solely on local contextual features.

However. this interaction has not been shown for the unsupervised setting, either in the case where just the part-of-speech tagger is unsupervised and gold-standard dependencies are provided, or where both the part-of-speech inducer and the dependency

[8] All tokens of the same word must have the same tag (also known as *hard clustering*–see section 3.4.1); although this constraint does not allow for syntactic ambiguity it has been proven useful for unsupervised part-of-speech induction systems.

parser are fully unsupervised. This work is the first to show that using dependencies for unsupervised part-of-speech induction is indeed useful, in both scenarios.

5.2.5 Evaluation

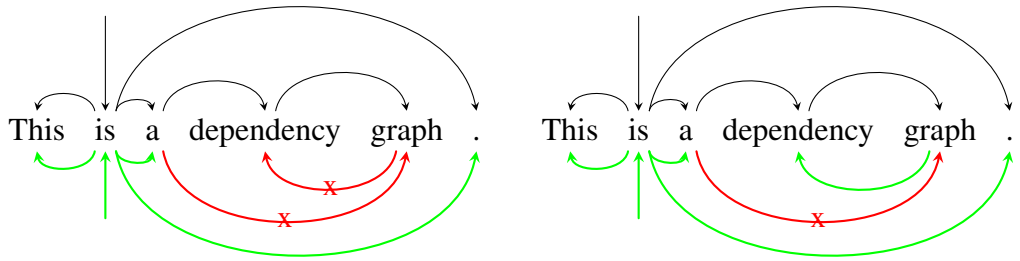
In my discussion of evaluation methods for part-of-speech induction in section 3.2 I mentioned that any intrinsic evaluation of unsupervised systems will suffer from the problem of MATCHLINGUIST, namely that induction systems will not be able to discover the same structure as predicted by the annotators and that different annotators (and different corpora) will have different annotation schemes or adhere to different linguistic theories.

The same is true for the evaluation of dependency induction (and dependency parsing): there are a number of dependency treebanks for many languages and most of them use different annotation schemes. In some case the differences are subtle but they lead to significantly different results.

The standard evaluation metrics used for unsupervised dependency induction are *directed* and *undirected* accuracy. They refer to the number of correct dependencies predicted by the unsupervised system, either taking into account the direction of the dependency or not. Figure 5.5 demonstrates the calculation of both scores. As shown in the discussion concerning headedness in the previous section, there is no consensus about the head (and therefore the direction) assignment in many dependency derivations. In these cases the undirected accuracy score [**undir**] may seem like a better choice, since it does not penalise for these annotation differences. However, undirected accuracy will not discriminate between cases of genuine headedness ambiguity (as in the examples of table 5.1) and truly false head assignments.

Another problem with unsupervised accuracy (and much more so for the directed score) is that it does not allow *edge-flips*. Edge-flipping (Schwartz et al., 2011) refers to the phenomenon where the local dependencies between at most three words are switched. As shown in figure 5.6a, edge-flipping might occur by assigning the preposition as the head of the infinitive ('to'→'go') or determiners as heads of nouns in a prepositional phrase. These are common errors among unsupervised dependency parsers and, as we saw in section 5.2.1, they represent valid interpretations under some definitions of headedness (where the *governors* of the dependencies are heads).

However, as we saw in figure 5.5b, even the less restrictive undirected accuracy metric penalises them, since the *edge-flip* operation not only changes the direction of



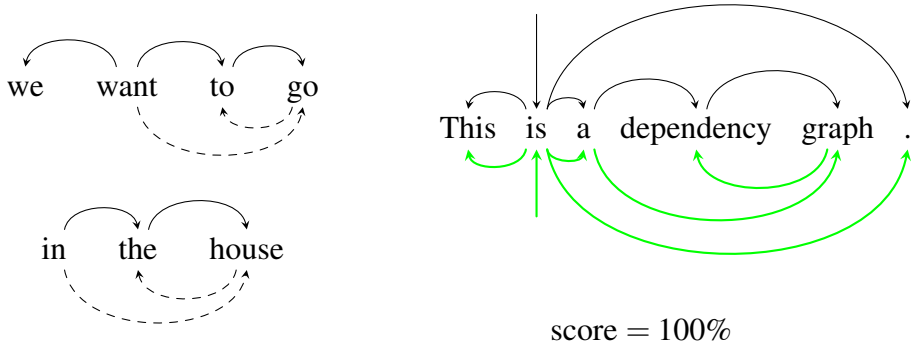
score = 66.6%

(a) Directed accuracy evaluation

score = 83.3%

(b) Undirected accuracy evaluation

Figure 5.5: Comparison of unsupervised dependency evaluation metrics. The gold-standard dependencies are shown above and the induced below the sentence.



(a)

score = 100%

(b)

Figure 5.6: Examples of *edge-flipping* (marked by dashed edges) and the proposed *Neutral Edge Detection* evaluation metric. **ned** allows the edge-flip between 'a' and 'graph', since 'graph' is the grandparent of 'a'.

the dependency of the adjacent words but introduces a new dependency between non-adjacent words (that is, between the word and its gold grandparent). To account for this, Schwartz et al. (2011) proposed an even less restrictive version of the unsupervised accuracy metric, *Neutral Edge Detection* [**ned**], which marks a dependency as correct if the induced parent for a word is either its gold parent, its gold child, or its gold grandparent. Figure 5.6b illustrates the **ned** evaluation. For all the dependency induction experiments below I will be reporting **undir** and **ned** scores.

5.3 Experiments

The following sections describe the experiments on the interaction between parts of speech and dependencies. I begin in section 5.4 with a proof-of-concept experiment using gold-standard dependencies as features for part-of-speech induction. Following that, sections 5.5–5.7, describe the experiments of the *iterated learning* framework: induced parts of speech are used for unsupervised dependency induction and the induced dependencies are then used as features for a new 'generation' of part-of-speech induction, creating a feedback loop between the two induction components. Finally, I introduce a fully joint part-of-speech and dependency induction model in section 5.8.

5.3.1 Experimental setup

5.3.1.1 Systems

For all the experiments in this chapter I will be using the BMMM system described in the previous chapter with 500 sampling iterations, the random initialiser and the following features: the 100 most frequent context words (± 1 context window), the suffixes extracted from the Morfessor system Creutz & Lagus (2005) and the extended morphology features of Haghighi & Klein (2006).

For the iterated learning experiments of section 5.5, as well as the joint model experiments of section 5.8, I will be using the original version of the DMV model[9] as it is the most straightforward to implement and extend. The parser for the iterated learning experiments of section 5.6 is the *TSG*-DMV parser of Blunsom & Cohn (2010)[10]

5.3.1.2 Corpora

As with most unsupervised methods in NLP, the aim here is to demonstrate the effectiveness of my system in languages other than English. I therefore used the CoNLL-X 2006 shared task dataset (Buchholz & Marsi, 2006) containing 13 languages for my tests. I also used the WSJ portion of the Penn Treebank (Marcus et al., 1993) mainly as a development corpus, but also to provide an easier comparison with the part-of-speech and dependency induction literature. The main problem with the Penn Treebank is that it contains only phrase-structure constituency information. To acquire

[9] I used the baseline DMV implementation of Gillenwater et al. (2010) with Klein & Manning's (2004) *harmonic initialiser* (explained in section 5.2.2.1).

[10] Implementation acquired on request from the authors.

the gold-standard dependencies from the WSJ corpus I used the LTH Constituent-to-Dependency Conversion Tool[11] (Johansson & Nugues, 2007) on the NP-bracket-corrected version of the Penn Treebank 3 corpus created by Vadas (2009).

A subset of eight languages from the CoNLL dataset, as well as the WSJ, have been used in the recent PASCAL challenge on grammar induction (Gelling et al., 2012), which makes the comparison with other multilingual part-of-speech and dependency induction systems easier, so for the experiments of sections 5.5–5.7 I used only those nine languages.

For my initial unsupervised dependency induction experiments I removed sentences that contained more than 10 words. This was done mostly for historical reasons: following Klein & Manning (2004), subsequent approaches to dependency induction have, until very recently, used only up-to-10-word sentence corpora. Even in the recent challenge on grammar induction (Gelling et al., 2012) the organisers provided an evaluation with the 10-word cutoff threshold (however, they also had evaluations on 20-word and full-length sentences). Another reason for keeping the 10-word sentence length restriction was efficiency during the development and testing of my hypotheses since the DMV systems I used are extremely slow on longer sentences.

Unlike most work in dependency parsing however, I did not remove punctuation marks. They are important contextual markers for part-of-speech-tag prediction (and even dependency induction, as shown by Spitkovsky et al., 2011c) and most unsupervised part-of-speech induction systems evaluate on corpora with full punctuation. Thus the results presented in sections 5.5, 5.6 and 5.8 are directly comparable to other part-of-speech induction systems[12], but not to other dependency induction systems.

5.4 BMMM with Gold-Standard Dependencies

I begin my investigation into the combination of dependencies and part-of-speech tags with a proof-of-concept scenario. To see whether using dependencies as features for part-of-speech induction is helpful, I will be using gold-standard dependencies as token-level features of the BMMM system similarly to the use of morphological and alignment features (see section 4.3).

Because the different kinds of features are assumed to be independent in the BMMM,

[11] Available at: `http://nlp.cs.lth.se/software/treebank_converter`

[12] The results are compatible in principle. In practice they are compatible only on the 10-word sentence corpora used here. To get a direct comparison one must retrain all systems using the same corpus. I will address this issue in section 5.7 where full-sentence-length corpora are used.

it is easy to add more features into the model; this simply increases the number of factors in equation 4.7. To incorporate dependency information, I add a feature for word-word dependencies. In the model, this means that for a word type j with n_j tokens, we observe n_j dependency features (each being the head of one token of j). Like all other features, these are assumed to be drawn from a class-specific multinomial $\phi_z^{(d)}$ with a Dirichlet prior $\beta^{(d)}$.

To use dependency information within the framework of the BMMM described in the previous chapter, I add a multinomial distribution over word-word dependencies so that it models the number of times a word was headed by another word. In the terms of the description of the model in the last section, this is equivalent to adding a token-level observed variable $f_{jk}^{(D)}$ for each token $k = 1 \ldots n_j$ of word type j and class i:

$$\begin{aligned} \phi_i^{(D)} | \beta^{(D)} &\sim \text{Dirichlet}(\beta^{(D)}) \\ f_{jk}^{(D)} | \phi_{z_j}^{(D)} &\sim \text{Multinomial}(\phi_{z_j}^{(D)}) \end{aligned} \tag{5.2}$$

Using lexicalised head dependencies introduces sparsity issues in much the same way contextual information does. To deal with sparsity in the case of context words, the BMMM and most vector-based clustering systems use a fixed number of most frequent words as features; however, in the case of dependencies I use part-of-speech tags—either induced or gold-standard—as grouping labels. This avoids the issue of having to use only a certain number of words, as the parts of speech provide a natural way of abstracting away from the words. To obtain the dependency feature vectors, I aggregate the head dependency counts of words that have the same part-of-speech tag, so the dimension of $\phi_z^{(d)}$ is just the number of part-of-speech tags. If the parts of speech are induced I will be using the tags of the previous iteration of the system, since it is impractical to change the dependency counts each time the current part-of-speech sequence is generated.

5.4.1 Results

Figure 5.7 presents the average results on the full versions of the WSJ and the 13 languages of the CoNLL-X dataset. These results show that the inclusion of gold-standard dependencies yields significantly better results (**m-1** t = 4.09, p-value = .001 and **vm** t = 4.42, p-value = .001 using a one-sample independent t-test, as described in section 3.3.7). The performance increased in every language with the exception of Danish where **m-1** dropped by 1.2 and **vm** by 1.4 points (see table B.6 in the appendices). The

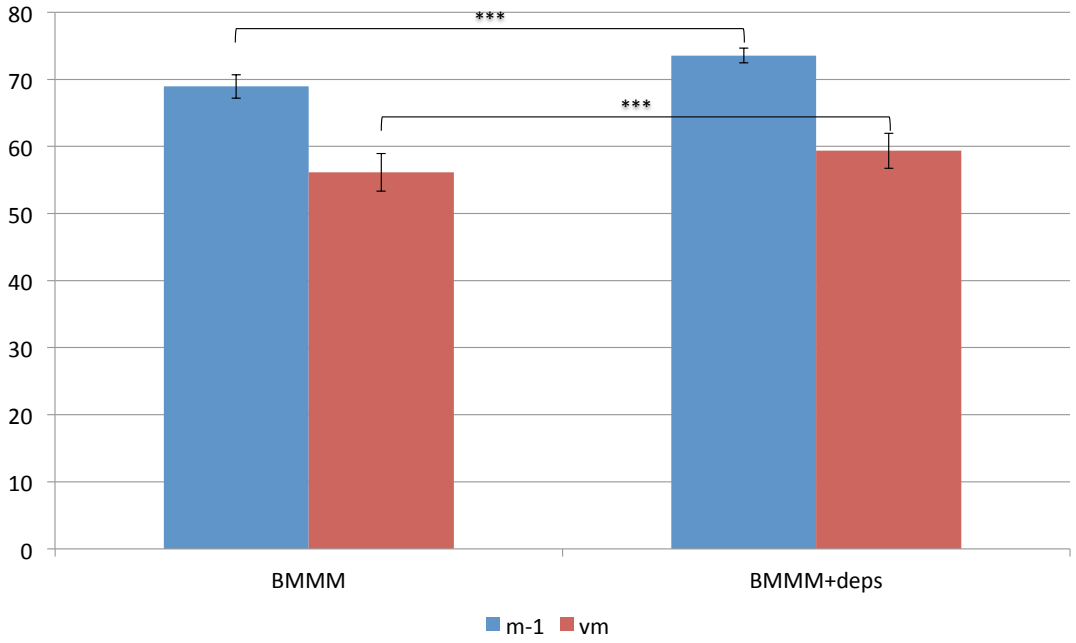


Figure 5.7: Many-to-one (**m-1**) and V-Measure **vm** part-of-speech induction results with and without the use of gold-standard dependencies. BMMM refers to the final BMMM model used in section 4.4.4 (includes morphology features). Statistical significance is measured for the difference in scores using a one-sample independent t-test.

reason for this decrease in performance might be related to a property of the language itself or to the style of dependency annotation used in the Danish dependency treebank. We will be able to test this further in the next section.

These results support the hypothesis that dependency structures produce useful features for unsupervised part-of-speech induction. Like word-alignments, used in the previous chapter, dependencies capture non-local information about the sentences, giving support to the theories of parts of speech that transcend the local word level. This is not to say that the distributional hypothesis of Harris (1951) is incomplete. In fact, if used in its original form[13] Harris' hypothesis can be said to capture syntactic as well as semantic properties. What is shown here is that by keeping the distributional features at a very manageable, local level and adding the syntactic features, the inducer is able to capitalise on the non-local nature of the parts of speech.

The next step is to see if this trend continues when the quality of the dependency structures is decreased.

[13]For two words to belong to the same cluster they must share exactly their total environments in the corpus.

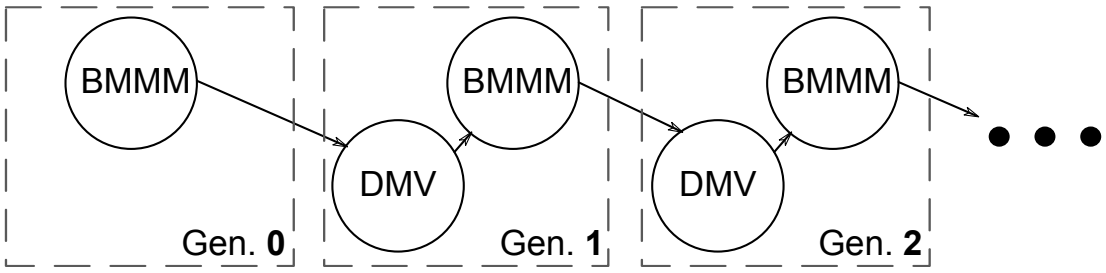


Figure 5.8: The iterated learning paradigm for inducing both part-of-speech tags and dependencies.

5.5 The Iterated Learning Framework

This section examines the effect of using induced dependencies as features for the part-of-speech inducer. Although DMV (like most unsupervised systems) depends on gold-standard part-of-speech information, I will use it in a fully unsupervised pipeline. One reason for doing so is to use dependency parsing as an extrinsic evaluation for unsupervised part-of-speech induction (Headden et al., 2008). As discussed in section 5.2.3 the quality of the dependencies drops with the use of induced tags. Instead of relying on large unannotated corpora for recovering better part-of-speech tags (Spitkovsky et al., 2011a) I use the dependency parser's output to influence the part-of-speech inducer, thus turning the pipeline into a loop.

To achieve this, I performed an *iterated learning* experiment. The term is borrowed from the language evolution literature meaning "the process by which the output of one individual's learning becomes the input to other individuals' learning" (Smith et al., 2003). Here we treat the two systems as the individuals[14] that influence each other in successive generations starting from a run of the original BMMM system without dependency information (figure 5.8). We start with a run of the basic BMMM system using just context and morphology features (generation 0) and use the output to train the DMV. To complete the first generation, I then use the induced dependencies as features for a new run of the BMMM system in the same way I incorporated the gold-standard features in section 5.4.

As there is no single objective function, this setup does not guarantee that either the quality of part-of-speech tags or the dependencies will improve after each generation. However, in practice this iterated learning approach works well.

[14]Note here that this is not directly analogous to the language evolution notion of iterated learning; here instead of a single type of individual we have two separate systems that learn/model different representations.

5.5.1 Results

Figure 5.9a presents the result of the iterated learning experiments on WSJ10 where only directed dependencies were used as features (same setup as the gold-standard dependencies). We can see that although there is some improvement in the **m-1** score after the first generation, **vm** does not improve (in fact it decreases by 0.1%). Statistical significance scores could not be calculated here since these results are from a single language (see section 3.3.7 for the significance testing assumptions).

When the undirected dependencies were used as features (figure 5.9b) the improvement over iterations is substantial: nearly 8.5% increase in **m-1** and 1.3% in **vm** after only 5 iterations. This finding seems to support the idea proposed in section 5.2.1, that headedness is not a clearly defined concept, and that the information captured by a particular annotation scheme might not correlate with performance in downstream tasks. In other words, it seems to be the case that the unsupervised systems can capture useful information (for the purposes of the part-of-speech inducer) that the gold-standard annotation marks as wrong[15].

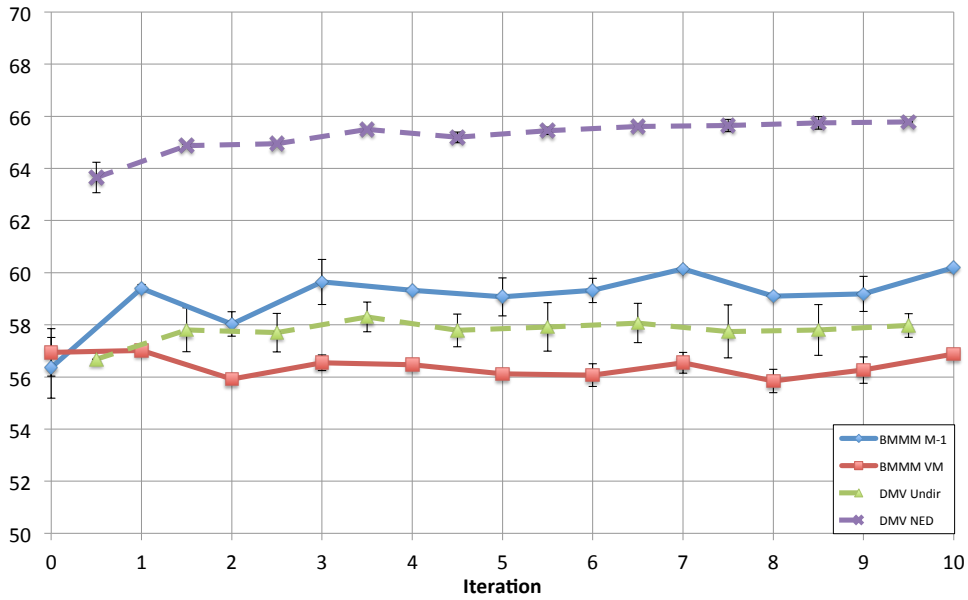
We can also see that the results of the DMV parser are improving as well: 3% increase in **ned** and 4.5% in **undir**. The improvement seems to follow the increase in quality of the part-of-speech tags. As expected the gains are smaller when only directed dependencies are used (2.1% and 1.3% for **ned** and **undir** respectivelly). This trend is to be expected, since as Headden et al. (2008) show, there is a (weak) correlation between the intrinsic scores of a part-of-speech inducer and the performance of an unsupervised dependency parser trained on the inducer's output.

5.5.1.1 Qualitative analysis of induced clusters

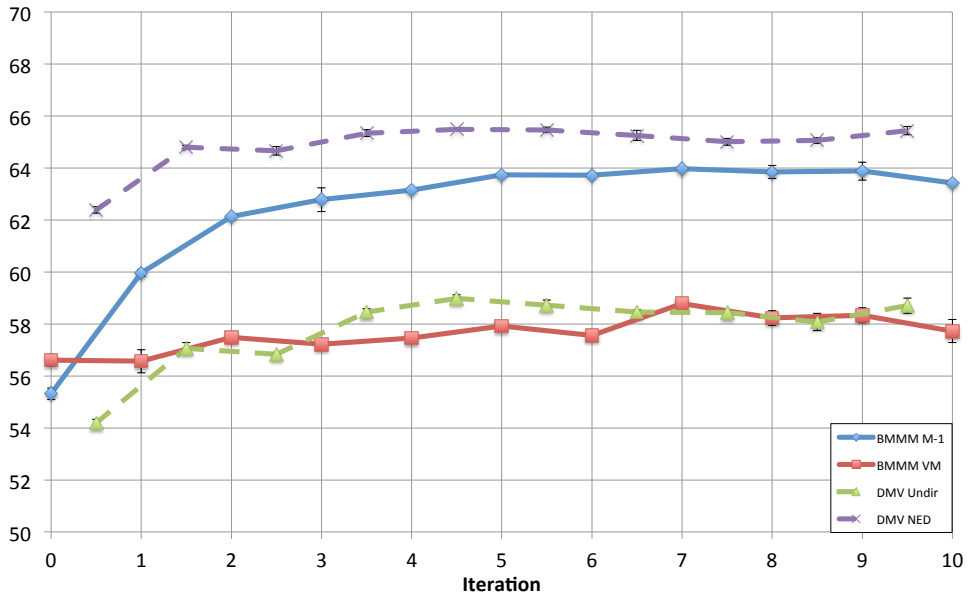
Although the unsupervised part-of-speech induction metrics have shown an undeniable increase in performance when using the iterated learning framework, it would be interesting to examine whether there are any qualitative differences between the outputs of the BMMM before and after the iterated learning. This will potentially help to show the effect of dependencies as features for part-of-speech induction.

Figures 5.10 and 5.11 show confusion matrices between gold-standard parts of speech and induced clusters for iterations 0 and 10 respectively. On first examination the confusion matrix of the basic BMMM looks more concentrated than the one of

[15] A further proof of this claim is the fact that, even with gold-standard dependencies, using directed and undirected dependencies leads to a significant improvement (in fact, the results of table B.6 were produced using both directed and undirected features).



(a) Using only directed dependencies as features



(b) Using directed and undirected dependencies as features

Figure 5.9: Iterated learning results on the 10-word version of WSJ. The performance of the part-of-speech inducer is shown in terms of many-to-one accuracy (BMMM M1) and V-Measure (BMMM VM) and the performance of the dependency inducer is shown using undirected dependency accuracy and neutral edge detection (DMV Dir and DMV NED respectively).

	NN	NNP	DT	NNS	JJ	RB	IN	VBD	VBZ	CD	PRP	VB	VBP	VBN	TO	CC	
39	2989	1808	491	891	1469	1205	248	468	319	107	283	502	254	370	8	32	11444
18	2	-	-	5	-	15	12	644	970	-	1	63	516	4	-	-	2232
2	53	300	1223	8	146	8	28	-	-	73	1	1	1	-	-	-	1842
25	4	-	-	-	-	28	1334	-	6	-	-	-	-	-	-	236	1608
17	43	40	-	1124	-	1	1	-	43	74	-	-	1	-	-	-	1327
33	21	42	682	7	19	-	4	-	-	16	-	4	-	-	-	-	795
27	21	-	-	7	19	3	1	2	-	585	-	13	-	2	-	-	653
29	448	129	-	15	33	-	-	-	-	-	-	1	3	-	-	-	629
13	3	24	16	33	-	15	-	-	-	1	455	1	-	-	-	-	548
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	596	-	596
43	5	11	147	-	-	-	3	-	-	-	306	-	-	-	-	-	472
38	2	-	-	-	-	77	111	2	13	1	-	9	8	-	-	288	511
36	8	-	-	1	142	6	-	28	2	-	-	1	-	226	-	-	414
4	187	-	-	8	-	22	-	-	-	194	-	-	3	-	-	-	414
23	-	-	-	-	-	338	-	-	-	-	-	-	-	4	-	-	342
14	28	150	11	5	68	-	11	-	5	-	-	3	-	-	-	31	312
10	111	108	-	82	1	1	-	-	1	-	-	2	-	1	-	-	307
5	1	-	-	-	1	-	-	-	-	-	-	292	20	-	-	-	314
41	-	-	-	29	5	3	-	256	-	-	-	-	-	21	-	-	314
24	20	251	-	3	2	-	-	-	1	-	-	2	2	-	-	-	281
9	20	65	-	17	150	-	-	-	1	10	-	-	-	-	-	-	263
35	32	145	-	-	1	41	-	-	-	16	-	-	-	-	-	-	235
12	27	151	-	34	1	-	-	1	-	-	-	-	-	-	-	-	214
34	2	83	-	-	2	16	7	-	-	62	-	-	-	1	-	-	173
30	-	-	-	-	-	-	-	-	-	173	-	-	-	1	-	-	174
16	22	110	-	11	18	1	-	-	-	-	-	-	-	-	-	-	162
44	10	108	-	-	7	25	-	-	-	-	-	-	-	1	-	-	151
19	121	-	-	3	-	2	-	1	-	-	-	-	-	-	-	-	127
11	-	-	-	-	-	-	-	94	-	-	-	-	-	2	-	-	96
6	36	-	-	-	2	-	-	-	-	-	-	56	-	-	-	-	94
26	24	64	-	-	-	-	-	-	-	-	-	2	-	-	-	-	90
31	8	52	-	-	13	-	3	-	-	-	-	1	-	-	-	-	77
40	9	9	-	7	26	-	13	-	-	-	-	-	-	21	-	-	85
7	-	-	-	-	-	-	-	-	-	-	-	-	-	73	-	-	73
8	-	-	-	-	1	-	-	6	-	-	-	-	-	52	-	-	59
20	39	-	-	12	-	-	-	-	-	-	-	-	-	-	-	-	51
22	24	-	-	-	18	-	-	-	-	-	-	-	-	-	-	-	42
42	-	-	-	40	-	-	-	-	-	-	-	-	-	-	-	-	40
15	28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	28
0	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12
	4348	3662	2570	2342	2144	1807	1776	1502	1361	1312	1046	953	808	779	604	587	

Figure 5.10: Confusion matrix for the output of the BMMM at iteration 0. This is an abbreviated matrix: gold-standard part-of-speech tags with frequency of less than 500 are not shown for reasons of clarity. Similarly, clusters that only corresponded to those tags are omitted.

	NN	NNP	DT	NNS	JJ	RB	IN	VBD	VBZ	CD	PRP	VB	VBP	VBN	TO	CC	
26	355	167	169	185	302	455	150	70	72	70	220	74	27	81	-	24	2421
16	21	19	-	7	26	28	1439	24	27	-	-	3	4	6	-	236	1840
44	745	91	-	272	136	117	12	11	15	10	49	54	18	37	-	-	1567
39	23	29	1265	4	45	7	23	2	-	20	-	-	-	-	-	-	1418
27	3	11	-	8	4	6	13	390	956	-	1	3	2	1	-	-	1398
25	880	281	-	69	33	1	-	1	8	1	-	32	10	-	-	-	1316
15	388	120	-	28	571	30	10	10	3	53	-	18	6	8	1	-	1246
20	91	36	-	970	7	1	-	-	9	-	1	1	-	-	-	-	1116
6	116	217	181	17	367	2	7	16	2	25	2	5	7	8	4	-	976
7	24	15	15	10	9	145	66	110	85	-	-	75	71	6	-	291	922
22	83	496	4	24	44	90	22	-	5	84	2	9	3	6	-	1	873
5	110	587	-	52	39	1	5	4	17	-	1	6	8	1	-	24	855
8	106	8	-	45	69	24	3	234	73	-	-	22	29	225	-	-	838
40	50	214	154	6	-	-	3	-	-	1	306	2	-	-	-	1	737
18	3	7	672	-	5	-	-	-	-	11	-	-	-	-	-	-	698
36	1	-	-	4	4	-	-	-	-	671	-	1	-	1	-	-	682
19	2	-	-	-	1	-	-	188	-	-	-	31	443	-	-	-	665
34	5	11	-	2	12	453	-	25	-	-	-	13	3	113	-	-	637
31	19	4	-	2	5	1	-	2	6	-	6	483	104	-	-	-	632
35	91	432	1	24	20	3	7	4	14	2	-	10	7	4	-	-	619
12	1	-	-	-	-	-	4	-	-	-	-	1	-	1	596	-	603
10	182	-	-	128	1	5	-	-	1	270	-	-	1	-	-	-	588
29	13	15	-	62	1	16	-	-	-	1	455	4	-	-	-	3	570
37	139	149	-	160	25	30	-	11	15	7	2	-	5	7	-	-	550
28	134	164	11	83	19	2	9	-	5	2	-	5	-	1	3	7	445
24	11	3	-	15	213	3	-	13	-	-	-	2	1	171	-	-	432
23	312	4	-	19	20	7	-	1	2	-	-	-	2	2	-	-	369
14	184	56	-	23	44	1	-	1	1	2	-	1	2	2	-	-	317
21	-	-	-	-	-	315	-	-	-	-	-	-	-	-	-	-	315
30	2	1	-	10	7	4	1	262	9	-	1	1	-	10	-	-	308
41	-	172	98	-	-	1	-	-	-	-	-	2	-	-	-	-	273
43	45	-	-	6	40	49	-	59	1	26	-	-	1	21	-	-	248
11	52	117	-	9	28	-	-	-	-	-	-	-	-	16	-	-	222
9	57	-	-	-	5	-	-	-	-	55	-	73	-	-	-	-	190
13	5	-	-	1	-	10	-	48	35	-	-	20	54	-	-	-	173
2	37	-	-	40	1	-	-	4	-	-	-	-	-	49	-	-	131
38	1	96	-	-	18	-	-	-	-	-	-	-	-	-	-	-	115
0	26	36	-	29	-	-	1	-	-	-	-	1	-	1	-	-	94
32	3	41	-	26	-	-	-	-	-	-	-	-	-	-	-	-	70
1	13	47	-	2	3	-	-	-	-	-	-	-	-	-	-	-	65
42	14	16	-	-	20	-	-	12	-	-	-	1	-	1	-	-	64
	4348	3662	2570	2342	2144	1807	1776	1502	1361	1312	1046	953	808	779	604	587	

Figure 5.11: Abbreviated confusion matrix for the output of the BMMM at iteration 10 (see description of figure 5.10).

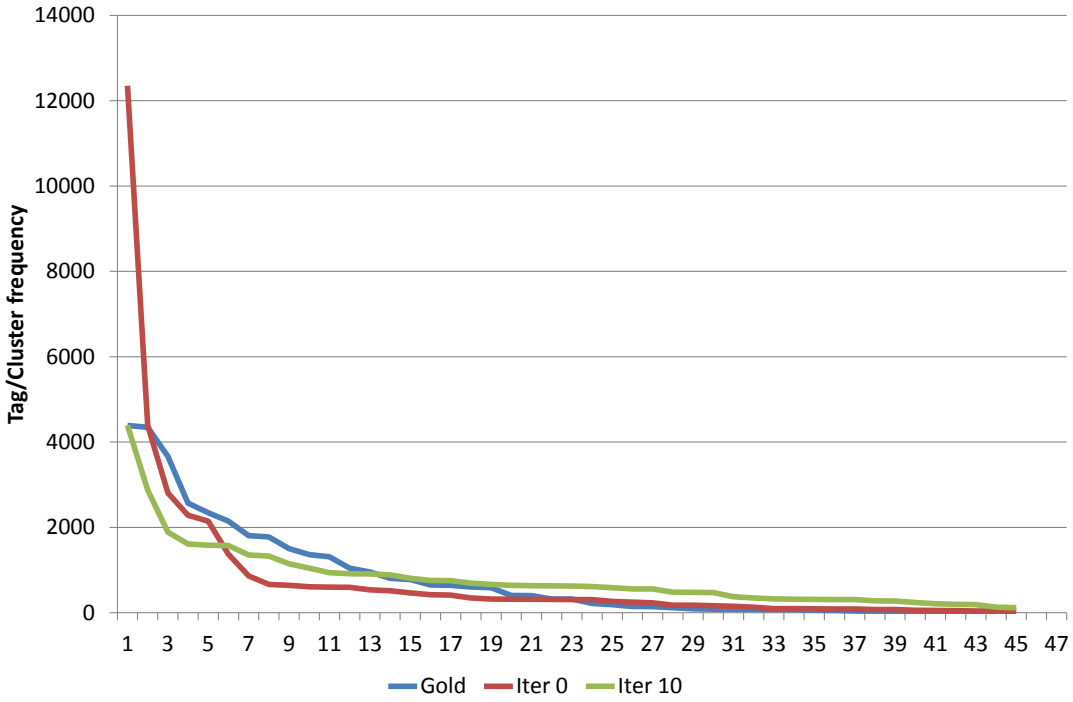


Figure 5.12: Tag/cluster frequency distribution between iterations 0, 10 and gold part-of-speech tags.

the 10th iteration. Indeed most of the clusters in generation 0 are more 'pure': for instance the three smallest clusters (42, 15 and 0) correspond to only one gold-standard part-of-speech tag each (NNS, NN and NNP respectively) whereas the three smallest clusters in generation 10 have correspondences that spread out to to 3, 4 and and 6 tags respectively.

However, this concentration of the smaller clusters comes at a cost. The biggest cluster in generation 0 (cluster 39) is disproportionately bigger than the rest and much less concentrated. It contains 11,444 words of which only 2,989 are NN (the most frequent tag). This means that cluster 39 contains 8,455 incorrectly clustered words, which more than all the errors of the 17 biggest clusters in iteration 10 put together. This is more clearly illustrated by the cluster and gold-standard tag frequency distribution shown in figure 5.12. As we can see, the cluster frequency distribution of iteration 10 more closely follows that of the gold-standard tags.

5.5.1.2 Results in other languages

As the results in figure 5.9b show, after the first five iterations the rate of improvement seems to level, so for all subsequent experiments I will be using a maximum of five

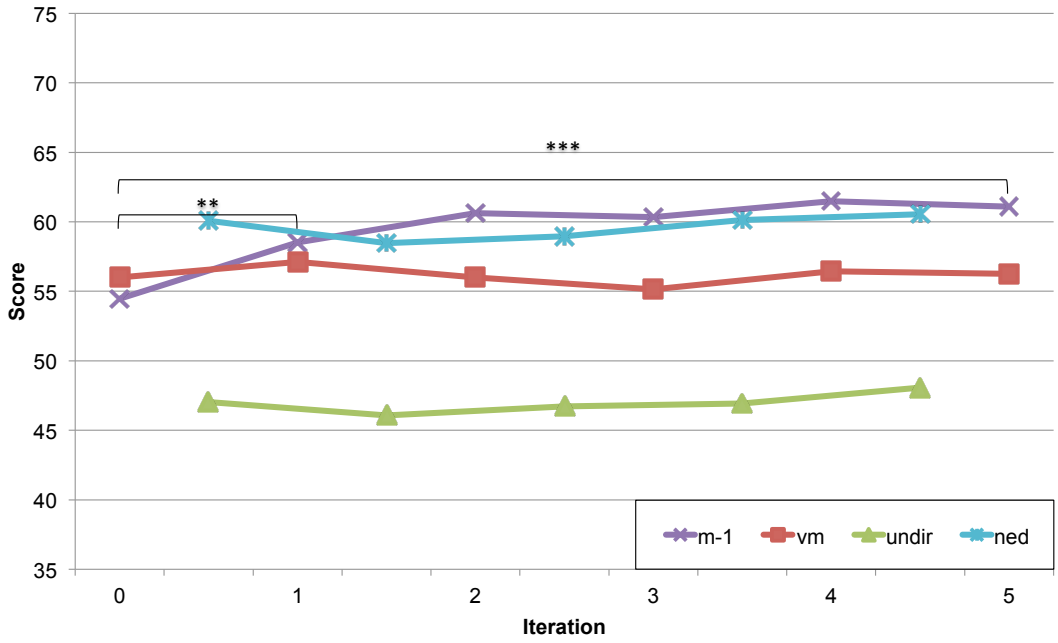


Figure 5.13: Iterated learning experiment results on up to 10-word sentences, averaged over the nine languages of the PASCAL Challenge on grammar induction (Gelling et al., 2012) using the BMMM and DMV systems. Significance tests are run between iterations 0–1 and 0–5 (α levels were adjusted using Bonferroni correction to account for the two comparisons). The significant effect shown here is only for the **m-1** scores between iterations 0–1 and 0–5. No other differences were significant.

learning iterations.

The results in the other languages are similar to those in English. Figure 5.13 shows the average performance of the part-of-speech tagger and the DMV dependency parser after five iterations over all nine languages of the PASCAL Challenge. The average **m-1** score increases continuously reaching a maximum improvement of over 7% after 4 iterations, and **vm** increases to a maximum 3% improvement over the baseline BMMM system. Interestingly, as the numerical results in table B.8 show, performance in Danish (which was the only language where performance dropped when using gold-standard dependencies) increases drastically after the 5 iterations, yielding a better **vm** score than the gold-standard dependency case. This seems to support the hypothesis that the manually annotated dependencies in Danish are not suitable for part-of-speech induction and that the unsupervised parser can find more appropriate dependencies.

Significance tests were run between the baseline (0 iteration) and the first and fifth

iterations[16]; I used Bonferroni correction (Dunn, 1961) to adjust the significance levels ($\alpha/2 = 0.025$), in order to account for the fact that I ran two (interdependent) comparisons between runs 0–1 and 0–5. The difference in **m-1** between the 0th and 1st iterations is significant (t = 4.13, p-value = .003) as well as the difference between iterations 0 and 5 (t = 5.08, p-value = .001). None of the differences in **vm** are significant (t = 1.15 and 1.79, p-value = .281 and .112 for differences between 0–1 and 0–5 respectively).

The dependency accuracy scores in figure 5.13 present a different pattern. After an initial decrease, undirected accuracy improves by over 1% at iteration 5 and NED improves by 0.5%. The overall improvement is not significant (p-value = 0.5524 for **undir** and 0.7795 for **ned** in the 5th iteration) and much less than for the WSJ corpus (Christodoulopoulos et al., 2012), hinting at the problem of over-engineering models for English. In the case of DMV, it has been shown that the initialiser also plays a crucial role in the performance of the model, and the "harmonic" initialiser of Klein & Manning (2004) is not ideal for all languages (Gimpel & Smith, 2012).

5.6 Using a state-of-the-art parser

The main reason to use the basic DMV parser was its simplicity. However, in terms of parsing performance the basic model has been superseded by a number of newer systems. For this reason I will replace the basic DMV model with a state-of-the-art parser and compare the results in the iterated learning task. The system I chose was the Tree Substitution Grammar DMV parser (*TSG*-DMV) of Blunsom & Cohn (2010) as it was one of the best performing systems across all languages in the PASCAL Challenge on grammar induction (Gelling et al., 2012).

The main intuition behind the *TSG*-DMV system is the use of a more complex grammar than the original DMV. *Tree Substitution Grammar* (TSG) is a variant of *Tree-Adjoining Grammar* (TAG, Joshi et al., 1975)[17] where derivations are built by combining tree fragments called *elementary trees* at non-terminal substitution sites called *frontier non-terminals* (see figure 5.14a for an example).

[16] Another test that could have been used for the iterated learning results is analysis of variance (ANOVA) which generalises the t-test to compare the means of more than two groups and account for the multiple comparisons; however, the number of samples was not enough to provide a powerful ANOVA analysis.

[17] TAG is a mildly context-sensitive formalism which means it is more expressive than the context free grammar of the original DMV; however, TSG does not use the *adjunction* operator of TAG. Therefore it is hard to tell how much more expressive power TSGs have over CFGs.

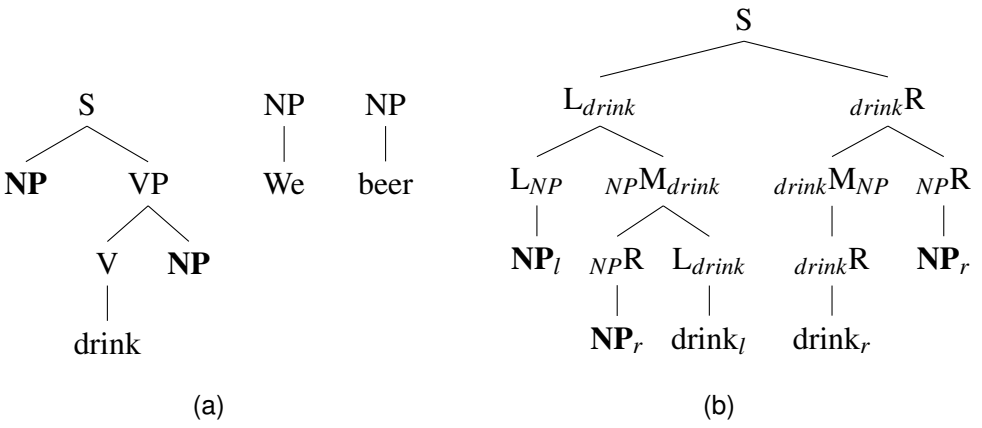


Figure 5.14: Tree Substitution Grammar examples. 'NP→We' and 'NP→beer' are *elementary trees* and the bold nodes are *frontier non-terminals* (substitution sites). Figure (b) shows the split-head version of the 'S' elementary tree of (a).

In order to induce dependency structures efficiently, Blunsom & Cohn (2010) build their TSG structures based on the underlying lexicalised CFG-DMV trees. More specifically, they use a variant of the *split-head* constructions (Eisner, 2000) that allows them to parse in polynomial $O(n^3)$ time, by splitting each terminal and processing left and right dependencies independently (see figure 5.14b for an example).

To ensure that the model does not generate a large number of highly detailed trees in the induced grammar, Blunsom & Cohn (2010) define a hierarchical non-parametric model over the space of the TSG trees. The model is a four-level Pitman-Yor Process (Teh, 2006), each of which can be thought as the non-parametric extension of the Bayesian Dirichlet model presented in section 4.2[18].

5.6.1 Results

Figure 5.15 presents the average results over all nine languages for the combination of the *TSG*-DMV and BMMM systems. The performance for part-of-speech induction is slightly better than before[19]—and increases continuously over both metrics, reaching a significant difference by iteration 5 ($t = 5.53$, p-value = .000 for **m-1** and $t = 2.69$ p-value = .027 for **vm**); this is not reflected in the performance of the *TSG*-DMV, which keeps decreasing across iterations, but the differences are not significant ($t = -1.37$, p-value = .207 for **undir** and $t = -1.31$, p-value = 0.227 for **ned**).

[18]More precisely, the Pitman-Yor Process is a generalisation of the Dirichlet Process which in turn is the infinite-dimensional extension of the Dirichlet Distribution of the BMMM.

[19]A difference of 0.5 in **m-1** score—not statistically significant.

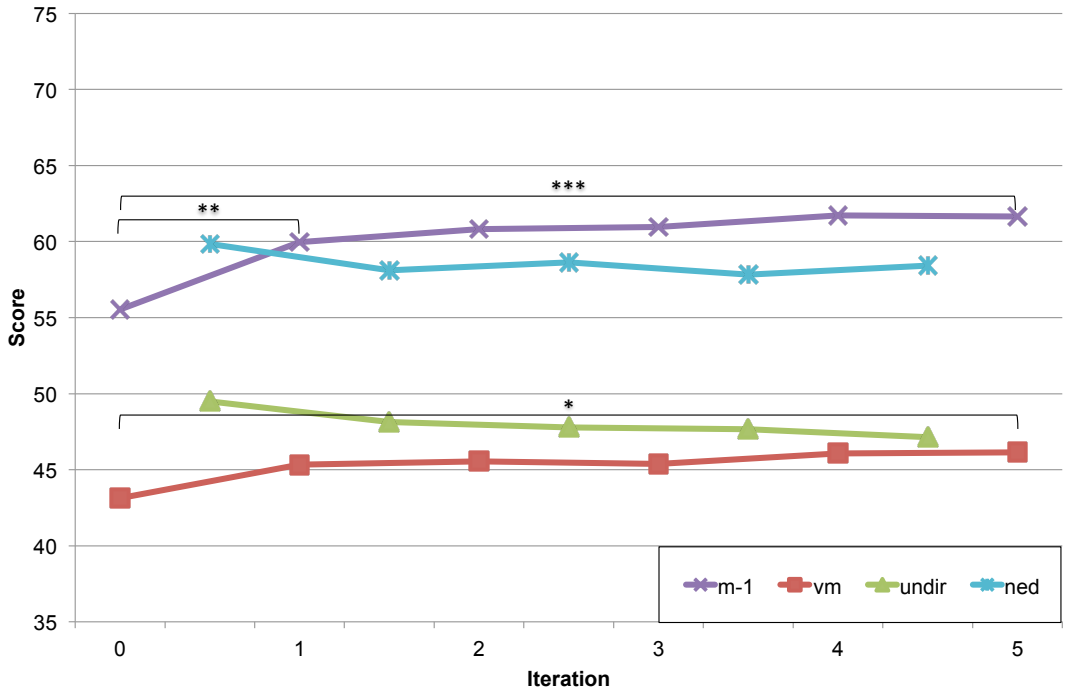


Figure 5.15: Iterated learning experiment results on up to 10-word sentences, averaged over the nine languages of the PASCAL Challenge using the BMMM and TSG-DMV systems. Significance tests are run between iterations 0–1 and 0–5 (with Bonferroni correction). The significant effect shown here is for the **m-1** scores between iterations 0–1 and 0–5 and for **vm** between iterations 0–5. No other differences were significant.

The findings from the results of the iterated learning experiments suggest that there is a correlation between the performance of the induced dependencies and the induced part-of-speech tags; also, the iterated learning framework allows the BMMM to capitalise on the dependency parser, and vice-versa. When DMV is replaced by a better parser, the quality of the induced part-of-speech tags increases, suggesting that the part-of-speech induction task reflects the results of the intrinsic evaluation of the parsers (in accordance with the findings of Headden et al., 2008).

However, the ever decreasing performance of the *TSG*-DMV parser—despite not being statistically significant—does not reflect the quality of the induced tags. One possible explanation is that the *TSG*-DMV model uses a more complex lexicalised grammar and relies less on the quality of the part-of-speech tags. It is also important to remember that *TSG*-DMV was developed using gold-standard parts of speech and its behaviour with unsupervised tags has not been examined.

5.7 Beyond 10-word sentences

One obvious limitation of the iterated learning experiments presented above is the use of short (up to 10-word) sentences. Not only does it make the comparison with other part-of-speech induction systems difficult, but also reduces the amount of available data, in some cases quite dramatically (e.g. in the original Czech corpus there are 1,503,739 words whereas in the 10-word versions there are only 161,174). There is an implicit comparison to other systems since these experiments show an improvement on the BMMM baseline, which in turn has been compared to a number of other systems in section 4.4.4. However, given the reduction in available data, and the fact that distributional statistics are greatly affected by it, a more comprehensive evaluation is required.

It has been common practice in the dependency induction task to use short-length sentences due to the complexity of the task. Training on full length sentences is still a computationally intensive task—and this is especially true for the more complicated systems. Furthermore, as Blunsom & Cohn (2010) report, it is much harder for unsupervised models to learn from longer sentences since they are much more ambiguous (see section 5.2.2). However, even though the systems can only be trained on sentences with < 10 words, there is no reason why these systems should be tested on small sentences as was traditionally the case. In the past couple of years this trend seems to be declining. For instance see Spitkovsky et al. (2010a); Blunsom & Cohn (2010);

Spitkovsky et al. (2011b) and the PASCAL Challenge on grammar induction (Gelling et al., 2012) that reported results on 10-, 20- and full-length sentences.

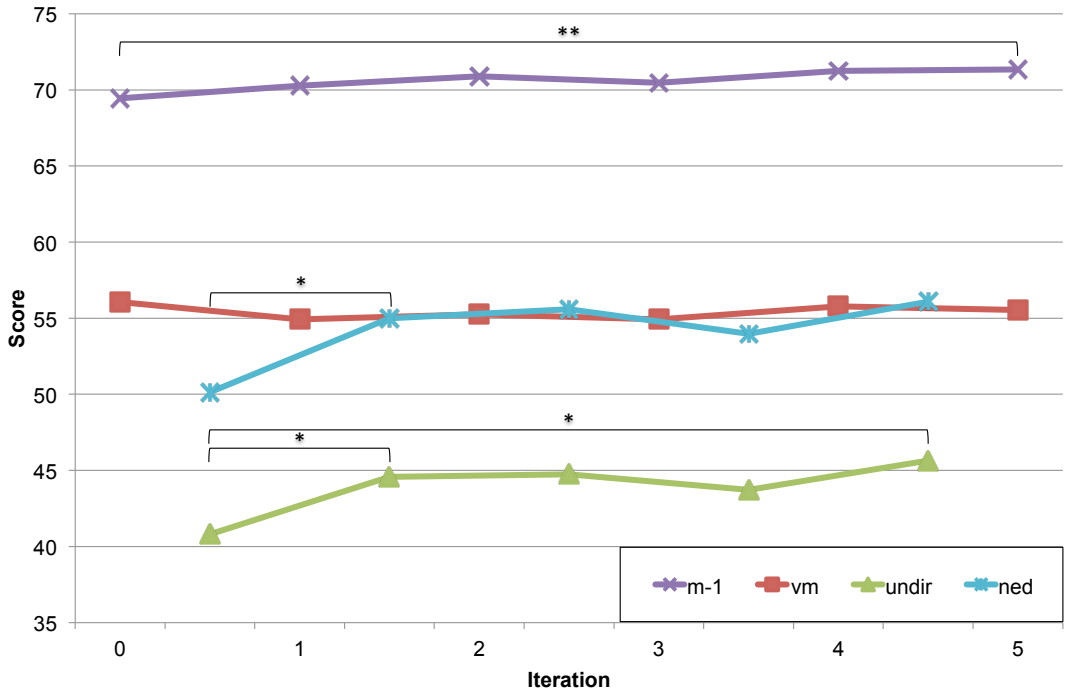
To investigate the effect of longer sentences on the iterated learning setup, I will be testing both the DMV and the *TSG*-DMV parsers on full-length sentences. I will also try training the DMV model on full-length sentences to test the increased ambiguity claim of Blunsom & Cohn (2010). The complexity of the *TSG*-DMV model makes it intractable to train on longer sentences.

5.7.1 Results

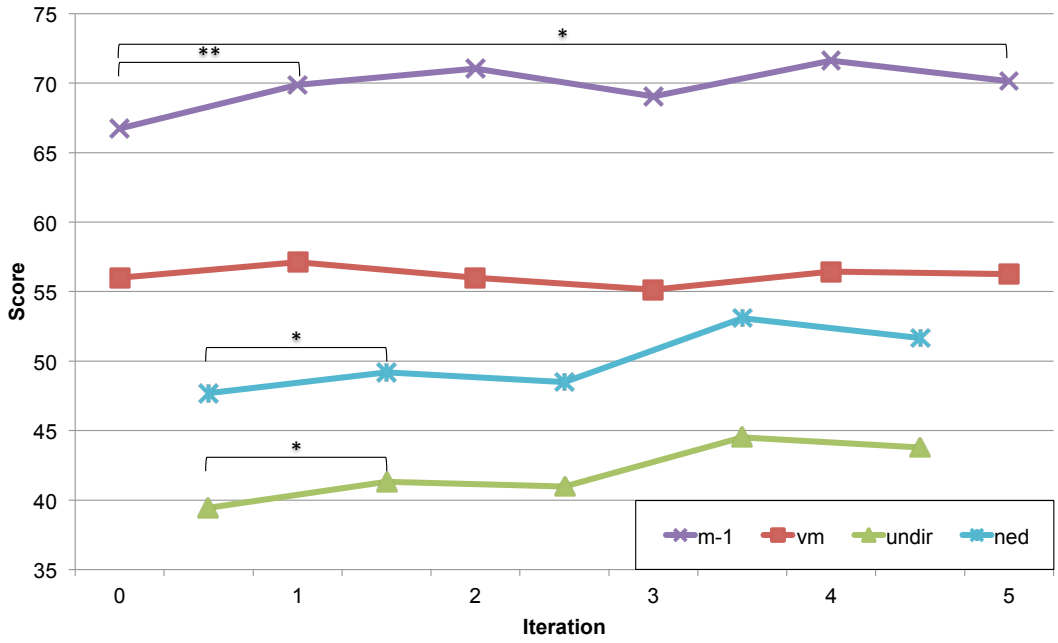
Figures 5.16–5.17 show the average results of the iterated learning experiments on sentences of all lengths. Overall the scores of the BMMM system are higher—in most cases beating the performance with gold dependencies (see table B.12–B.16)—since it takes advantages of the larger amount of data available. However, the gains in performance after the 5th iteration are much smaller, but still significantly different from the scores of the baseline model, at least for **m-1** (p-value = 0.0050 for the DMV trained on 10-word sentences, 0.0146 for the DMV trained on all sentence lengths, and 0.0020 for *TSG*-DMV).

Similarly to the case of 10-word sentences, tables B.12 and B.16 show that unlike the basic DMV, the *TSG*-DMV system has a hard time generalising its results to longer sentences when trained only on up to 10-word sentences. The parsing accuracy drops continuously (but the differences are not significant), despite the relatively stable performance of the BMMM. One reason for this might be that the structures induced by the *TSG*-DMV are highly suitable for shorter sentences (hence its superior performance on the 10-word tests), whereas longer sentences might contain fundamentally different structures. Some examples include long-range and crossing dependencies, both of which are rare in short sentences.

Validating Blunsom & Cohn's claim, the performance of the DMV parser decreases slightly when trained on all sentence lengths, but similarly to the 10-word sentences, its accuracy keeps increasing over the iterations and this does not affect the performance of the BMMM (see table B.14). One possible explanation is that even when using the full corpus for training, DMV seems to be learning the same (localised) structures—the longer data simply adding some noise.



(a) 10-word sentences training



(b) Training on all sentences

Figure 5.16: Iterated learning experiment results on all sentence lengths, averaged over the nine languages of the PASCAL Challenge using the BMMM and DMV systems, trained with 10-word sentences and all sentences.

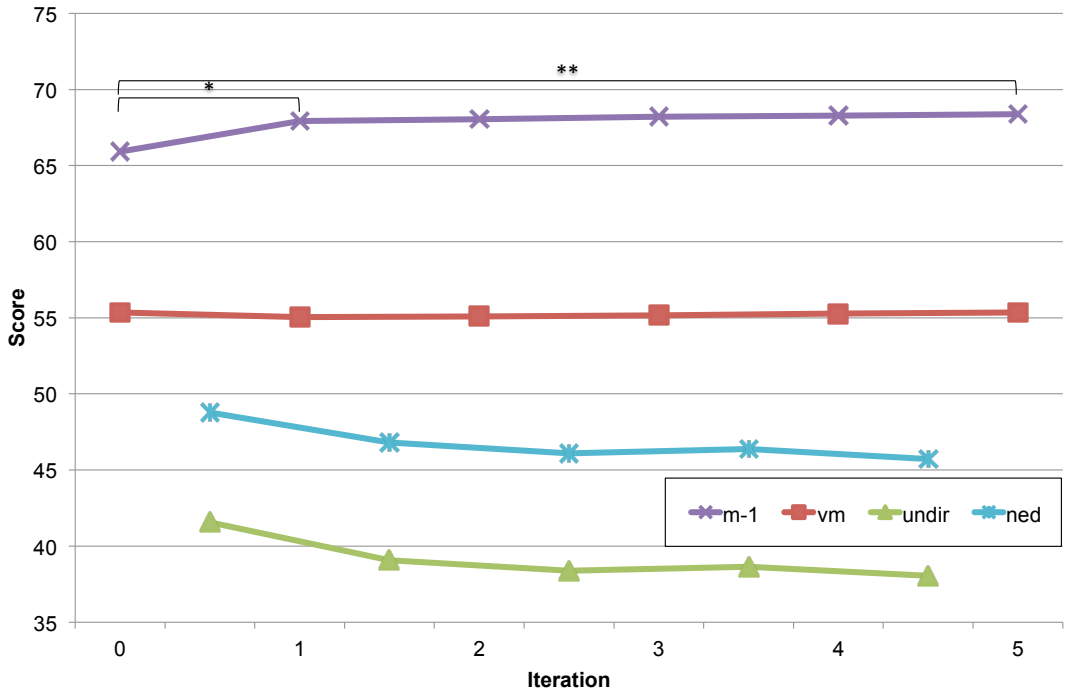


Figure 5.17: Iterated learning experiment results on all sentence lengths, averaged over the nine languages of the PASCAL Challenge using the BMMM and *TSG*-DMV systems, trained with 10-word sentences.

5.8 A Fully Joint Model

The obvious extension of the iterated learning system would be a full joint model of part-of-speech and dependency induction. However, unlike supervised models (Li et al., 2011; Auli & Lopez, 2011) the joint unsupervised search space (all possible dependency structures $\times$ all possible part-of-speech-tag sequences) is prohibitively large. One possible solution (Cohen et al., 2011) is a joint decoding process where the dependency model is trained on a part-of-speech-tag *lattice* that limits the possible tag sequences. However, this method uses a tag dictionary[20] whereas we are focusing on induction without external knowledge so had to develop an alternative approximation.

First, consider how the full joint inference system would work. In the standard BMMM model, within every step of the sampling process a part-of-speech tag is chosen for each word type as a draw from a multinomial distribution that is formed from the class mixing priors and the feature likelihoods. As explained in section 4.2, the features are assumed to be conditionally independent, and therefore the total feature

[20] A list of all possible tags seen with a particular word type, in this case for separate set of training data (see section 3.1).

likelihood is simply the product of different observed features. Under this independence assumption we can simply treat the probability of the dependency structure of the whole corpus given a part-of-speech-tag sequence as another likelihood factor in equation 4.7, repeated here:

$$P(\vec{f}_j|\mathbf{f}_{-j}, z_j = z, \mathbf{z}_{-j}, \beta) = \prod_{k=1}^{F} \frac{\Gamma(m_{jk,z} + \beta)}{\Gamma(m_{\cdot,z+F\beta})}$$

However, this means that for every possible tag the sampler considers we need to repack the parse charts for the whole corpus to compute the likelihood. (Note that the sampler reassigns all tokens of a given word type to a new tag at the same time, which changes the DMV probabilities of many sentences at once.)

Algorithm 3 The joint inference algorithm.

```
INITDMV-LEX
for wordType = 1 → M do
    wordClass ⇐ classAssign[wordType]
    UNASSIGNCLASS(wordType, wordClass)
    for class = 1 → Z do
        prior[class] ⇐ CLASSPRIOR(class)
        tagLL[class] ⇐ FEATLIKELIHOOD(class)
        depLL[class] ⇐ scale(DMV-P(classAssign))
        p[class] ⇐ prior[class] × tagLL[class] × depLL[class]
    wordClass ⇐ MULTINDRAW(p[class])
    classAssign ⇐ ASSIGNCLASS(wordType, wordClass)

function DMV-P(classAssign)
    AGGREGATECOUNTS
    for sent = 1 → S do
        for span = 1 → d do
            l_sent ⇐ ∏_{D_d ∈ deps_h} P(D_d(h))        ▷ Sentence likelihood (eq. 5.4)
        l_total ⇐ l_total × l_sent
    return log(l_total)
```

Since this method is computationally infeasible, I define an approximation to the chart-packing step that estimates the probability of a full dependency tree by multiplying the probabilities of all the subtrees up to a specific depth. This allows for a

reduction of the complexity of the chart-packing step from $O(n^3)$ to $O(n^2)$ since the maximum dependency span is now a constant. Algorithm 3 shows the PoS sampling process with the embedded dependency step.

The algorithm starts by initialising a lexicalised version of the DMV[21] using the harmonic initialiser (line 1). After a full iteration of EM is performed using the same inside-outside algorithm of the original DMV, we have the model parameters (e.g. $P(w_1|w_2)$) and a set of word-word dependencies.

Then, at each step of the part-of-speech sampling process, the sampler considers every possible class for each word type and (keeping all other class assignments fixed) it generates a tag sequence based on the current proposal. Using this temporary tag sequence we aggregate the expected counts of the lexicalised DMV model (summing over the probabilities of words with the same part-of-speech tag), thus creating a part-of-speech-tag-based re-initialised model (line 13).

With the initialised DMV model parameters we run one partial inside step (filling up the chart up to depth d—line 16) to estimate the approximate likelihood of the corpus under the current tag-sequence.

More formally, for each word type j, we need to calculate the product of the likelihood of the model over all the sentences $s = 1,\ldots,S$ given the proposed tag sequence $\mathbf{z} = \{z_j, \mathbf{z}_{-\mathbf{j}}\}$, where the probability of a specific sentence is the sum of the probabilities over all possible dependency trees $deps_s$ rooted at $\diamondsuit$. Given the independence of each feature *kind* of the BMMM (see equation 4.10) we only need to focus on the dependency features $f^{(D)}$ (see equation 5.2):

$$P(f_j^{(D)}) = \prod_{s=1}^{S} P(s|z_j = z, \mathbf{z}_{-\mathbf{j}}) = \prod_{s=1}^{S} \sum_{D \in deps_s} P(D(\diamondsuit)|z_j = z, \mathbf{z}_{-\mathbf{j}}) \tag{5.3}$$

This approximation is based on the assumption that the probability of the sentence is proportional to the product of the probabilities of the local trees of a certain depth d:

$$P(D(\diamondsuit)|z_j = z, \mathbf{z}_{-\mathbf{j}}) \propto \prod_{D_d \in deps_h} P(D_d(h)|z_j = z, \mathbf{z}_{-\mathbf{j}}) \tag{5.4}$$

for all dependency trees rooted at h with depth at most d.

Note here that the two models define their probability distributions over two different things. The dependency model defines its generative probability over the entire sentence strings, whereas the BMMM generates sets of features for each word token/type. This means that at this stage the scale of the dependency log-likelihood is

[21] For this joint model I created my own implementation of the DMV model, based on Franco M. Luque's Python version: `http://cs.famaf.unc.edu.ar/~francolq/en/proyectos/dmvccm`

very different from that of the other features used by the tagger. To make the log-likelihood factors comparable, I apply a linear transformation ($f(x) = \mu_j + \delta$) so that the maximum (and minimum) value of the dependency log-likelihood coincides with the maximum (and minimum) value of the other features' log-likelihood.

$$\mu_j = (\max(\mathbf{tagLL}_j) - \min(\mathbf{tagLL}_j))/(\max(\mathbf{depLL}_j) - \min(\mathbf{depLL}_j))$$

The scaled log-likelihood is then calculated using:

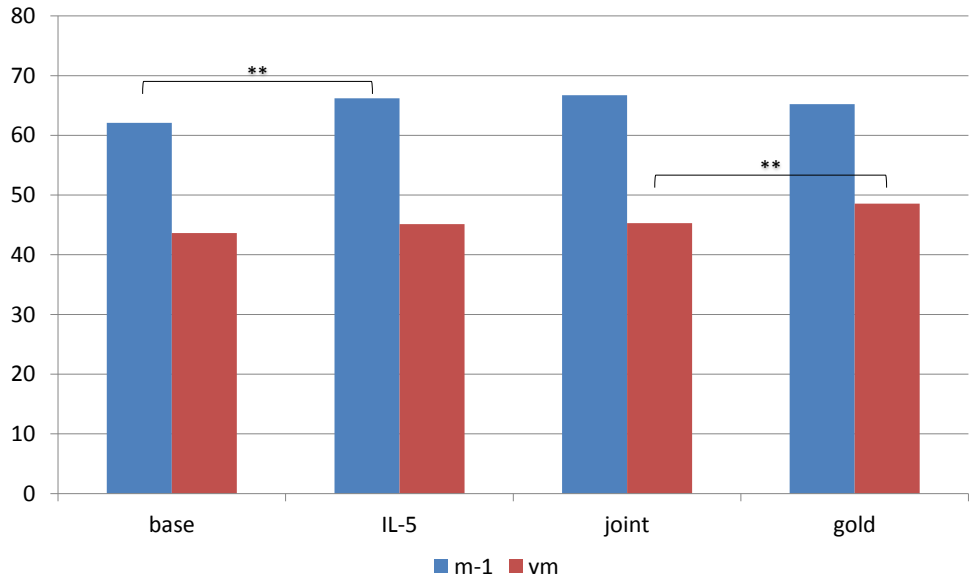
$$depLL_{ij} = \mu_j \times (depLL_{ij} - \min(\mathbf{depLL}_j)) + \min(\mathbf{tagLL}_j)$$

The posterior probability of the class is then computed by multiplying the class prior, feature and dependency probabilities. Repeating the process for all the classes I construct a multinomial distribution from which the new class for the current word type is drawn.

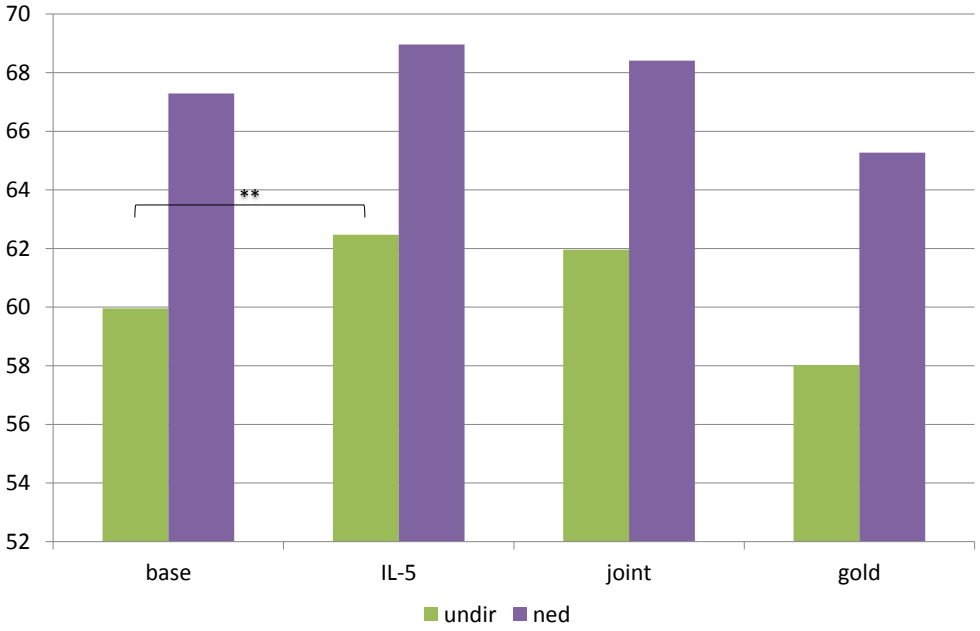
There are two issues with this approximation. First, by using sentence spans up to d the model is incapable of creating long-range dependencies. However, these constructions are quite rare, at least in the English corpora (Rimell et al., 2009) and even more so in the smaller-sentence versions of the corpora used here. Second, the use of a maximum span means that we cannot construct a full chart for each sentence and therefore cannot perform a full EM iteration (since we cannot create the outside scores). This means that the DMV model will only improve slightly over the harmonic initialiser every time. However, since we are interested in comparing the different proposed tag sequences, it is sufficient to compute the relative differences of the partial inside scores.

5.8.1 Results

Figure 5.18 shows the results of the joint inference on the part-of-speech and dependency induction tasks. Due to the memory requirements of the inference algorithm, the joint model could not run on the larger corpora (Chinese, Czech and German). Even with the smaller corpora, due to time restrictions, the model was able to run for just one sampling iteration. Nevertheless the results in the remaining languages show the effectiveness of the joint approach: on the part-of-speech induction it improves 0.5% (**m-1**) and 0.2% (**vm**) on average over the performance of the final iterated learning model. The improvement is greater over the BMMM model without dependencies (4.6% on **m-1** and 1.7% on **vm**). The difference between the iterated learning model and joint



(a)



(b)

Figure 5.18: Part-of-speech (5.18a) and dependency induction results (5.18b) on CoNLL data after 5 generations of iterated learning (**IL-5**) and for the joint inference. **base** for the part-of-speech induction task is the BMMM system trained on just context and morphological features (generation 0) and **gold** is the BMMM using gold-standard dependencies. For dependency induction, **base** is the DMV system trained on the baseline BMMM and **gold** is the DMV trained on gold-standard parts of speech. Significance results are shown for successive systems (**IL-5** vs. **base**, **joint** vs. **IL-5**, etc.).

model is not significant ($t = 0.93$, p-value = .375 for **m-1**; $t = 0.31$, p-value = .767 for **vm**) whereas the improvement of both models against the baseline is: the difference between the iterated learning model and the baseline was tested at $t = 3.51$, p-value = .007 and $t = 2.4$, p-value = .04 for **m-1** and **vm** respectively.

We can see further evidence of the overall effectiveness of using unsupervised dependency features with the BMMM (despite using a sub-optimal dependency parser) when we look at the results of the model that uses gold-standard dependencies. While on average **vm** is significantly lower ($t = 4.05$, p-value = .003), the joint model performs 1.5% better on **m-1** but the difference is not significant ($t = -1.45$, p-value = .181).

The situation is similar for the dependency induction scores. While the joint model does not outperform the iterated learning one, both of them produce better dependency parses than the DMV using gold-standard parts of speech. On average the improvement is about 3% for both **undir** and **ned** but the differences are not significant ($t = -1.46$, p-value = .177 and $t = -1.6$, p-value = .145 respectively).

These results are promising; nevertheless, further investigation is required to provide more efficient methods of sampling and convergence. For the time being, the iterated learning method provides a more viable option for combining part-of-speech and dependency induction.

5.9 Conclusion

In this chapter I have presented an extension of the BMMM system that used dependency structure as features for part-of-speech induction. In this way, and by taking advantage of the interaction between dependency inducers and part-of-speech tags, I have developed an iterated learning method that combines a dependency induction system with the BMMM to produce higher quality, syntactically-aware parts of speech.

Next, I experimented with the dependency induction part of the iterated learning system. I replaced the basic DMV model with a state-of-the-art parser and instead of relying on sentences of up to 10 words—as was the case for most dependency parsers until recently—I used full-length sentences for both training (for DMV only), and testing. The results showed that a better dependency model results in an increase in the quality of the induced tags (more so than the basic DMV model) and in some cases a better performance than was achieved by using gold-standard parts of speech.

The iterated learning method also helped the DMV system. Its performance kept

increasing with each iteration, reflecting the increase in quality of the part-of-speech tags. This was true not only in the case of 10-word corpora but also when full sentences were used. However, unlike the basic DMV, the performance of *TSG*-DMV kept decreasing throughout the iterations, unable to benefit from the increasing quality of the induced tags, suggesting that it was relying more on lexical information (unlike the basic version of DMV). Also, when tested with the full-length sentences, its performance was significantly worse than that of the basic DMV. This seems to suggest that the basic DMV is more flexible and generalises more easily over longer sentences, or conversely, that because of its complexity the *TSG*-DMV model finds it harder to generalise over long sentences—having been trained on up to 10-word sentences only.

I have also presented a preliminary attempt to create a fully joint model of parts of speech and dependencies, showing that a closer interaction of the these two levels is indeed helpful. The results also suggest that the iterated learning method is a viable proxy for the fully joint model since their performances are not significantly different. However, the fully joint approach is much more computationally intensive and difficult to extend; the iterated learning method is a more viable alternative.

The experiments of this chapter have shown that, using the iterated learning method, it is possible to connect multiple levels of linguistic structure, achieving more accurate analyses. I will further explore this interaction in the next chapter by revisiting the morphology and alignment features, already used in the BMMM.

CHAPTER 6

Using Iterated Learning for Morphology and Word Alignments

Ita verba [...] quarum rerum signa essent, paulatium colligebam[1]

Augustine (398, 1.8.13)

6.1 Introduction

In addition to part-of-speech and dependency induction covered in chapters 4 and 5 respectively, I will concentrate on two more areas of unsupervised NLP research: *morphological segmentation* and *word alignment*. These areas were chosen primarily because of the immediacy of the connections that we can draw between all of them and this study serves as a starting point for our discussion about the interconnected nature of these areas. It should be easy to draw connections between some of the areas described here and other areas such as named entity recognition, anaphora resolution or semantic parsing but these connections will have to be addressed in future work.

Before presenting the results of the iterated learning experiments, I will briefly present some background information for these tasks and discuss some issues regarding their evaluation.

[1] In this way, little by little, I learnt to understand what things the words [...] signified

6.2 Morphological Segmentation

Morphemes are considered the fundamental units of language[2] providing syntactic and semantic information at a more atomic level than words. This is especially true in languages with productive morphology, either agglutinative, inflectional or reduplicative. Even in the case of isolating languages (with little or no morphology) we can consider the words as morphological stems and apply the same syntactic/semantic treatment.

Morphological segmentation is thus considered the first level in the NLP hierarchy (see figure 1.1). Supervised statistical approaches or language-specific rule-based approaches perform extremely well for a small selection of languages (e.g see Eryiğit & Adalı, 2004 for Turkish, Kaalep, 1997 for Estonian and Sgarbas et al., 1995 for Greek) but in the last decade there has been a rising interest in unsupervised, language-independent systems. Goldsmith (2010) and the Morpho Challenge 2005 competition (Kurimo et al., 2006) provide a good overview of the landscape[3].

Morphemes exist in two forms: inflectional or derivational. The main difference is whether a morpheme changes the meaning or part of speech of the word it is attached to. Inflectional morphemes modify the grammatical properties of the word, without changing its meaning (or part of speech):

(6.1) a. work/V + ed → worked/V
b. word/N + s → works/N

On the other hand, derivational morphemes change the part of speech of the main word as in (6.2-a); or change the meaning of the main word as in (6.2-b).

(6.2) a. align/V + ment → alignment/N
b. under + stand/V → understand/V + ing → understanding/N

Another classification of morphemes can be made by examining their placement. In English morphology is mostly concatenative; that is, the morphemes are attached at the beginning (prefixes) or the ending of the word (suffixes). In other languages (like ancient Greek in the following examples), morphemes can be *infixed* either by inserting

[2] At least in NLP; there are some who disagree with this statement and propose alternative atomic units. One example is the *Nanosyntax* theory (Starke, 2009).

[3] It important to distinguish between the task of morphological segmentation—also known as surface segmentation—and morphological *analysis* (Kurimo et al., 2010) where the goal is not only to distinguish the different morphemes but to identify their roles (e.g. 'books' will be analysed as 'book' + plural). Here we are interested in the segmentation task only.

the morpheme inside the stem of the word as in example (6.3-a), or by reduplicating part of the stem, example (6.3-b):

(6.3) a. **la**-m-bánō (I receive) → é-**lă**-bon (I received)
b. **dé**-rkomai (I see) → **dé**-**do**-rka (I saw)

The majority of the systems deal only with concatinative morphology (either derivational or inflectional), and only a few (e.g. Demberg, 2007) can handle the full range of morphological phenomena such as stem changes, reduplication, infixation etc.

One of the most successful segmentation strategies used is based on the principle of Minimum Description Length (MDL, Rissanen, 1978) which states that the best modelling hypothesis for a given set of data, is the smallest (the one that leads to the biggest compression). It has been used heuristically by Goldsmith (2001), and in a probabilistic maximum-a-posteriori (MAP) framework by the Morfessor system (Creutz & Lagus, 2005, 2006). Morfessor tries to minimise the size of the lexicon M that contains all the morphemes, by maximising the following equation:

$$\arg\max_{M} P(M|corpus) = \arg\max_{M} P(corpus|M)P(M)$$

where

$$P(M) = P(lexicon, grammar)$$

The joint probability of the lexicon and the grammar is then decomposed into progressively smaller units that capture a hierarchical description of the morphemes. The units correspond to properties such as the length of a morpheme, its frequency, its left and right context perplexity and its type (prefix, stem, suffix or non-morpheme) and are generated after hypothesising all potential morphemes that generate the lexicon M. Apart from being computationally efficient and empirically successful, this information-theoretic approach (also called the Neo-classical model) has been proposed as an alternative theory of morphology by Milin et al. (2009) for inflectional and Moscoso del Prado Martín et al. (2004) for derivational systems. For a review see Blevins (2013).

Other approaches to morphology segmentation calculate the segmentation boundary (or boundaries) probability by examining the probabilities of the transitions between letters. Some examples include the systems of Goldwater et al. (2006b), Demberg (2007) and the joint part-of-speech induction/morphology segmentation system of Sirts & Alumäe (2012), described in more detail in the next section.

6.2.1 Influence of Parts of Speech on Morphology Segmentation

There are only a few approaches that that take syntactic information into account. The system described by Lee et al. (2011) learns syntactic category information jointly with segmentation and shows that the syntactic information is indeed helpful but their syntactic categories are few (only 5) and therefore too coarse to be useful to downstream tasks.

The system that I am going to be using is described by Sirts & Alumäe (2012). This is also a joint part-of-speech induction/morphology segmentation learning system but in this case there is constraint on the number of syntactic categories learnt and therefore the categories can more closely resemble traditional part-of-speech tags.

Sirts & Alumäe (2012) use the Hierarchical Dirichlet Process (HDP) framework of Teh (2006) to generate an HMM with parts of speech as hidden states and words as emissions, which in turn generate the word segments via a separate HDP. The two HDPs are conditioned on each other, meaning that the inference of the segments is calculated given the (infinite) distribution of parts of speech and vice-versa. The use of non-parametric models allows the system to infer both the number of tags and morphemes. The base distribution of each emitted morpheme is produced in a similar way to Goldwater et al. (2006b), by a Dirichlet distribution over characters multiplied by a geometric distribution over the morpheme length.

Since the two parts of the system (the morphology and the part-of-speech tags) are induced by separate, but interdependent distributions, the inference of part-of-speech tags can be decoupled from that of segmentation. This means that the part-of-speech sequence can be fixed to a predefined input and still keep the interdependency between the segments and the part-of-speech tags. As with the rest of the systems that we will examine, this modification is key to the reverse interaction between part-of-speech induction and morphological segmentation.

6.2.2 Influence of Morphology on Part-of-speech Induction

The influence that morphological information has over part-of-speech induction has been demonstrated by the systems of Clark (2003), Berg-Kirkpatrick et al. (2010) and Blunsom & Cohn (2011) described in section 3.4.1, as well as in the BMMM system (section 4.3.3). Most of the high-performing part-of-speech induction systems either model morphology directly, or use it as a feature, because morphological structure is isolated within word boundaries.

6.2.3 Evaluation

Morphological segmentation systems are evaluated against gold-standard word segmentations using precision, recall and f-score. Usually there are multiple gold-standard segmentations and the systems are given full points if they match any of them (as was the case in Morpho Challenge 2005).

One problem with evaluating morphology segmentation stems from the fact that inflectional and derivational morphology are not distinguished by the unsupervised segmentation systems (nor by the gold-standards for that matter). However, it is often desirable to distinguish between derivational and inflectional segmentations, as the former provide evidence for coarse-grained part-of-speech distinctions while the latter usually help distinguish between subcategories of one part of speech.

6.2.4 Experiments

For the iterated learning experiments I will be using a similar setup to the one presented in the previous chapter. The BMMM will start by inducing clusters from raw input (without morphological features) using the same inference settings. The induced parts of speech will then be used as input to the morphology segmentation system. I will be using the joint part-of-speech/morphology segmentation system of Sirts & Alumäe (2012), fixing the parts of speech to those obtained from the BMMM and using the default settings described in the paper. This effectively means that only the segmentation inference component of that system will be used.

It is important to state that there is no reason to assume that the iterated learning approach will produce better results than the original joint model of Sirts & Alumäe (2012); at best the iterated learning system should perform on par with the joint model—similarly to the results of the iterated vs. joint approaches to dependency induction shown in section 5.8. The main advantage of the iterated learning approach (even if it does not reach the performance of the joint system) is that it can be further extended to include dependency and alignment features—as shown in the next chapter—whereas a fully joint model of all these NLP levels will be prohibitively complex.

For an easier comparison with the results of the previous chapters I will use the nine languages of the PASCAL challenge (Gelling et al., 2012): Arabic, Basque, Czech, Danish, Dutch, English, Portuguese, Slovene and Swedish.

To examine the performance of the segmentation component of my iterated learning setup, gold-standard segmentation data are needed. Unfortunately, Morpho Chal-

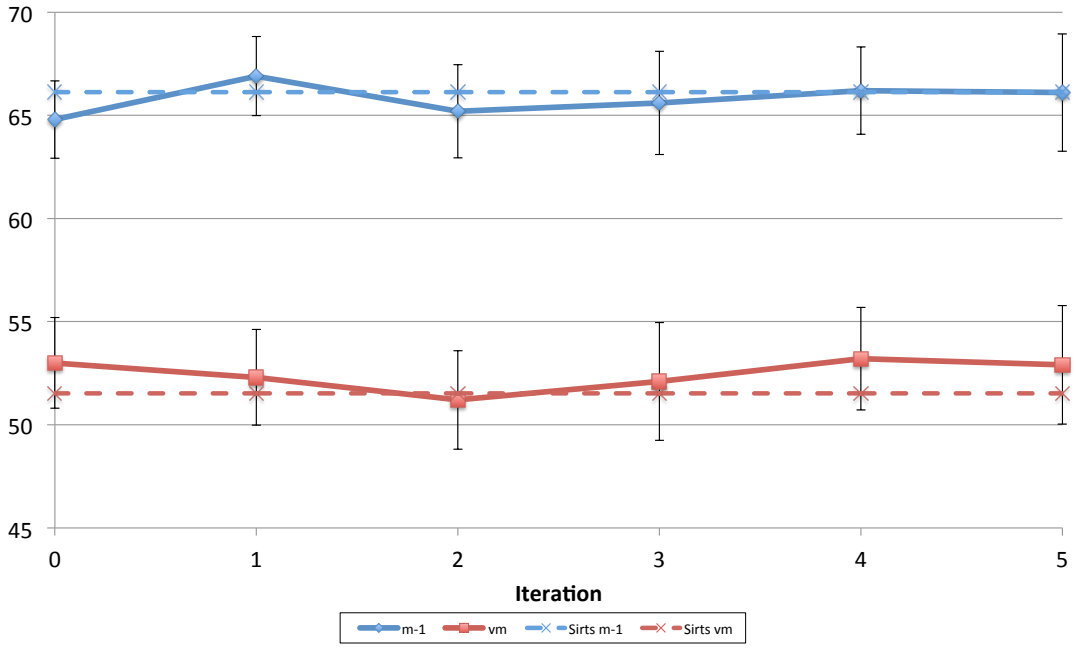


Figure 6.1: Part-of-speech induction results for the part-of-speech induction and morphology segmentation iterated learning experiments, averaged over the nine languages of the PASCAL challenge corpus. **Sirts** shows the average performance of the joint system of Sirts & Alumäe (2012) (note that the systems were tested on a slightly non-overlapping set of languages).

lenge (Kurimo et al., 2010), one of the main competitions in the area, provides data only for English, Finnish, German and Turkish. However, the CELEX morphological database (Baayen et al., 1995) from which the English gold-standard has been taken also provides morphological annotations for Dutch. I will therefore use these two languages (English and Dutch) for the evaluation of the segmentation system.

6.2.4.1 Results

Figure 6.1 presents the **m-1** and **vm** results from the iterated learning experiments, averaged over all nine languages. We can see that despite a slight initial peak and subsequent decrease, the **m-1** performance of the BMMM improves at the end of the 5 iterations to 1.3% over the baseline; the **vm** decreases in the first two iterations and then start increasing to reach the same score as the baseline in iteration 5. It is also interesting to note that the significant **m-1** increase on iteration 2 shows the inverse pattern in **vm**.

6.3 Word Alignments

As mentioned in section 4.3.2, word alignment is the task of determining the translation relationships between words, sub-word units or multi-word expressions in two or more languages. It is a vital component of statistical machine translation and although some supervised models have been proposed (e.g. Haghighi et al., 2009), the majority of available systems are unsupervised. I will now briefly discuss some influential word alignment systems, including Giza++, used in my experiments.

6.3.1 Word alignment models

The basic premise of statistical machine translation is that the best translation string is the one that maximises the following equation:

$$e_1^I = \arg\max_{e_1^I} P(e_1^I | f_1^J) \tag{6.1}$$

where $e_1^I = e_1, e_2, \ldots, e_I$ is a string in the target language e and f_1^J is the string the source language f. Equation 6.1 can be redefined using Bayes rules as:

$$e_1^I = \arg\max_{e_1^I} P(e_1^I) P(f_1^J | e_1^I)$$

where $P(e_1^I)$ is the language model which captures the grammaticality of the string and $P(e|f)$ is the translation model which can be thought of as the correspondences of alignments between the words in the two strings. Since language modelling has been addressed in other NLP areas such as speech recognition, the machine translation community mostly focuses on the alignment model.

One of the earliest and most comprehensive approaches to word alignment modelling is the IBM 1–5 models proposed by Brown et al. (1993). The first two models (IBM 1–2) represent the translation probability as a product of three independent distributions: the length distribution $p(J|I)$, the index alignment distribution $p(i|j,I)$ and the translation distribution $p(f_j|e_i)$:

$$p(f_1^J | e_1^I) = p(J|I) \prod_{j=1}^{J} \sum_{i=1}^{I} \left[p(i|j,I) p(f_j|e_i) \right] \tag{6.2}$$

Empirically, this means that to generate the source word f_j from a target word e_i we have to do the following[4]:

[4]Like the generative models of section 4.2, the generative story follows the opposite direction of the inference which is what we are interested in producing.

- choose the length of the source string J according to $p(J|I)$
- for each $j = 1,\ldots,J$ choose an alignment index $a_j = i$ according to $p(i|j,I)$
- choose a word f_j according to $p(f_1^J|e_1^I)$

The difference between the first two IBM models is that IBM1 assumes a uniform alignment distribution whereas IBM2 learns a non-uniform distribution from the data.

Vogel et al. (1996) introduced an HMM-based variation on IBM2, which captures the intuition that translated words tend to preserve their local neighbourhoods, irrespective of the distance of their positions from the source words. For example the bracketed words in the following examples are in close proximity to each other in both languages despite the change in absolute positions.

(6.4) a. Well, I think if we can make it [at eight] [on both days]
b. Ja, ich denke wenn wir das hinkriegen [an beiden Tagen] [acht Uhr]

The model uses the Markov approximation, where each aligned index is dependent only on the previous aligned index:

$$p(f_1^J|e_1^I) = \prod_{j=1}^{J}\sum_{i=1}^{I} p(i|i',I)\cdot p(f_j|e_i) \tag{6.3}$$

where $p(i|i',I)$ is the alignment probability from the previous aligned index $a_{j-1} = i'$ to the current one $a_j = i$ which in turn is dependent only on the jump width $(i-i')$ and not on the actual indices.

Getting back to the models of Brown et al. (1993), IBM3 extended the previous two models by adding two more distributions: a *fertility* distribution that allowed for multiple source words to be aligned to a single target word, and a *distortion* distribution that could model a reordering of words in the source language. Model 4 removes some of the independence assumptions of IBM3 and model 5 fixed the deficiency introduced in IBM4. Models 4 and 5 also introduce the use of word classes (discussed in section 6.3.3).

The Giza++ system of Och & Ney (2000, 2003), used in section 4.3.2 and the iterated learning experiments of the current chapter, is an extension of the GIZA module of the Egypt machine translation system (Al-Onaizan et al., 1999). Och & Ney extended Vogel et al.'s HMM-based model to include the fertility distribution of IBM3 as well as the efficiency changes of IBM4–5[5].

[5]This is why, unofficially, the HMM model of Och & Ney (2000) is considered as IBM6.

As mentioned in section 4.3.2, it is common to run Giza++ (or any other alignment model) in both directions (*source* $\rightarrow$ *target* and *target* $\rightarrow$ *source*) and then combine them deterministically by excluding those alignments that only appear on one direction. A more recent approach to word alignment is the *Dual Decomposition* model of DeNero & Macherey (2011) where they combine the two directional HMMs (using the dual decomposition method of Rush et al., 2010) to induce a state-of-the-art, joint bi-directional alignment model. This system, however, is computationally intensive and requires very large parallel corpora[6] and therefore is not suited for the experiments described here.

6.3.2 Influence of word alignments on part-of-speech induction

One of the few models to examine the influence of multilingual information on parts of speech was the model of Snyder et al. (2009), expanded by Naseem et al. (2009). Snyder et al. presented a model of part-of-speech *disambiguation* (see section 3.1) that used word alignment information. Specifically they used alignment information to draw super-lingual tags, each one corresponding to the set of aligned tags (there could be more than two languages). The distribution of super-lingual tags was then added to a monolingual non-parametric HMM-based model of part-of-speech disambiguation using a product-of-experts approach. However, Snyder et al. (2009) assumed the alignments to be fixed (induced in a pre-processing step) and did not attempt to re-estimate them using their part-of-speech tags.

A similar approach is followed by Das & Petrov (2011) who project gold-standard part-of-speech labels from English to languages without annotated data[7]. They construct graphs between the two languages where the vertices are labelled and unlabelled words and use alignment information from an unsupervised system to compute the similarity between the vertices. Similarly to Naseem et al. (2009) they use the alignments as a 'black-box' (i.e. not influenced by the part-of-speech information) but Das & Petrov keep only high confidence alignments in order to reduce the noise.

[6] To get a sense of the scale differences, the Europarl corpus (Koehn, 2005) contains $\sim$30M words, whereas the MULTEXT-East 1984 corpus contains $\sim$120k words.

[7] According to the discussion in section 3.1 this approach cannot be considered unsupervised, even though their system is general enough to be used in a completely unsupervised way.

6.3.3 Influence of Parts of Speech on Word Alignment Induction

In the HMM-based model of Och & Ney (2000) as well the IBM models 4 and 5 of Brown et al. (1993) the alignment probability of equation 6.1 is refined using word classes as a proxy for the previous target word (e_{α_j}) and current source word (f_j):

$$p\left(\alpha_j|\alpha_{j-1}, C(e_{\alpha_j}), C(f_j), I\right) \tag{6.4}$$

Och & Ney (2000) use a system called **mkcls**, which is based on the clustering algorithm of Kneser & Ney (1993), while Brown et al. (1993) use the Brown et al. (1992) clustering algorithm described in section 3.4.1. However, neither study examines the connection between the word classes and part of speech tags[8]; they always keep the number of classes fixed to 50 for both languages (French and English) and they do not investigate the use of different clustering methods or even the use of gold-standard part-of-speech tags.

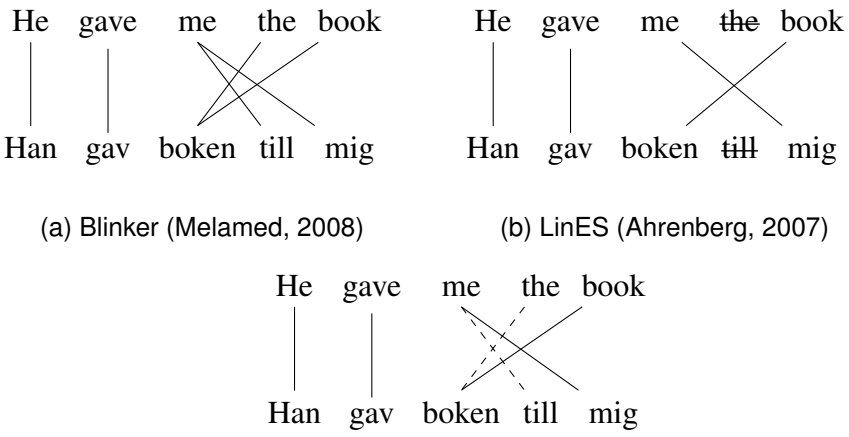
6.3.4 Evaluation

As with all the previous NLP tasks examined, evaluating unsupervised word alignment systems is quite difficult. While there are no labels or dependency direction to match against, there are still competing annotation guidelines that lead to different gold-standard data. These guidelines might differ across multiple dimensions (Holmqvist & Ahrenberg, 2011): the size of the aligned units (words/phrases), correspondence criteria (semantic/structural), treatment of untranslated items (null alignments) and confidence levels (*sure*/*possible* alignments).

Holmqvist & Ahrenberg (2011) review three different annotation guidelines: Blinker (Melamed, 2008), LinES (Ahrenberg, 2007) and the guidelines of Lambert et al. (2005) for the European Parliament Plenary Session (EPPS). Figure 6.2 presents a comparison using the English-Swedish phrase-pair *He gave me the book — Han gav boken till mig*. We can see major differences between these three guidelines: Blinker and EPPS allow for multi-word alignments in both directions (in 1-to-*many* relations); the LinES guidelines use *Null* alignments instead. Finally, EPPS allows for both *Sure* and *Possible* alignments, whereas LinES and Blinker allow only *Sure* links.

A further mark of difficulty of this task is the inter-annotator agreement, which is not as high as in the case of part-of-speech tagging. Melamed (1998) reports a average

[8]However, Brown et al. (1993, p. 280) mention that with these classes 'we can account for such facts as the appearance of adjectives before nouns in English but after them in French'.



(a) Blinker (Melamed, 2008)

(b) LinES (Ahrenberg, 2007)

(c) EPPS (Lambert et al., 2005)

Figure 6.2: Comparison between word alignment guidelines. Solid lines represent *Sure* links, dashed lines represent *Possible* links and strikethrough are *Null* links. (Source: Holmqvist & Ahrenberg, 2011)

inter-annotator agreement of 82%[9], Kruijff-Korbayová et al. (2006) 93%, and Graca et al. (2008) report an agreement of 91.6%, all well below the 98% mark of the part-of-speech tagging task (see section 2.3.2).

The most popular evaluation metrics used for word alignments are precision, recall and *alignment error rate* (**AER**), defined as follows (Och & Ney, 2003):

$$\text{precision} = \frac{|A \cap S|}{|S|}, \text{recall} = \frac{|A \cap P|}{|P|}$$

$$\text{AER} = \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

where $A\{(j, a_j | a_j > 0)\}$ corresponds to the set of proposed alignments, S to *sure* and P to *possible* gold-standard alignments. Since **AER** is an error rate, lower scores are better.

6.3.5 Experiments

For the iterated learning experiments between word alignment and part-of-speech induction I will use the BMMM model as presented in the previous chapter (with morphology features). It will replace the **mkcls** component of the Giza++ system. The rest of the parameters in Giza++ are set to their defaults[10]. Unlike the default setting

[9] Or 92% with the exclusion of function words.

[10] As can be found in the system implementation: `https://code.google.com/p/giza-pp/`.

of **mkcls** (which is 50 clusters), I will use the gold-standard number of part-of-speech tags for each language. As a proof-of-concept experiment, I will compare the performance of the **mkcls** system against the BMMM both on the part-of-speech and word alignment induction tasks.

To obtain evaluation results on both tasks, I will be using the English-French *Hansard* corpus (Germann, 2001), a portion of which (around 450 sentences) was manually annotated by Och & Ney (2000). The annotators were following guidelines similar to those of EPPS (described above) and produced *Sure* and *Possible* links but left non-aligned words without marking them as *Null*. To examine the influence of the size of the corpus, I will start by using only the gold-annotated 447 sentences (referred to as 0.5k for convenience) as training corpus and then progressively increase the size to 1k, 5k and 50k sentences.

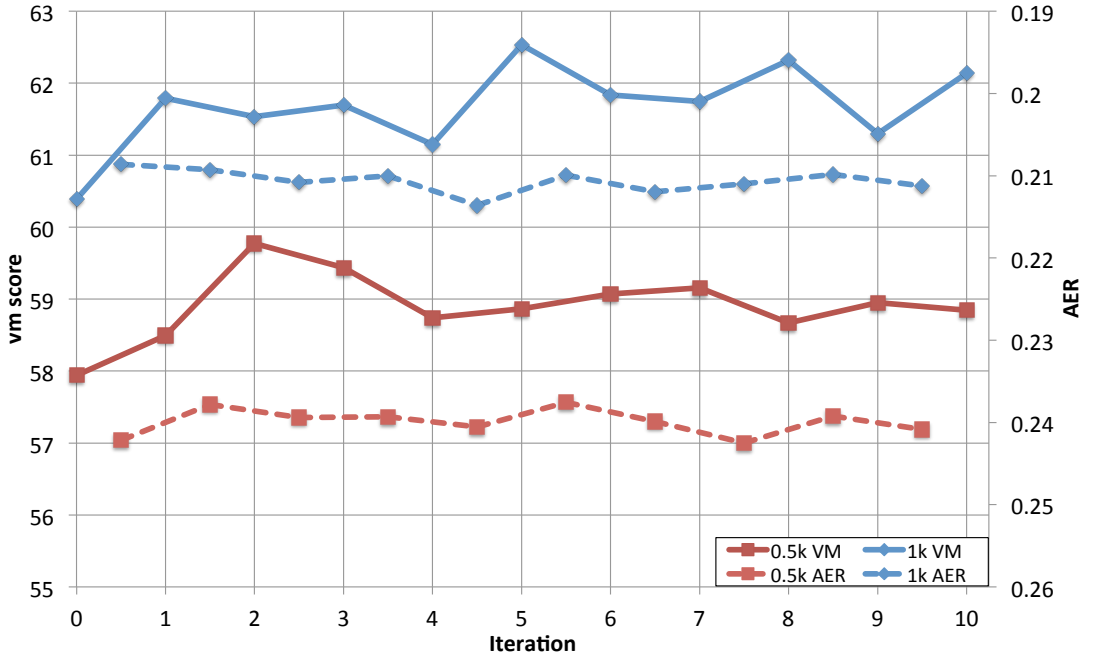
Since the *Hansard* corpus does not contain gold-standard parts of speech, I will use the Stanford Tagger (Manning, 2011) with supervised models for English and French to obtain a 'proxy' gold-standard annotation.

To examine the performance on other languages, I will use the parallel MULTEXT-East corpus (Erjavec, 2004) used previously in chapter 3; however, this corpus does not contain word-level alignments, so it is not possible to examine the performance of the Giza++ component. Since an exploration of all possible alignment pairs over five iterations requires a significant amount of time, I will only consider the alignments between English and the remaining seven languages of the corpus.

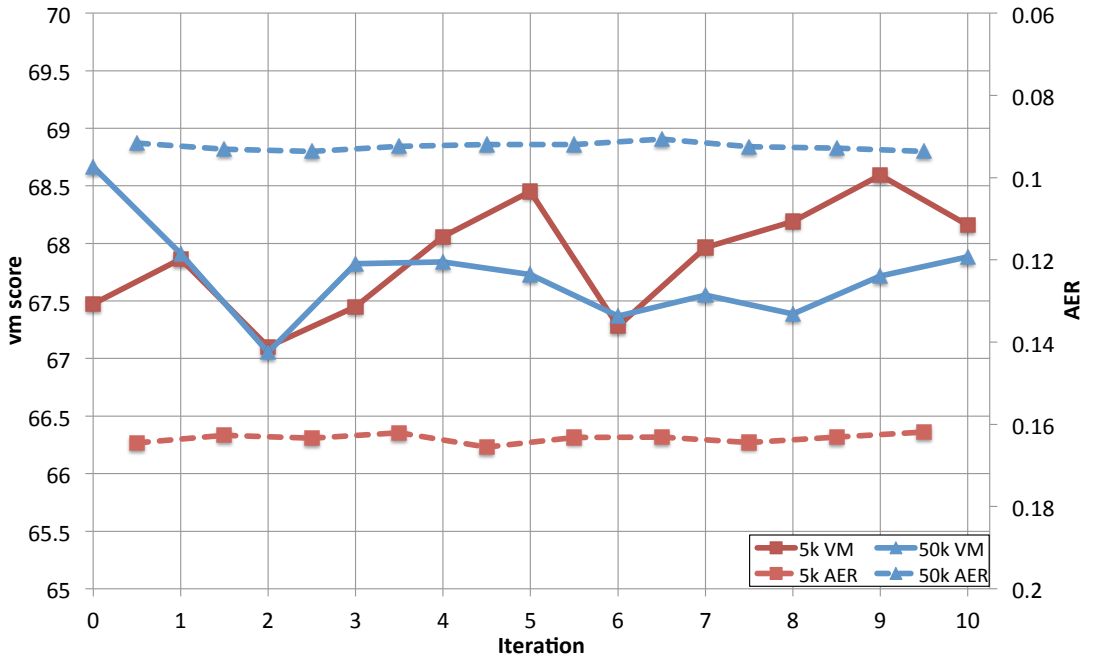
6.3.5.1 Results

Table 6.1 presents the results of the proof-of-concept experiment on the test section of the *Hansard* corpus. We can see that while **mkcls** achieves a slightly better **m-1** score, BMMM scores 1.2% higher on **vm** which results in a relative error reduction of 5.1% in the alignment model. The supervised parts of speech reduce the **AER** only by another 5.8% showing that BMMM is a good candidate to replace **mkcls**.

When the same corpus is tested in the iterated learning setting (figure 6.3), the results show that the interaction between word alignment and parts of speech is not as strong as it was for morphology or dependencies. One important factor seems to be the size of the corpus: when using only the test section (~500 sentences) or double the amount of text (figure 6.3a), the performance of the BMMM is increasing—despite small dips in some iterations—and the word alignment error rate mirrors that performance. If we add more text, **AER** shows little to no change throughout the iterations



(a) Training on 0.5k and 1k corpus



(b) Training on 5k and 50k corpus

Figure 6.3: Part-of-speech induction and bidirectional word alignment iterated learning results on the Hasard corpus. Lower **AER** is better but axis has been reversed for easy of reference.

	mkcls	BMMM	Supervised
m-1	64.8	64.4	-
vm	56.0	57.2	-
AER	0.254	0.241	0.227

Table 6.1: Part-of-speech induction and bidirectional word alignment results for the test section of the *Hansard* corpus. The part-of-speech results are calculated against the supervised labels. For **AER**, lower is better.

and **vm** results are more erratic than before. This implies that the IBM models in Giza++, while using part of speech information, rely less on it if more lexical information is present.

On the MULTEXT-East corpus (figure 6.4) we can see that performance on both **m-1** and **vm** reaches a maximum at the first iteration and then declines and stabilises to an improvement of 0.3% (**m-1**) and 0.9% (**vm**) over the baseline, but these improvements are not statistically significant (t = 1.17, p-value = .287 for **m-1** and t = 1.54, p-value = .174 for **vm**). This pattern is similar to the first five iterations of the *Hansard* corpus (figure 6.3) suggesting that the way alignments are used by BMMM might not be ideal, creating a weaker link between the two components.

Nevertheless, the initial performance peak of both systems should be enough to be used as a stepping stone for the last part of the iterated learning experiments presented in the following chapter.

6.4 Conclusion

In this chapter I have extended the iterated learning model to include two more NLP levels: morphological segmentation and word alignment. I have presented briefly the methods used by the component systems and discussed some of the difficulties in each area.

I have shown that the interaction between morphology, alignments and part-of-speech induction is beneficial for the part-of-speech induction task, and, while the performance gains are not statistically significant, they are on par with a fully joint induction system (for the morphology induction task) or better than currently used methods (in the case of word alignments). This indicates that the effect is present and,

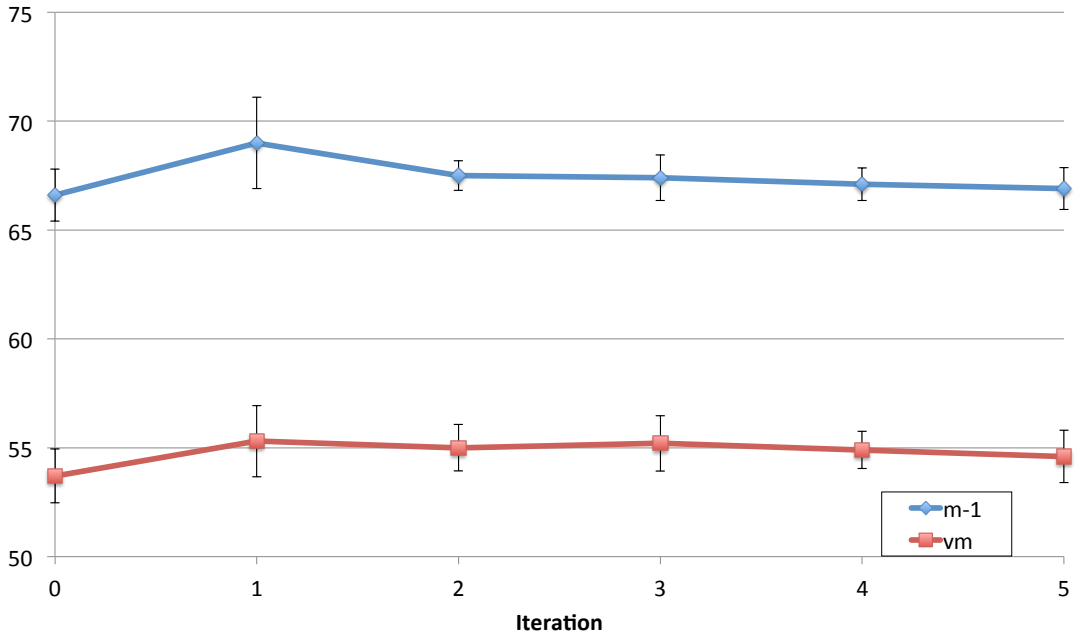


Figure 6.4: V-Measure (**vm**) and many-to-one (**m-1**) part-of-speech induction results for iterated learning experiments with word alignment averaged over the seven of the MULTEXT-East corpus (each aligned with English).

at least in the case of word alignments, beneficial for both systems.

There are a number of possible extensions to the current framework for a better interaction with both the morphology segmentation and word alignment systems. As I have already mentioned in chapter 4, the BMMM could be extended to use more morphological features (prefixes, infixes, multiple suffixes). In addition to this, the morphological segmentation component could be extended to handle more complex morphological properties, such as reduplication, stem change, infixation, etc.

Another interesting extension would be to combine the tasks of morphological segmentation and word alignment and produce morpheme alignments. This idea is theoretically appealing: since the early days of statistical machine translation, alignment was thought of as existing not necessarily between words, but between *cepts*—atomic units of meaning (Brown et al., 1993). Since morphemes are the basic semantic units, it makes sense to replace the idea of word cepts with *morpheme-cepts* and try to maximise the probability of morphological segmentation jointly with that of the probability of the alignments between the morphemes. There has been limited work in this area: Snyder & Barzilay (2008) used induced word alignments (obtained separately) to produce an unsupervised segmentation and morpheme alignment, and recently Eyigöz

et al. (2013) produced word and morpheme alignments by embedding a simple IBM Model 1 that produces morpheme alignments into an HMM that handles word alignment. The results show—similarly to the current work—that allowing for interactions between various NLP levels leads to improvements for all the systems involved.

CHAPTER 7

Cross-lingual Clusters

In whatever language, people may discover the spirit, the breath, the perfume, the traces of the original polylinguism.

Eco (1995, p.353)

7.1 Introduction

In the previous chapters I have presented evidence of the interaction between the various levels of NLP and how that interaction benefits unsupervised part-of-speech induction. In this chapter I will develop a holistic unsupervised system that takes advantage of the work described in the previous chapters.

As I have stated in the introduction of this dissertation, the ultimate goal of unsupervised systems is to be able to do linguistics 'in the wild'; that is, to be able to perform linguistic analysis from raw texts only, outside the experimental settings and corpora of the NLP literature. To that end, the aim of this chapter is produce a system that can be used as a 'black box' on unannotated (and perhaps parallel) collections of text.

I will be testing various ways of incorporating all the various NLP components used previously in coupled iterated learning experiments in a single tool that will be a proxy for a joint morphology, part-of-speech, dependency and word-alignment induction system. As part of the exploration of the concept of 'linguistics in the wild' I

will demonstrate the use of the final system in a completely unannotated corpus. I will also describe the creation of such a corpus that consists of Bible translations in 100 languages. I will be presenting my preliminary results only in the languages that I am familiar with (English, Greek). I will be producing aligned (cross-lingual) clusters and will be informally examining the similarities and differences between the members of these clusters, similarly to the typological analysis of Naseem et al. (2009, p. 35–36) and the examination of the bilingual clusters of Och (1999). Even though the work presented in this thesis does not fully explore the potential of the parallel Bible corpus, I believe that the kind of analysis presented here, as well as the existence of a massively parallel unannotated corpus, are going to be valuable starting points for future research.

7.2 Exploration of induction chains

Having established that the iterated learning model of the previous chapters can be used to combine part-of-speech induction with morphology, syntactic dependencies and word alignments individually, an obvious extension would be to create an iterated model of induction *chains*; that is, using every one of the NLP systems of the previous experiments with the BMMM acting as the mediator (like in figure 5.1). Since each of the peripheral systems can be used at any point in the chain, a decision needs to be made concerning the optimal sequence (or path). One reasonable choice—highlighted in figure 7.1—might be to follow the traditional NLP pipeline (morphology, lexicon, syntax, alignments) but starting with the distributional part-of-speech induction (iteration 0).

To test several alternative paths in a reasonable amount of time, I will focus on a single pair of languages, since the ultimate goal is to produce an aligned set of word type clusters. I will use the English-Bulgarian texts from the MULTEXT-East corpus, since they were the two languages also used for the development of the BMMM as well as being very dissimilar to each other (both in terms of script but also in terms of morphological richness). For each component system, I will be using the best configuration settings found in the previous chapters.

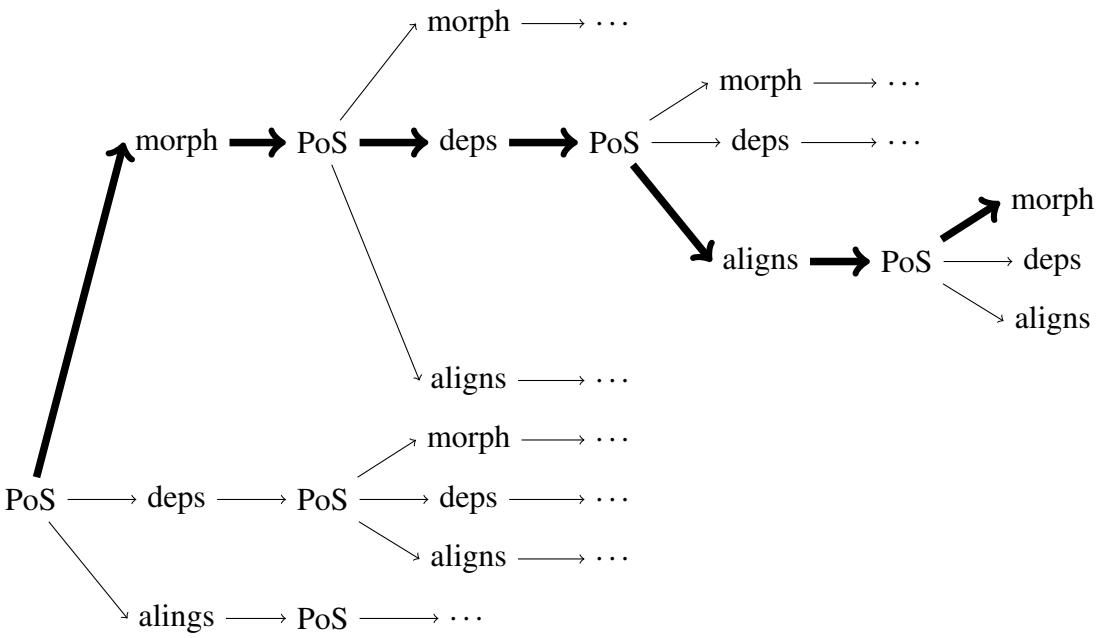


Figure 7.1: An induction chain (highlighted) through all possible induction paths.

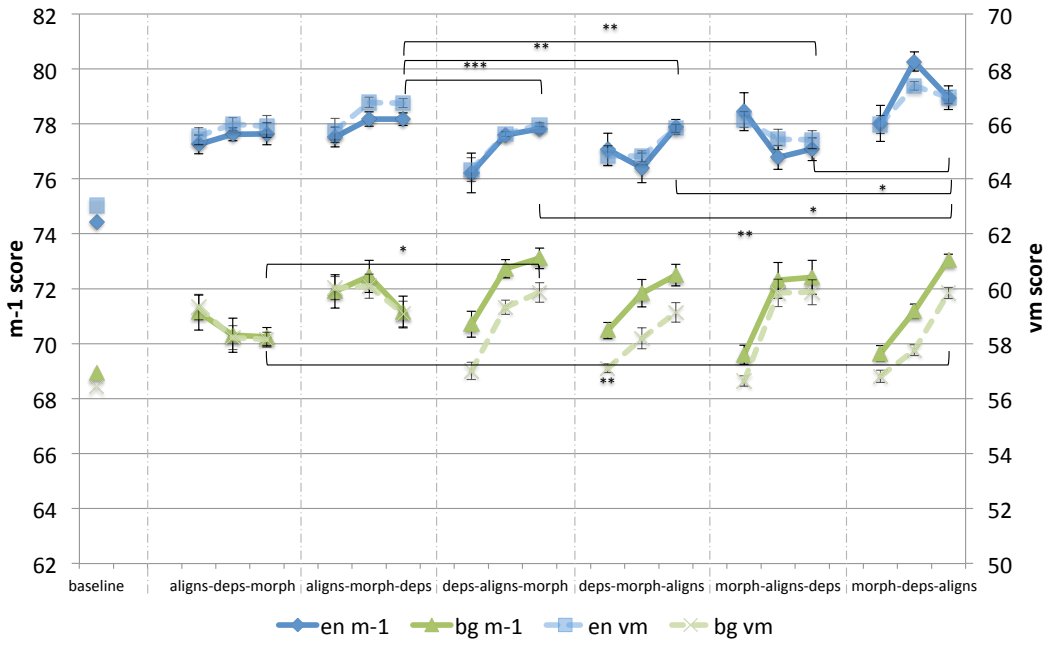


Figure 7.2: Average V-Measure (**vm**) and many-to-1 (**m-1**) scores, over 10 runs across six different induction chains for English and Bulgarian. **vm** and **m-1** are presented on different axes to allow overlapping and easier comparison. Significance values are shown only for **vm** scores.

7.2.1 Results and discussion

Figure 7.2 presents average **vm** and **m-1** results, over 10 runs of alternative induction chains of size 3. That is, each chain starts with a run of the baseline BMMM (the same across all chains), and then I progressively add the morphology, dependency and alignment systems in all possible combinations, running the BMMM between each step. This yields six possible chains: alignments, dependencies, morphology (**aligns–deps–morph**); dependency, morphology, alignments (**deps–morph–aligns**); etc.

Since the aim of this experiment is to choose a general-purpose process that will be used as a 'black box' of part of speech induction, I am interested in the performance at the end of each chain. However, the overall chain-internal trends are also worth examining.

The first result worth pointing out is that the performance at the end of each chain is significantly better than the BMMM baseline (most p-values $< .0001$, all $< .05$—not marked in figure 7.2) with an average difference of over 3% for both **m-1** and **vm** in both languages.

In English the best performing chain seems to be the sequence **morph–deps–aligns**. Despite a performance drop after the alignments stage, the final score is still better than the final score of the second best performing chain (**aligns–morph–deps**) but not significantly: the average difference in **m-1** is 0.78% ($t = 1.74$, p-value = .117) and the difference in **vm** is 0.21% ($t = 0.67$, p-value = .517).

In Bulgarian, the two best chains are **morph–deps–aligns** and **deps–aligns–morph**. Their performance is almost identical: the latter is 0.05% better in **m-1** ($t = 0.25$, p-value = .805) and 0.02% in **vm** ($t = 0.148$, p-value = .886). Unlike in English, the alignment stage in **morph–deps–aligns** increases the performance, suggesting that Bulgarian benefits from aligning with English, but not vice-versa. This result is in line with the experiments in chapter 4 where English proved to be the best candidate alignment language for the majority of the MULTEXT-East languages.

The results from both languages suggest that **morph–deps–aligns** is a good candidate for a multi-level, cross-lingual part-of-speech induction system: it performs on par with the next best chain and it has the added practical benefit that it does not require the two languages to be training in parallel until the last step. This is because, in chains where the alignment stage is not the last, each of the following BMMM stages (using alignment features) would require both languages. In **morph–deps–aligns**, each language could be trained separately and joined with the other only at the last BMMM

stage.

A final point is that **morph–deps–aligns** seems to validate the idea of the traditional NLP pipeline view: the system starts with morphology, then moves to syntax and finally considers multiple languages. This, however, might be an artifact of having only three stages in each chain. Since the iterated learning framework serves as a proxy for a fully joint system, by running each stage more than once (total chain size of six or nine etc.) the system would begin to approximate a fully joint model of morphology, dependency and word alignment induction at which point the order of operations should—in theory—be of little importance. This is an interesting topic for further exploration.

In conclusion, the experiments of this section have shown that it is possible to combine all three levels of linguistic analysis (morphology, syntax, alignments) as features in BMMM to induce better part-of-speech categories than any single level alone. By comparing all possible induction chains of size 3, I have decided to use **morph–deps–aligns** as the 'black box' for my concluding exploratory study. This will be an application of part-of-speech induction 'in the wild'; I will use the induction chain on a corpus with no gold-standard annotation to induce cross-lingual part-of-speech clusters in an attempt to demonstrated the intended use of unsupervised systems.

7.3 Using the Bible as a parallel corpus

In an attempt to access parallel material from as many and as diverse languages as possible, a highly translated text is needed. According to United Bible Societies (2013) there are at least 2,527 translations of parts of the Bible and 475 full translations. These numbers exceed by far the translations of any other work of literature. According to Wikipedia (2013) the next most translated work of literature is 'Pinocchio' with 260 languages.

There are a number of advantages to using the Bible as a corpus. Not only it has been translated into numerous languages, it has been translated into a much more diverse set of languages than any other book. This is mostly due to the efforts of 'missionary linguists' such as the Summer Institute of Linguistics (SIL, Brend & Pike, 1977) that combine anthropological and linguistic research with missionary expeditions in remote locations and as a result produce Bible translations[1].

[1]The SIL efforts are not without criticism, both linguistic (e.g. Nevins et al., 2009 on Dan Everett's studies of the Pirahã grammar) and ethical (Calvet, 1987, p. 205–17).

Another advantage of the Bible is the size of the text. The complete canonical 66 books contain on average ∼800k words which, while seemingly small compared to modern (parallel) corpora[2], is much bigger than any single work of literature[3]. A final advantage is that most of the Bible translations collected here are either public domain, or—as in the case of the King James Version—free to use for research purposes.

One of the most important issues that needs to be discussed is the 'faithfulness' of the biblical translations. Ever since the first official translations of the original biblical texts from Aramaic, Hebrew and Greek, there have been numerous discussions about the style and fidelity of translation. There are two competing translation methods: *word-for-word* (or formal equivalence), in which the literal meaning of each words as well as the syntactic structure is preserved where possible; and *sense-for-sense* translation (or dynamic equivalence), in which the 'spirit' or emotional effect of the text is kept. The former method is more appropriate for the type of analysis required here and has been put forward as the preferred method by Catholic Church (2001). However, some of the translation guides used by the 'missionary linguists' follow the latter method. For instance Nida & Taber (1969) provide a theoretical framework as well a set of principles for Bible translations. As part of their suggestions on the form of language, they advise:

> - Content is to have priority over style.
> - Contextual consistency is to have priority over verbal consistency.
> - Long, involved sentences are to be broken up on the basis of receptor-language usage.
> - Nouns expressing events should be changed to verbs whenever the results would be more in keeping with receptor-language usage.
>
> (Nida & Taber, 1969, p. 182)

This does not imply that every Bible translator has followed these principles, but given that goal of the 'missionary linguists' was to convey the message of the Bible, it makes sense that they would choose a more content-sensitive approach to their translations.

Another problem related to the style and tone of the text is the use of antiquated language. This is especially problematic in languages (mostly Western European) where Bible translations were created hundreds of years in the past. Even if modern translations exist, often the editors would choose a more archaic style of writing to match the earlier text and to give the appropriate gravity to the material. Some exceptions exist, at least in English. As Resnik et al. (1999) showed, the New International Version (NIV)

[2]For instance the British National Corpus (Leech, 1992) has ∼100M words and the Europarl corpus (Koehn, 2005) has on average ∼30M words.

[3]For instance the size of the average fiction novel is about 100k words, while 'Pinocchio' is ∼45k.

covers a significant variety of present-day terms as found in Longman Dictionary of Contemporary English (LDOCE, Proctor, 1978) and in the Brown Corpus (Francis, 1964).

For many translations, it is an open question whether the writing style of the Bible is representative of present-day language, but given the limited availability of written sources in some languages, and the breadth of available translations, the Bible corpus represents the best resource for cross-linguist analysis. Indeed there have been a number of projects that used Bible translations as either a primary or secondary source of material (Resnik et al., 1999; Yarowsky & Ngai, 2001; Kanungo et al., 2005).

The Bible has also been used as a source of universal semantic analysis. Wierzbicka (2001) has produced a semantic interpretation of parts of the New Testament, including the 'Sermon on the Mount' and the parables. This line of research falls closely in line with the present work, suggesting that there are shared underlying cross-lingual structures in the Bible. However, my investigation, while attributing semantic properties to parts of speech, remains on the syntactic side of the cross-lingual similarities spectrum.

A final issue with the use of the Bible as a parallel corpus is the fact that the alignment information is limited to verses. While it is often the case that a verse corresponds to a whole sentence, there are verses that span more than two sentences, or are limited to sub-sentence phrases. The exact number varies depending on what is considered to be sentence-final punctuation. When counting only '.' and '?', out of the ∼30,000 verses, only 4,000 contain multiple sentences. However, this number increases to 10,000 if we include ';' and more than half the verses if we add ':' as a sentence-final marker.

7.3.1 Acquiring and converting source material

Despite the great number of translations, most of the Bible texts exist only in printed or even audio form. This is expected since some of the translated languages exist only in verbal form, and even if an alphabet is introduced most speakers of that language would be illiterate. Furthermore, even when textual resources have been available for years, electronic copies are hard to obtain. In English, for instance, one of the most widespread Bibles, the King James Version, is not made available in electronic form by the official licensing body (the Scottish Bible Board). This means that there is a limited availability of machine-readable bibles available online.

Bible Database (http://bibledatabase.net/)

```
<h2>Genesis 1</h2>
<blockquote>
1:1  In die begin het God die hemel en die aarde geskape. <br>
<br>
1:2  En die aarde was woes en leeg, en duisternis was op die
wŕreldvloed, en die Gees van God het gesweef op die waters. <br>
```

Unbound Bible (http://unbound.biola.edu)

```
01O 1 1 10 In die begin het God die hemel en die aarde geskape.
01O 1 2 20 En die aarde was woes en leeg, en duisternis was op die
wŕreldvloed, en die Gees van God het gesweef op die waters.
```

GospelGo (http://gospelgo.com)

```
<p><a name="Genesis"></a>
<p><a>Genesis 1</a><p>
1 In die begin het God die hemel en die aarde geskape.
1 En die aarde was woes en leeg, en duisternis was op die
wŕreldvloed, en die Gees van God het gesweef op die waters.
```

Bible Gateway (http://www.biblegateway.com)

```
<div class='heading passage-class-0'>
<h3>Genesis 1 </h3><p class="txt-sm">Het Boek (HTB)</p></div>
<div class='passage result-text-style-normal text-html'>
<p class="verse chapter-1">
    <span id="nl-HTB-1" class="text Gen-1-1">
    <span class="chapternum">1 </span>In het begin heeft God de
    hemelen en de aarde gemaakt.</span>
</p>
<p class="verse">
    <span id="nl-HTB-2" class="text Gen-1-2">
    <sup class="versenum">2 </sup>De aarde was woest en leeg en de
    Geest van God zweefde boven de watermassa. Over de watermassa
    lag een diepe duisternis.</span>
</p>
```

Figure 7.3: Different Bible online versions of Gen:1-2 in Afrikaans (last figure is in Dutch).

```
<cesDoc ...>
<cesHeader ...>
...
<wordCount>828388</wordCount>
<byteCount units="bytes">5418715</byteCount>
...
<langUsage>
    <language iso639="afr" id="af">Afrikaans</language>
</langUsage>
</cesHeader>
<text>
<body id="Bible" lang="af">
<div id="b.GEN" type="book">
<div id="b.GEN.1" type="chapter">
    <seg id="b.GEN.1.1" type="verse">
        In die begin het God die hemel en die aarde geskape.
    </seg>
    <seg id="b.GEN.1.2" type="verse">
        En die aarde was woes en leeg, en duisternis was op die
    wlreldvloed, en die Gees van God het gesweef op die waters.
    </seg>
    ...
</div>
</div>
</body>
</text>
</cesDoc>
```

Figure 7.4: Level 1 CES annotation

There are however, a few websites that offer access to public domain, machine-readable versions of the Bible in multiple languages. The four main sources used here were the Bible Database, the Unbound Bible, GospelGo and the Bible Gateway websites. Each one offered the Bible in different formats, some containing HTML and others plain text. Figure 7.3 presents a comparison of the different versions.

In order to unify all the different styles of annotation under a well-defined universal format, I followed Resnik et al. (1999) in using the Corpus Encoding Standard (CES, Ide, 1998), conforming to the level 1 annotation guidelines. Practically, this means that each Bible was formatted as an XML file, containing nested `<div>` elements corresponding to books and chapters, and `<seg>` elements that corresponded to verses. Each of the verses was marked with a serial ID. Figure 7.4 shows the same two verses of figure 7.3 as formatted by custom scripts and hand-corrected in cases of inconsistent source formatting.

7.3.2 Corpus information

The full corpus contains 100 languages from across the world (see table 7.1 for the names of the languages). In an attempt to expand the scope of the linguistic phenomena examined, I tried to include a diverse set of languages. As table 7.2 shows, the majority of languages are non-Indo-European and 39 of the languages are spoken by fewer than 1 million speakers.

Figure 7.5 presents a geographical distribution of the languages that cover almost all the continents, and Appendix C contains detailed linguistic information about every language.

One limiting factor presented in table 7.2 is that 45 out of the 100 languages contain only partial texts. In most cases this means that only the New Testament was available for that language, but in a few cases even less text exists. This is due to the efforts of the missionary-linguists discussed in section 7.3, since the primary mission is to convert people to Christianity, their primary focus is the New Testament parts of the Bible, most importantly the gospels. This means that if we want to use all 100 languages, we are limited to the smallest amount of text contained in any of them.

One final problem was the fact that not all the canonical verses (i.e. verses that appear in the original Greek, Hebrew and Aramaic) are present even in the official translations. For instance, in the Marathi translation, the first verse of the first chapter in the Book of Ezekiel is verse no. 5, with no information about the previous four verses.

Achuar-Shiwiar	Gaelic (Scottish)†	**Polish**
Afrikaans	Galela	**Portuguese**
Aguaruna	**German**	Potawatomi†
Akawaio	**Greek**	**Q'eqchi'**
Albanian	Gujarati	Quichua
Amharic	**Haitian Creole**	Romani
Amuzgo	**Hebrew**	**Romanian**
Arabic	**Hindi**	**Russian**
Armenian†	**Hungarian**	**Serbian**
Aukan	**Icelandic**	Shuar (Jivaro)
Barasana-Eduria	**Indonesian**	**Slovak**
Basque	**Italian**	**Slovene**
Bulgarian	Jakalteko	**Somali**
Cabécar	**Japanese**	**Spanish**
Cakchiquel	K'iche'	Swahili
Campa (Ashninka)	Kabyle	**Swedish**
Camsá	**Kannada**	Syriac
Cebuano	**Korean**	Tachelhit
Chamorro†	**Latin**	**Tagalog**
Cherokee	Latvian	Tamajaq (Tuareg)†
Chinantec (Quiotepec)	**Lithuanian**	Telugu
Chinese	Lukpa	**Thai**
Coptic	**Malagasy**	**Turkish**
Croatian	**Malayalam**	Ukranian
Czech	Mam	Uma
Danish	Manx†	Uspanteco
Dinka	**Maori**	**Vietnamese**
English	**Marathi**	Wolaytta
Esperanto	**Myanmar (Burmese)**	Wolof
Estonian	Nahuatl (Tetelcingo)	**Xhosa**
Ewe	**Nepali**	**Zarma**
Farsi (Persian)	**Norwegian**	Zulu
Finnish	Ojibwa	
French	Paite (Chin)	

Table 7.1: Languages in the Bible Corpus. The languages containing the full Bible text are highlighted. Most of the remaining languages contain the New Testament part of the Bible only (languages marked with † contain smaller parts).

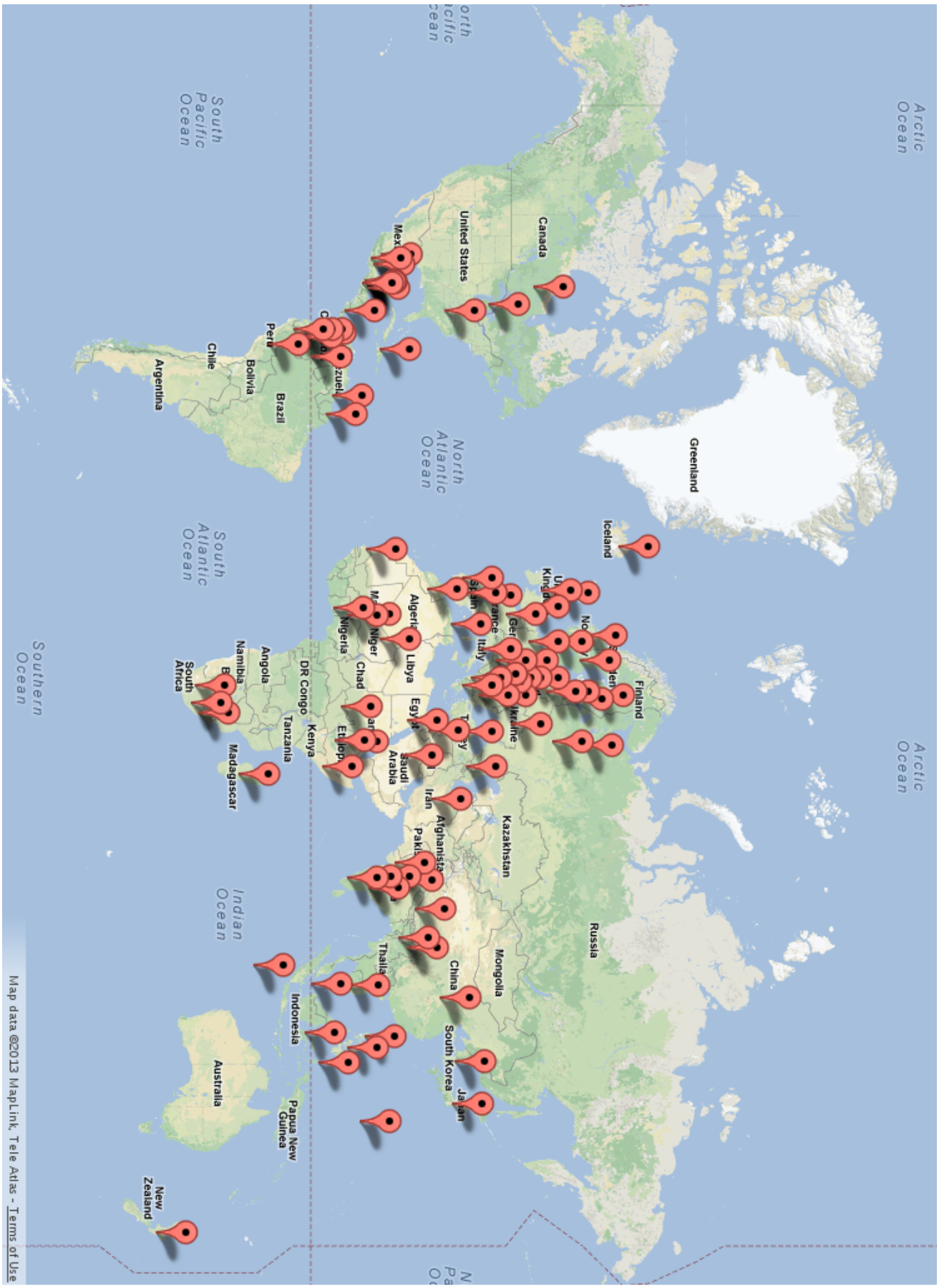


Figure 7.5: A map of the geographical distribution of languages in the Bible Corpus

Non-Latin script	28
<1M speakers	39
Non-Indo-European	66
Partial Texts	45
Total Languages	**100**

Table 7.2: Bible Corpus statistics

One possible explanation is that the missing verses are contained in the verses that come before, or after them. This is a reasonable assumption, since in some languages it might not be easy to follow the sentence structure of the original text (e.g. a sentence that is split across two verses). To account for this, I chose to ignore verses where text was missing even in one of all the languages[4]. Despite this drastic approach, the overall loss of text across languages was reasonable: on average, each bible contains about 643,000 words and after the elimination of the problematic verses the average word count was about 549,000—a 14.7% reduction.

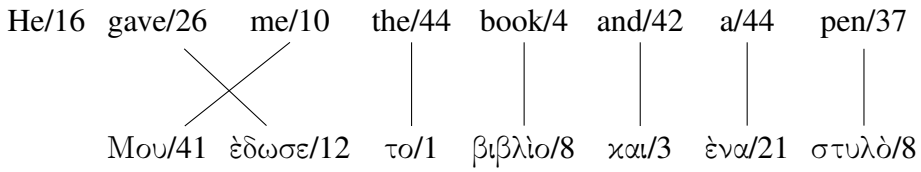
7.4 Experiments

The experimental setup is the same induction chain experiments described in section 7.2. I will use the **morph–deps–aligns** setup, starting with a run of the baseline BMMM with 45 clusters on each of the two languages I will be using for my analysis (English and Greek), and progressively add the morphology, dependencies and alignment features.

The output of the word alignment step will also be used for obtaining many-to-many cross-lingual clusters. These are generated by observing the cluster IDs of all the words that align across the two languages and collecting them into a list. For instance, in the following example clusters 4 and 37 in English will be aligned with cluster 8 in Greek; similarly English cluster 44 will be aligned with clusters 1 and 21 in Greek, capturing the ambiguity between the use of ἑνα as a determiner and a numeral.

[4]The alternative approach that I tried was to use a simple heuristic where if a verse is missing in any language, then its contents in all the other languages are merged with the previous verse. However, since there are no guarantees that the text is indeed present in the previous (or the next) verses, the quality of the alignment would be compromised.

(7.1)



However, in the real world, statistical word alignment is very noisy (i.e. there are many false alignments). If we allow all the aligned words to influence the alignment of clusters we will end up, with lots of spurious cross-lingual clusters. For this reason, I will use a cutoff threshold based on the total number of aligned words between each pair of clusters. After some empirical analysis, I used a threshold of 50% of the number of aligned words of the most aligned pair. This cutoff is applied unidirectionally from English to Greek, meaning that there might be some Greek clusters that have no alignments.

7.4.1 Results

Using the English and Greek versions of the Bible and after running all six steps of the **morph–deps–aligns** induction chain described in section 7.2 (morphology segmentation, dependency induction, word alignments and part-of-speech induction after each step), I created the set of aligned clusters seen in figure 7.6, limiting the alignments by the 50% threshold mentioned above. The first thing to notice is that there is a small amount of Greek clusters that is aligned with most of the English ones. This can be explained by the fact the distribution of words per cluster is more skewed in Greek: the average cluster size is 1090.8 words with a standard deviation of 2718.7 words whereas in English each cluster has 585 words on average and the standard deviation is 1181.4. This means that there are a few clusters that contain the majority of words and therefore the cluster alignments will be bias towards those clusters.

Figure 7.6 also presents some examples of the cross-lingual clusters that emerge. If we look at cluster 14 in English, it contains a mixture of pronouns and proper nouns[5]. It is aligned mostly with clusters 3, 27 and 29. Clusters 3 and 29 mostly contain pronouns: αυτο[=that], εκαστος[=each one], τουτον[=this (masc.)], ταυτην[=this (fem.)], παντες[=everyone], συ[=you], αυτος[=him]. Cluster 27 contains proper nouns (Θεος[=God], Ιησους[=Jesus], Μωυσης[=Moses]), but also nouns that refer to people (βασιλευς[=king], υιος[=son], ανθρωπος[=person], ιερευς[=priest]).

[5]The clustering of course is not noise free; there is no obvious reason why ‘soon’ would be in this cluster.

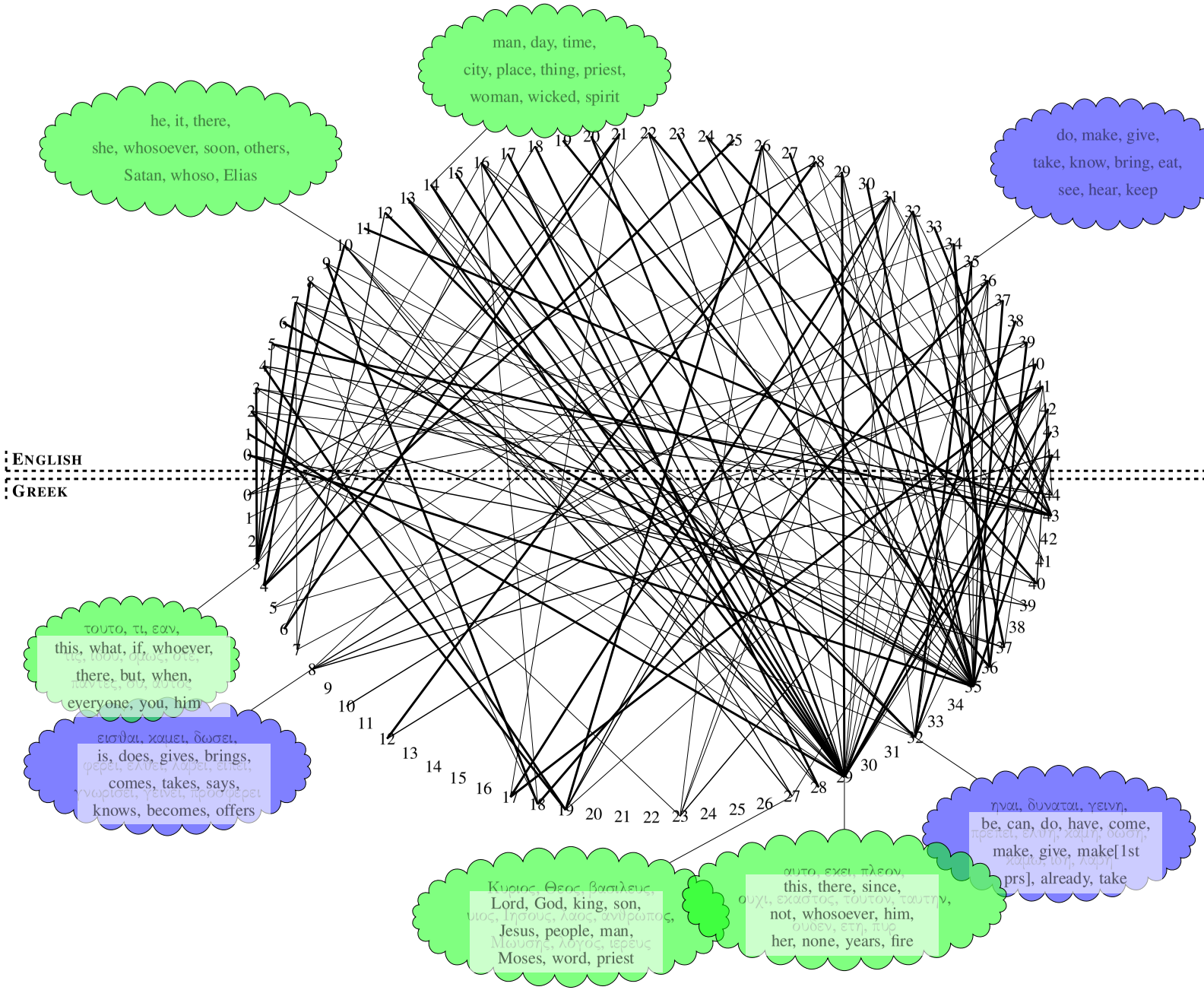


Figure 7.6: Aligned clusters between English and Greek. The thickness of the line represents the strength of the alignment (the number of aligned words between each pair of clusters). Some example cross-lingual clusters are given, with matching colours and the top-10 words for each cluster (translations for the Greek clusters are also provided).

Cluster 35 in English is mostly aligned with clusters 32 and 8 in Greek. The words contained in the English cluster are infinitives (or 1st and 2nd person present tense verbs). Interestingly, in Greek this cluster is aligned to two clusters both containing 3rd person verbs. Cluster 8 contains 3rd person singular present tense verbs (καμει[=make], δωσει[=give], φερει[=bring]) which are used in *to*-infinitive clauses (e.g. θελει να καμει=[(she) wants to make]). In this case we can make the claim that cluster 35 in English contains a 'hidden' morphological element, namely the 3rd person singular.

Another important discovery comes from looking at cluster 32 in Greek (the other aligning cluster to 35 in English). It contains 3rd person singular verbs again, but this time in the subjunctive mood. This means that cluster 35 also contains a 'hidden' semantic element of the subjunctive mood. Even though the subjunctive is rarely used overtly in English, the alignment between clusters 35 and 32, implies that it is semantically present like in the following example:

(7.2)

ο	δε	Κυριος	ας	καμη	το	αρεστον	εις	τους	οφθαλμους	αυτου
the	and	Lord	let-he	do	the	pleasing	to	the	eyes	his

'and the Lord do that which seemeth him good'

These examples demonstrate the usefulness of this system for typological analyses that can potentially uncover underlying semantic/morphosyntactic similarities between languages. By using the fully unsupervised system, we can take advantage of the lack of constraints posed by existing tagsets or linguistic theories in general, and instead discover patterns emerging from the data.

7.5 Conclusion

In this final chapter, I have brought together all the work from the previous chapters to create a tool for fully unsupervised part-of-speech induction that approaches the task holistically, encompassing the tasks of morphology segmentation, dependency induction and word alignments. I examined various induction chains where these tasks were used in different order (**aligns–deps–morph**, **deps–morph–aligns**, etc.), and selected the chain **morph–deps–aligns** as the best setting.

The ultimate goal of this multi-level system is to be used as a tool for linguistic analysis 'in the wild'. In an ideal case, one could find parallel data that would enable the creation of aligned cross-lingual clusters. These clusters would be used to examine

the differences and similarities of parts of speech across different languages. As a proof of concept, I wanted to use this system in a real-world scenario—in a corpus without gold-standard data—similar to what a linguist might encounter. To this end, I created a massively parallel corpus of Bible translations in 100 languages. I described some of the difficulties in creating this corpus and reviewed some of the properties of the Bible as a parallel corpus.

Finally, I presented my preliminary results in English and Greek. I have shown that the system is capable of discovering cross-lingual clusters that expose similarities and differences in the part-of-speech systems of these languages. The aim of this experiment was to show how a typologist might use this tool to guide them to uncover the shared 'hidden' structure of aligned clusters. This work is far from finished; one could imagine this system being extended in many different ways, including the addition of a user interface that would enable easier exploration of the aligned clusters. I hope that the proof-of-concept demonstration offered in this chapter will lead to more extensive typological work in the future.

CHAPTER 8

Conclusion

The purpose of this thesis was the development of tools to help with the discovery of patterns that traditionally correspond to parts of speech, across multiple levels of analysis (morphological, lexical, syntactic), based solely on raw text. I have offered an in-depth analysis of parts of speech both from a linguistics and an NLP perspective.

I have developed a part-of-speech induction system (BMMM), capable of incorporating multiple sources of features, and the *iterated learning* framework: a method where different NLP systems can be combined with the BMMM by training each component system on the output of the other system in each iteration. Through this iterated learning system, I have shown that taking a view of parts of speech that includes distributional, morphological, syntactic and alignment features leads to improvements in the corresponding NLP tasks (dependency parsing, morphological segmentation, word alignment and part-of-speech tagging).

The success of my approach was exemplified not only by performance improvements in traditional NLP tasks (such as part-of-speech or dependency induction), but also by providing a tool that can perform a multilevel linguistic analysis on multiple languages to induce clusters that reveal latent cross-language similarities as exemplified with the Greek and English examples in chapter 7.

While this thesis does not claim to offer a revised linguistic theory of parts of speech, it does propose a more holistic view of NLP that in turn not only provides empirical results such as the ones demonstrated here, but could also lead to contributions in theoretical linguistics.

More analytically, chapter 2 presented a review of the historical evolution of part-of-speech systems both in traditional linguistic research and as part of modern corpus-driven NLP. I presented some of the challenges posed by defining what parts of speech are, and discussed to what extent computational accounts of parts of speech align with linguistic predictions.

In chapter 3 I presented an overview of unsupervised part-of-speech induction. I discussed issues concerning evaluation of unsupervised systems in general, and examined empirically some of the most commonly used evaluation metrics. Finally, I presented a comparison of a number of unsupervised part-of-speech induction systems.

Chapter 4 presented my new part-of-speech induction system incorporating the most successful features of the systems examined in the previous chapter. The BMMM is based on the generative Bayesian framework and can be easily extended to use multiple local and non-local features such as contextual, morphological and multilingual word alignment information.

The BMMM was further extended in chapters 5 and 6 where I developed the idea of the iterated learning framework. This framework allowed me not only to use dependency relations (chapter 5), morphology segmentations and word alignments (chapter 6) as features, but also to induce them alongside parts of speech, in an iterative manner, taking advantage of the interdependency between these structures and part-of-speech tags.

Finally, in chapter 7 I combined the ideas from the previous three chapters in a proof-of-concept demonstration of chains of linguistic structure induction using a verse-aligned Bible corpus in 100 languages. I discussed the challenges in the creation of the corpus and presented some qualitative analysis of the multilingual clusters.

8.1 Future Work

As mentioned in the concluding remarks of the individual chapters, there are certainly a lot of avenues that require further exploration. One of the most interesting directions for this research is the development of fully joint unsupervised statistical models for the multiple levels of NLP. There has been limited success in joint morphology and part-of-speech models[1] but, to date, there is no model of joint part-of-speech and dependency induction, let alone a model of more than two levels at time.

[1]The limitation mainly refers to the kinds of morphological processes that these systems are able to capture.

Another potential future direction that follows directly from the discussion in chapters 2, 3 and 5 is the development of evaluation methods that are not based on gold-standard annotation. We need to find tasks with objective goals that do not rely on theory-specific annotations and develop ways to use our unsupervised systems for those tasks. For instance, an unsupervised dependency parser could be used as a (syntactic) language model for speech recognition. Using this approach, the quality of two competing parsers could be judged independently of theories of syntactic dependency or headedness. This will not only test the systems in question but also provide a testbed for competing linguistic theories: if a specific theoretical annotation can be shown to produce a better performing NLP system based on an objectively defined task (keeping all other aspects of the experiment the same) then it can be argued that the theory in question is better than its competitors.

Finally, it would be interesting to examine the induction of syntactic categories that work directly on syntax and morphology, thus avoiding the problem of three different tasks altogether. To achieve this we will have to rely on a categorical grammar formalism such as CCG (Steedman, 2001) where not only is the syntax lexicalised (i.e. each syntactic category encodes directly its syntactic function with no need for grammatical rules), but also the morphology and even the semantics are captured in the lexical 'tags'. Bisk & Hockenmaier (2012) have shown that it is possible to generate syntactic categories and a dependency structure with minimal external information. It would be interesting to see if such methods can be extended to handle morphology, semantics and other linguistic phenomena.

APPENDIX A

Tagsets of English Corpora

Table A.1: Excerpt from the Brown corpus tagset as reported by (Atwell et al., 1994)

Tag name	**Description**	**Examples**
(	opening parenthesis	*(*
)	closing parenthesis	*)*
*	negator	*not n't*
,	comma	*,*
–	dash	–
.	sentence terminator	*. ?*
:	colon	*:*
ABL	determiner/pronoun, pre-qualifier	*quite such rather*
ABN	determiner/pronoun, pre-quantifier	*all half many*
ABX	determiner/pronoun, double conjunction or pre-quantifier	*both*
AP	determiner/pronoun, post-determiner	*many other next*
AP$	determiner/pronoun, post-determiner, genitive	*other's*
AP+AP	determiner/pronoun, post-determiner, hyphenated pair	*many-much*
AT	article	*the an no*
BE	verb "to be", infinitive or imperative	*be*
BED	verb "to be", past tense, 2nd person singular or all persons plural	*were*
BED*	verb "to be", past tense, 2nd person singular or all persons plural, negated	*weren't*

Table A.1: (continued)

Tag	Description	Examples
BEDZ	verb "to be", past tense, 1st and 3rd person singular	*was*
BEDZ*	verb "to be", past tense, 1st and 3rd person singular, negated	*wasn't*
BEG	verb "to be", present participle or gerund	*being*
BEM	verb "to be", present tense, 1st person singular	*am*
BEM*	verb "to be", present tense, 1st person singular, negated	*ain't*
BEN	verb "to be", past participle	*been*
BER	verb "to be", present tense, 2nd person singular or all persons plural	*are art*
BER*	verb "to be", present tense, 2nd person singular or all persons plural, negated	*aren't ain't*
BEZ	verb "to be", present tense, 3rd person singular	*is*
BEZ*	verb "to be", present tense, 3rd person singular, negated	*isn't ain't*
CC	conjunction, coordinating	*and or but*
CD	numeral, cardinal	*two one 1*
CD$	numeral, cardinal, genitive	*1960's 1961's .404's*
CS	conjunction, subordinating	*that as after*
DO	verb "to do", uninflected present tense, infinitive or imperative	*do dost*
DO*	verb "to do", uninflected present tense or imperative, negated	*don't*
DO+PPSS	verb "to do", past or present tense + pronoun, personal, nominative, not 3rd person singular	*d'you*
DOD	verb "to do", past tense	*did done*
DOD*	verb "to do", past tense, negated	*didn't*
DOZ	verb "to do", present tense, 3rd person singular	*does*
DOZ*	verb "to do", present tense, 3rd person singular, negated	*doesn't don't*
DT	determiner/pronoun, singular	*this each another*
DT$	determiner/pronoun, singular, genitive	*another's*
DT+BEZ	determiner/pronoun + verb "to be", present tense, 3rd person singular	*that's*

Table A.2: Excerpt from the Lancaster-Oslo-Bergen (LOB) corpus tagset (Marshall, 1983)

Tag name	**Description and examples**	
&FO	formula	*10*:-1**: dE *:238**:U*
&FW	foreign word	*de Welt von*
!	exclamation mark	*!*
(	opening parenthesis	*(*
)	closing parenthesis	*)*
’	opening quotation mark	*‘ “*
’	closing quotation mark	*’ ”*
–	dash	–
,	comma	,
.	full stop	.
...	ellipsis	…
:	colon	*:*
;	semicolon	*;*
?	question mark	*?*
ABL	determiner/pronoun, pre-qualifier	*such quite rather*
ABN	determiner/pronoun, pre-quantifier	*all half*
ABX	determiner/pronoun, double conjunction or pre-quantifier	*both*
AP	determiner/pronoun, post-determiner	*more most last*
AP”	determiner/pronoun, post-determiner, ditto	*few good many*
AP$	determiner/pronoun, post-determiner, genitive	*latter’s former’s other’s*
APS	determiner/pronoun, post-determiner, plural	*others*
APS$	determiner/pronoun, post-determiner, plural, genitive	*others’*
AT	article, singular	*a an every*
ATI	article, singular or plural	*the no nae*
BE	verb “to be”, infinitive or imperitive	*be*
BED	verb “to be”, past tense, 2nd person singular or all persons plural	*were*
BEDZ	verb “to be”, past tense, 1st and 3rd person singular	*was*
BEG	verb “to be”, present participle or gerund	*being*
BEM	verb “to be”, present tense, 1st person singular	*am ’m*
BEN	verb “to be”, past participle	*been*
BER	verb “to be”, present tense, 2nd person singular or all persons plural	*are ’re art*
BEZ	verb “to be”, present tense, 3rd person singular	*is ’s iss*
CC	conjunction, coordinating	*and but or*
CC”	conjunction, coordinating, ditto	*well as*

Table A.2: (continued)

Tag	Description	
CD	numeral, cardinal	*1958 13 two*
CD$	numeral, cardinal, genitive	*8's 3's 5's*
CD-CD	numeral, cardinal, hyphenated pair	*1955-6 15-20 1861-1940*
CD1	numeral, cardinal, one	*one 1 'un*
CD1$	numeral, cardinal, one, genitive	*one's 1's*
CD1S	numeral, cardinal, one, plural	*ones 'uns*
CDS	numeral, cardinal, plural	*hundreds thousands dozens*
CS	conjunction, subordinating	*though that as*
CS"	conjunction, subordinating, ditto	*if that as*
DO	verb "to do", uninflected present tense, infinitive or imperitive	*do*
DOD	verb "to do", past tense	*did*
DOZ	verb "to do", present tense, 3rd person singular	*does doth*
DT	determiner/pronoun, singular	*another this that*
DT$	determiner/pronoun, singular, genitive	*another's*
DTI	determiner/pronoun, singular or plural	*any some enough*
DTS	determiner/pronoun, plural	*these those*
DTX	determiner, pronoun or double conjuction	*either neither*
EX	existential there	*there*
HV	verb "to have", uninflected present tense, infinitive or imperitive	*have 've hast*
HVD	verb "to have, past tense	*had 'd*
HVG	verb "to have", present participle or gerund	*having havin'*
HVN	verb "to have", past participle	*had*
HVZ	verb "to have", present tense, 3rd person singular	*has 's hath*
IN	preposition	*by from at*
IN"	preposition, ditto	*of from spite*
JJ	adjective	*large likely out-dated*
JJ"	adjective, ditto	*up off luxe*
JJB	adjective, attributive-only	*left-wing rival chief*
JJB"	adjective, attributive-only, ditto	*army called*
JJR	adjective, comparative	*higher better worse*
JJR"	adjective, comparative, ditto	*wearing*
JJT	adjective, superlative	*best fiercest bitterest*
JJT"	adjective, superlative	*selling*
JNP	adjective, word-initial capital	*African British Rhodesian*
MD	modal auxillary	*may will should*
NC	cited word	*many thanks Jimmy*

Table A.3: The SUSANNE corpus tagset as described in Sampson (1995)

Tag name	**Description and examples**
APPGf	*her* as possessive ≠ PPH01f
APPGh1	*its*
APPGh2	*their*
APPGi1	*my* as possessive
APPGi2	*our*
APPGm	*his* except as pronoun ≠ PPGm
APPGy	*your*
AT	*the* (whether as determiner or introducing the correlative construction of CGEL)
ATn	*no* as determiner or qualifier ≠ UH
AT1	indefinite article *a an*
AT1e	*every*
BTO	*in_order* introducing infinitive
CC	co-ordinating conjunction: *and and/or as_well_as plus &* solidus character ≠ plus IIm NN1c, solidus IIp YD
CCn	*nor*
CCr	*or*
CCB	*but* as co-ordinating conjunction ≠ ICSx RR
CS	subordinating conjunction (see list at end)
CSf	*for* as conjunction ≠ IF
CSg	*though* as subordinating conjunction ≠ RR
CSi	*if*
CSk	*as_if as_though*
CSn	*where* as subordinating conjunction (i.e. equivalent to "at the time at which") ≠ RRQq RRQr
CSr	*where* as subordinating conjunction (i.e. equivalent to "at the place at which") ≠ RRQq RRQr
CSA	*as* as subordinating conjunction or as preposition in comparative sense ≠ IIa RGa
CSN	*than* in all uses
CST	*that* as subordinating conjunction, including in its use in introducing relative clauses; non standard *as_how* (as in *I don't know as how I can*) ≠ *that* DD1a
CSW	*whether* in all uses
DAg	*own* as part of a genitive construction ≠ VVOv
DAr	*former latter* in all uses
DAy	*same selfsame*
DAz	*such* in all uses
DA1	*much little* ≠ *little* JJ
DA2	*many few* in all uses
DA2q	*several*
DA2R	*fewer*

Table A.3: (continued)

Tag	Description
DA2T	*fewest*
DAR	*more less* in all uses except *less* II
DAT	*most least* in all uses
DBa	*all* as determiner or pronoun ≠ NN1c, RR FB
DBh	*half* as determiner of pronoun ≠ NN1c, RR
DB2	*both* as determiner or pronoun ≠ LE RR
DD	*yon yonder* as determiner, *somesuch the_rest* ≠ *yon* RR, *yonder* RR
DDf	*enough* as pronoun or pre- or post-modifying a noun ≠ RGAf RRe
DDi	*some* as determiner or pronoun ≠ RGi
DDo	*a_lot*
DDy	*any* as determiner or pronoun ≠ RRy
DD1a	*that* as determiner, demonstrative pronoun, or qualifier (e.g. *that slowly*) ≠ CST
DD1b	*a_bit*
DD1e	*either* as determiner or pronoun ≠ LEe RR
DD1i	*this* in all uses including as qualifier (e.g. *this big*)
DD1n	*neither* as determiner or pronoun ≠ LEe RR
DD1q	*another each one_and_the_same*, as determiner or pronoun ≠ *each* RAq
DD1t	*a_little*
DD2	*a_few a_good_few a_good_many a_great_many*
DD2a	*those*
DD2i	*these*
DDQ	*what*
DDQq	*which* in interrogative uses ≠ DDQr
DDQr	*which* in relative uses ≠ DDQq
DDQGq	*whose* in interrogative uses ≠ DDQGr
DDQGr	*whose* in relative uses ≠ DDQGq
DDQV	*whichever whatever whichsoever whatsoever no_matter_which no_matter_what* ≠ *whatever* RAn, *whatsoever* RAn
EX	existential *there* ≠ RLh UH
FA	suffix (if separately wordtagged, e.g. because linked to stem by hyphen)
FB	prefix (if separately wordtagged, e.g. because linked to stem by hyphen)
FD	distorted word – used only in analysing speech
FO	indeterminate formula
FOc	formula or acronym for chemical substance, molecule, or, subatomic particle e.g. H_2SO_4 TNT DDT 14^C $C-14$ a (as in $\alpha-particle$) etc.
FOp	London postal district, British post-code, American "Zip code": *W.C.2*, *LA6 3AN*, *06520*, *0652O-1911*, etc.
FOqc	chemical equation, when analysed as a single word
FOqx	algebraic equation, when analysed as a single word

Table A.3: (continued)

Tag	Description
FOr	road name (*M6 B6480 I-95* etc.)
FOs	registration/reference/serial model number (contrast NP1z below)
FOt	telephone number (not including any exchange name spelled out in full)
FOx	algebraic expression with nominal as opposed to equative function (*alpha*, π or *pi*, dy/dx, etc.)

Table A.4: The Penn Treebank (PTB) corpus tagset (Marcus et al., 1993)

Tag name	Description	Examples
$	dollar	*$ -$ –$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$*
#	pound sign	*#*
"	straight double quote	"
‘	left open single quote	‘
“	left open double quote	“
’	right close single quote	’
”	right close double quote	”
(	opening parenthesis	*([{*
)	closing parenthesis	*)] }*
,	comma	,
.	sentence terminator	*. ! ?*
:	colon or ellipsis	*: ; ...*
CC	conjunction, coordinating	*& ’n and both but*
CD	numeral, cardinal	*mid-1890 nine-thirty forty-two*
DT	determiner	*all an another*
EX	existential there	*there*
FW	foreign word	*gemeinschaft hund ich*
IN	preposition or conjunction, subordinating	*astride among uppon*
JJ	adjective or numeral, ordinal	*third ill-mannered pre-war regrettable*
JJR	adjective, comparative	*bleaker braver breezier*
JJS	adjective, superlative	*calmest cheapest choicest*
LS	list item marker	*A A. B B. First*
MD	modal auxiliary	*can cannot could*
NN	noun, common, singular or mass	*common-carrier cabbage humour*
NNP	noun, proper, singular	*Motown Christos Shannon*
NNPS	noun, proper, plural	*Americans Americas Amharas*
NNS	noun, common, plural	*undergraduates scotches bric-a-brac*
PDT	pre-determiner	*all both half*

Table A.4: (continued)

Tag	Description	Examples
POS	genitive marker	*'s*
PRP	pronoun, personal	*hers herself him*
PRP$	pronoun, possessive	*her his mine*
RB	adverb	*occasionally unabatingly maddeningly*
RBR	adverb, comparative	*further gloomier grander*
RBS	adverb, superlative	*best biggest bluntest*
RP	particle	*aboard about across*
SYM	symbol	*% & * + ,*
TO	to as preposition or infinitive marker	*to*
UH	interjection	*Goodbye Goody Gosh*
VB	verb, base form	*ask assemble assess*
VBD	verb, past tense	*dipped pleaded swiped*
VBG	verb, present participle or gerund	*telegraphing stirring focusing*
VBN	verb, past participle	*multihulled dilapidated aerosolized*
VBP	verb, present tense, not 3rd person singular	*predominate wrap resort*
VBZ	verb, present tense, 3rd person singular	*bases reconstructs marks*
WDT	WH-determiner	*that what whatever*
WP	WH-pronoun	*that what whatever whatsoever*
WP$	WH-pronoun, possessive	*whose*
WRB	Wh-adverb	*how however whence whenever*

APPENDIX B

Part-of-Speech Review Results

B.1 Chapter 3 Results

	wsj	wsj-s
system	**vm / m-1**	**vm / m-1**
brown	63.0 / 67.8	59.6 / 66.5
clark	65.5 / 71.2	63.8 / 68.8
cw	60.6 / 71.6	55.2 / 63.7
bhmm	58.2 / 66.5	56.1 / 63.2
vbhmm	49.2 / 50.2	33.7 / 36.7
pr	54.8 / 62.5	45.0 / 53.3
feat	67.7 / 73.9	59.9 / 81.2

Table B.1: Performance of the different systems on the full WSJ and the 7k version (**wsj-s**), using **m-1** and **vm** [$|C|$:45, $|T|$:45]

system	wsj-s $\|T\|$=13 vm / m-1	wsj-s $\|T\|$=17 vm / m-1	multext-en $\|T\|$=13 vm / m-1
brown	52.5 / 80.8	55.1 / 79.2	56.9 / 81.0
clark	56.0 / 82.4	59.0 / 81.4	61.3 / 84.3
cw	47.0 / 76.9	49.4 / 75.9	53.3 / 80.5
bhmm	49.1 / 77.9	51.5 / 76.0	56.9 / 82.0
vbhmm	30.8 / 51.8	34.5 / 51.0	46.4 / 62.2
pr	39.1 / 68.3	41.9 / 67.6	47.6 / 72.5
feat	56.6 / 82.3	59.9 / 81.2	56.9 / 80.0

Table B.2: **m-1** and **vm** scores for the different systems on English MULTEXT-East (**multext-en**) and (**wsj-s**) corpora [$|C|$:45, $|T|$:{13,17}]

system	vm	m-1
brown	**68.8** (5.8)	**76.1** (8.3)
clark	68.6 (3.0)	74.5 (3.3)
bhmm	65.7 (9.5)	71.8 (8.6)
vbhmm	67.5 (**18.3**)	68.1 (**17.9**)
pr	67.2 (12.4)	71.6 (9.2)
feat	63.1 (-4.6)	69.8 (-4.1)
h&k	75.2	80.2

Table B.3: Scores on WSJ for the prototype-based part-of-speech induction system, with prototypes extracted from each of the existing systems [$|C|$:45,$|T|$:45]. Numbers in parentheses are the improvement over the same system without using the prototype step. Scores in bold indicate the best performance (improvement) in each column. **h&k** uses hand-annotated prototypes.

	corpus	**brown**	**clark**
WSJ	**wsj**	68.8 (5.8)	68.5 (3.0)
	wsj-s	62.3 (2.7)	67.5 (3.6)
MULTEXT-East	**Bulgarian**	53.7 (2.3)	50.2 (-7.1)
	Czech	49.9 (5.0)	48.0 (-4.0)
	English	58.5 (1.6)	57.9 (-3.3)
	Estonian	45.8 (4.9)	44.4 (-1.9)
	Hungarian	45.8 (0.1)	47.0 (-5.7)
	Romanian	53.2 (0.8)	52.7 (-3.3)
	Slovene	51.2 (2.9)	51.7 (-4.6)
	Serbian	48.0 (2.8)	46.4 (-4.9)

Table B.4: V-Measure scores for **brown+proto** and **clark+proto** on the MULTEXT-East and WSJ corpora. Numbers in parentheses indicate improvement over the base systems.

	Language	k-means	svd	clark	pyphmm	hcd	Tags	Types
WSJ	wsj	59.5 / 61.6	58.2 / 64.0	65.6 / 71.2	69.8 / 76.8	53.1 / 58.1	45	49,190
	wsj-s	56.7 / 60.1	54.3 / 60.7	63.8 / 68.8	-	-	45	16,850
MULTEXT-East	Bulgarian	50.3 / 59.3	41.7 / 51.0	**55.6 / 66.5**	-	-	14	16,352
	Czech	48.6 / 56.7	35.5 / 50.9	52.6 / 64.1	-	-	14	19,115
	English	56.5 / 65.4	52.3 / 65.5	60.5 / 70.6	-	-	13	9,773
	Estonian	45.3 / 55.6	38.7 / 55.3	44.4 / 58.4	-	-	13	17,845
	Hungarian	46.7 / 53.9	39.8 / 49.5	48.9 / 61.4	-	-	14	20,321
	Romanian	45.2 / 55.1	42.1 / 52.6	40.9 / 49.9	-	-	16	15,189
	Slovene	46.9 / 56.2	39.5 / 54.2	54.9 / 69.4	-	-	14	17,871
	Serbian	41.4 / 47.0	39.1 / 54.6	51.0 / 64.1	-	-	14	18,095
	average	47.6 / 56.2	41.1 / 54.2	51.1 / 63.1	-	-		
CoNLL06 Shared Task	Arabic	43.3 / 60.7	27.6 / 49.0	40.6 / 59.8	61.7 / 83.8	51.3 / 83.3	20	12,915
	Bulgarian	53.6 / 65.6	49.0 / 65.3	59.6 / 70.4	-	-	54	32,439
	Chinese	32.6 / 61.1	24.5 / 54.6	31.8 / 56.7	-	-	15	40,562
	Czech	-	-	47.1 / 65.5	52.3 / 83.1	40.2 / 72.3	12	130,208
	Danish	51.7 / 61.6	40.8 / 57.6	52.7 / 65.3	56.6 / 81.4	52.5 / 84.1	25	18,356
	Dutch	45.3 / 60.5	36.7 / 52.4	52.2 / 67.9	57.0 / 80.8	54.9 / 74.0	13	28,393
	German	58.7 / 67.5	54.1 / 64.2	63.0 / 73.9	-	-	54	72,326
	Japanese	76.1 / 76.2	74.4 / 75.5	78.6 / 77.4	-	-	80	3,231
	Portuguese	51.6 / 64.4	45.9 / 63.1	57.4 / 69.2	64.2 / 81.7	52.5 / 80.4	22	28,931
	Slovene	52.6 / 64.2	44.0 / 60.3	53.9 / 63.5	51.1 / 70.8	46.6 / 75.5	29	7,128
	Spanish	59.5 / 69.2	54.8 / 68.2	61.6 / 71.9	-	-	47	16,458
	Swedish	53.2 / 62.2	47.4 / 59.1	58.9 / 68.7	57.1 / 78.6	47.1 / 79.6	41	20,057
	Turkish	40.8 / 62.8	27.4 / 52.4	36.8 / 58.1	-	-	30	17,563
	average	51.6 / 64.7	43.9 / 60.1	53.4 / 66.8	57.1 / 80.0	49.3 / 78.5		

Table B.5: Results of systems not included in the review of section 3.4.4. The results for **pyphmm** and **hdc** are taken from the PASCAL challenge Gelling et al. (2012).

B.2 Chapter 5 Results

	BMMM		BMMM+deps	
Lang.	m-1	vm	m-1	vm
wsj	72.8	66.1	74.7 (1.9)	67.9 (1.8)
Arabic	61.5	42.4	66.4 (4.9)	44.2 (1.8)
Bulgarian	68.9	58.8	71.7 (2.8)	61.3 (2.5)
Chinese	69.4	42.6	75.8 (6.4)	45.8 (3.2)
Czech	65.7	48.4	74.8 (9.1)	57.4 (9.0)
Danish	71.1	59.0	69.9 (-1.2)	57.6 (-1.4)
Dutch	71.1	54.7	73.1 (2.0)	59.5 (4.8)
German	74.4	61.9	78.5 (4.1)	66.7 (4.8)
Japanese	78.5	77.4	81.2 (2.7)	79.5 (2.1)
Portuguese	76.8	63.9	77.8 (1.0)	64.2 (0.3)
Slovene	56.2	49.4	68.9 (12.7)	56.4 (7.0)
Spanish	71.7	63.2	76.0 (4.3)	66.2 (3.0)
Swedish	68.2	58.0	69.5 (1.3)	59.1 (1.1)
Turkish	58.7	40.2	71.3 (12.6)	45.1 (4.9)
average	68.6	55.4	73.5 (4.8)	58.7(3.3)

Table B.6: Results using gold-standard dependencies. The numbers in brackets show the difference between the performance of the baseline model (section 4.4.4) and the model using dependency features.

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
Lang.	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm
Arabic	62.6 / 38.6	61.8 / 36.7	62.6 / 36.7	61.8 / 36.7	63.5 / 37.3	63.3 / 37.2	59.1 / 37.8
Basque	61.3 / 47.9	67.6 / 52.5	66.4 / 49.0	66.6 / 46.4	67.6 / 49.7	67.1 / 46.8	- / -
Czech	63.9 / 43.7	64.8 / 43.2	65.3 / 43.1	65.4 / 43.2	66.3 / 43.8	66.3 / 43.8	69.6 / 48.1
Danish	37.1 / 38.5	43.0 / 43.4	47.7 / 45.4	46.1 / 44.5	47.9 / 46.1	47.7 / 45.6	60.7 / 41.0
Dutch	61.8 / 47.0	71.1 / 46.3	72.0 / 47.4	72.0 / 48.4	72.4 / 49.2	72.6 / 48.9	64.7 / 49.6
English	56.1 / 56.6	59.9 / 55.7	59.7 / 55.3	59.9 / 56.5	63.0 / 57.1	64.2 / 58.2	68.0 / 61.0
Portuguese	59.1 / 45.4	62.2 / 43.8	62.5 / 42.7	62.9 / 44.0	62.6 / 44.4	63.2 / 44.3	63.9 / 49.0
Slovene	39.6 / 29.6	43.4 / 31.8	53.0 / 41.4	53.3 / 41.2	53.3 / 40.9	53.1 / 41.0	66.0 / 43.0
Swedish	48.6 / 42.5	52.8 / 45.5	56.4 / 47.1	55.2 / 47.0	56.8 / 48.3	57.5 / 47.6	59.8 / 49.7

Table B.7: Iterated learning experiment results on up to 10-word sentences, for the 9 languages of the PASCAL Challenge on grammar induction (Gelling et al., 2012) using the BMMM and DMV systems. **gold** is the performance of the models using gold-standard dependencies.

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
M-1	58.3	58.5	60.6	60.4	61.5	61.1	64.0
VM	43.4	44.3	45.4	45.3	46.3	46.2	47.4
Undir	-	47.0	46.1	46.7	46.9	48.1	56.8
NED	-	60.1	58.5	59.0	60.1	60.6	68.8

Table B.8: Iterated learning experiment results on up to 10-word sentences, averaged over the 9 languages of the PASCAL Challenge using the BMMM and DMV systems.

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
Lang.	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm
Arabic	57.8 / 38.6	58.4 / 36.8	60.4 / 37.9	60.0 / 37.3	61.4 / 38.3	60.9 / 38.7	59.1 / 37.8
Basque	63.5 / 50.0	64.7 / 48.8	65.6 / 46.4	65.6 / 46.1	68.0 / 47.3	68.1 / 47.5	- / -
Czech	65.7 / 45.2	69.4 / 44.8	71.3 / 46.4	70.6 / 46.1	70.6 / 46.7	69.6 / 46.1	69.6 / 48.1
Danish	39.9 / 37.2	44.7 / 42.5	45.7 / 43.6	45.4 / 43.6	45.5 / 43.5	45.6 / 43.5	60.7 / 41.0
Dutch	66.1 / 45.2	69.3 / 47.2	70.3 / 48.3	70.3 / 49.1	71.7 / 49.8	72.1 / 49.2	64.7 / 49.6
English	54.8 / 56.0	60.7 / 58.1	62.5 / 58.3	63.4 / 57.7	64.0 / 58.5	64.2 / 58.1	68.0 / 61.0
Portuguese	61.6 / 41.4	63.5 / 43.0	62.7 / 44.3	63.9 / 44.1	64.2 / 44.3	63.5 / 44.9	63.9 / 49.0
Slovene	43.3 / 33.2	55.3 / 43.1	53.7 / 40.3	56.0 / 40.5	55.6 / 41.2	55.7 / 42.2	66.0 / 43.0
Swedish	47.2 / 41.3	53.7 / 43.6	55.2 / 44.3	53.3 / 43.9	54.5 / 45.0	55.1 / 45.1	59.8 / 49.7

Table B.9: Iterated learning experiment results on up to 10-word sentences, for the 9 languages of the PASCAL Challenge using the BMMM and *TSG*-DMV systems.

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
M-1	58.3	60.0	60.8	61.0	61.7	61.6	64.0
VM	43.4	45.3	45.5	45.4	46.1	46.1	47.4
Undir	-	49.5	48.1	47.8	47.7	47.1	70.1
NED	-	59.8	58.1	58.6	57.8	58.4	80.8

Table B.10: Iterated learning experiment results on up to 10-word sentences, averaged over the 9 languages of the PASCAL Challenge using the BMMM and *TSG*-DMV systems.

Iter.	0	1	2	3	4	5	gold
Lang.	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm
Arabic	58.0 / 38.1	56.7 / 34.7	62.3 / 40.2	54.2 / 35.3	62.2 / 40.8	60.1 / 39.3	66.3 / 43.6
Basque	69.2 / 53.1	70.5 / 52.7	70.0 / 51.8	72.1 / 52.8	70.6 / 53.1	71.8 / 52.3	- / -
Czech	73.5 / 53.6	75.3 / 54.3	75.6 / 54.8	75.6 / 54.6	75.6 / 55.2	75.8 / 54.9	71.4 / 53.5
Danish	56.9 / 59.2	59.0 / 58.9	59.4 / 59.2	61.2 / 60.0	60.3 / 59.8	60.3 / 60.2	69.7 / 56.9
Dutch	80.0 / 57.9	79.8 / 55.4	79.4 / 55.2	79.4 / 55.4	79.2 / 55.0	79.0 / 54.9	69.8 / 56.8
English	73.4 / 66.8	73.7 / 66.0	74.3 / 66.3	73.9 / 66.0	74.1 / 66.4	75.1 / 66.9	75.4 / 67.0
Portuguese	79.8 / 60.9	80.9 / 61.1	80.8 / 60.7	80.5 / 60.2	81.6 / 61.2	81.9 / 61.7	74.6 / 64.0
Slovene	64.0 / 56.0	67.5 / 53.6	66.9 / 51.6	67.3 / 51.7	67.3 / 51.6	67.7 / 51.6	64.7 / 57.0
Swedish	70.2 / 58.9	69.2 / 57.5	69.3 / 57.5	69.9 / 58.3	70.3 / 58.7	70.3 / 58.0	70.7 / 59.8

Table B.11: Iterated learning experiment results on all sentence lengths, for the 9 languages of the PASCAL Challenge using the BMMM and DMV systems, trained with 10-word sentences.

Iter.	0	1	2	3	4	5	gold
M-1	67.6	70.3	70.9	70.5	71.2	71.3	70.3
VM	55.8	54.9	55.3	54.9	55.8	55.5	57.3
Undir	-	40.8	44.6	44.7	43.7	45.6	47.1
NED	-	50.1	55.0	55.6	54.0	56.1	57.0

Table B.12: Iterated learning experiment results on all sentence lengths, averaged over the 9 languages of the PASCAL Challenge using the BMMM and DMV systems, trained with 10-word sentences.

Iter.	0	1	2	3	4	5	gold
Lang.	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm
Arabic	38.4 / 38.9	-	-	-	-	-	66.3 / 43.6
Basque	69.1 / 53.1	71.1 / 52.4	71.2 / 52.0	71.0 / 51.5	71.0 / 51.9	71.2 / 51.9	-
Czech	73.8 / 53.8	-	73.7 / 54.6	-	72.6 / 54.1	-	71.4 / 53.5
Danish	56.5 / 59.5	59.0 / 59.1	58.9 / 59.8	59.1 / 59.4	59.8 / 59.7	59.5 / 59.5	69.7 / 56.9
Dutch	77.8 / 57.2	78.6 / 55.7	78.7 / 55.1	79.0 / 55.2	79.2 / 55.1	80.5 / 55.7	69.8 / 56.8
English	72.8 / 65.7	73.7 / 65.9	-	-	-	-	75.4 / 67.0
Portuguese	78.7 / 60.9	-	78.2 / 60.9	-	78.1 / 60.4	-	74.6 / 64.0
Slovene	65.2 / 56.3	66.7 / 51.1	66.2 / 51.2	66.7 / 51.6	-	-	64.7 / 57.0
Swedish	68.2 / 58.5	70.2 / 58.4	70.4 / 58.3	69.4 / 58.0	69.0 / 57.4	69.2 / 58.0	70.7 / 59.8

Table B.13: Iterated learning experiment results on all sentence lengths, for the 9 languages of the PASCAL Challenge using the BMMM and DMV systems, trained with all sentences.

Iter.	0	1	2	3	4	5	gold
M-1	67.6	69.9	71.0	69.0	71.6	70.1	70.3
VM	55.8	57.1	56.0	55.1	56.4	56.3	57.3
Undir	-	39.4	41.3	41.0	44.5	43.8	47.1
NED	-	47.7	49.2	48.5	53.1	51.7	57.0

Table B.14: Iterated learning experiment results on all sentence lengths, averaged over the 9 languages of the PASCAL Challenge using the BMMM and DMV systems, trained with all sentences.

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
Lang.	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm	m-1 / vm
Arabic	33.6 / 34.6	38.8 / 38.9	38.9 / 38.6	38.6 / 38.3	39.2 / 38.2	38.8 / 37.4	66.3 / 43.6
Basque	69.1 / 53.1	71.2 / 51.8	71.5 / 52.0	71.8 / 52.7	72.7 / 53.0	72.7 / 53.0	- / -
Czech	73.8 / 53.8	74.5 / 54.5	73.9 / 54.4	74.5 / 55.0	74.2 / 54.6	73.9 / 54.3	71.4 / 53.5
Danish	56.5 / 59.5	58.7 / 58.1	59.2 / 58.5	59.1 / 58.0	57.9 / 58.6	57.8 / 58.7	69.7 / 56.9
Dutch	77.8 / 57.2	79.9 / 55.8	78.9 / 55.0	79.0 / 54.9	79.9 / 55.7	79.6 / 56.0	69.8 / 56.8
English	72.4 / 66.2	71.4 / 66.2	73.4 / 67.0	73.4 / 66.8	73.7 / 66.9	73.7 / 67.1	75.4 / 67.0
Portuguese	78.7 / 60.9	79.8 / 60.6	79.3 / 60.5	79.3 / 60.5	79.6 / 60.5	80.2 / 61.0	74.6 / 64.0
Slovene	65.1 / 55.8	66.7 / 51.2	67.0 / 51.9	68.6 / 52.6	68.0 / 51.6	68.9 / 52.5	64.7 / 57.0
Swedish	66.4 / 57.0	70.4 / 58.4	70.1 / 57.8	69.6 / 57.6	69.3 / 58.6	69.8 / 58.2	70.7 / 59.8

Table B.15: Iterated learning experiment results on all sentence lengths, for the 9 languages of the PASCAL Challenge using the BMMM and *TSG*-DMV systems, trained with 10-word sentences

Iter.	**0**	**1**	**2**	**3**	**4**	**5**	**gold**
M-1	67.7	67.9	68.0	68.2	68.3	68.4	70.3
VM	55.8	55.0	55.1	55.2	55.3	55.3	57.3
Undir	-	41.6	39.1	38.4	38.7	38.0	60.8
NED	-	48.8	46.8	46.1	46.4	45.7	70.6

Table B.16: Iterated learning experiment results on all sentence lengths, averaged over the 9 languages of the PASCAL Challenge using the BMMM and *TSG*-DMV systems, trained with 10-word sentences

	Baseline		IL-5		Joint		Gold	
language	m-1	vm	m-1	vm	m-1	vm	m-1	vm
Arabic	59.7	36.4	58.3	35.6	59.7	39.2	60.0	38.3
Bulgarian	66.2	43.9	73.9	47.2	75.6	47	71.2	54.8
Danish	51.6	34.8	61.8	41.7	64.4	39.7	59.1	42.3
Dutch	55.6	46.0	62.0	46.4	64.3	48	65.3	49.3
Japanese	89.3	70.9	89.5	71.6	89.0	71.5	82.0	74.1
Portuguese	59.0	47.2	62.7	45.4	63.7	43.0	63.7	49.6
Slovene	63.8	37.9	63.6	37.7	63.8	39.9	67.2	44.3
Spanish	62.2	40	66.8	41.6	63.4	42.0	62.9	46.1
Swedish	52.5	43.7	56.5	46.7	57.1	46.5	57.6	48.4
Turkish	61.1	35.5	66.8	37.3	66.1	36.2	63.1	38.5
average	62.1	43.6	66.2	45.1	66.7	45.3	65.2	48.6

Table B.17: Part-of-speech induction results on CoNLL data after 5 generations of iterated learning (IL-5) and for the joint inference. **Baseline** is the BMMM system trained on just context and morphological features (generation 0) and **Gold** is the BMMM using gold-standard dependencies.

	Baseline		IL-5		Joint		Gold	
language	Undir	NED	Undir	NED	Undir	NED	Undir	NED
Arabic	59.7	64.6	61.0	65.6	60.7	66.2	48.3	63.5
Bulgarian	56.0	64.0	60.1	66.7	60.7	67.0	46.6	56.2
Danish	55.7	65.2	65.2	71.9	59.7	66.0	64.3	70.2
Dutch	55.5	64.3	59.3	66.8	57.8	65.2	58.5	65.1
Japanese	81.2	88.4	80.2	87.8	80.9	88.3	83.8	90.4
Portuguese	61.7	69.2	61.3	67.5	61.5	67.8	62.7	68.3
Slovene	48.7	55.7	50.5	57.3	50.4	57.4	40.6	48.1
Spanish	55.8	63.5	58.3	65.1	58.3	65.1	59.1	65.9
Swedish	53.9	61.4	59.7	67.1	60.7	67.7	59.8	66.5
Turkish	71.4	76.6	69.1	73.8	68.9	73.4	56.5	58.5
average	60.0	67.3	62.5	69.0	61.9	68.4	58.0	65.3

Table B.18: Dependency induction results on CoNLL data after 5 generations of iterated learning (IL-5) and for the joint inference. **Baseline** is the DMV system trained on the baseline BMMM (generation 1) and **Gold** is the DMV trained on gold-standard parts of speech.

APPENDIX C

Bible Corpus Language Information

Table C.1: Linguistic details and available parts of the Bible corpus

ISO 639-3	Language	Family	Genus	Subgenus	Speakers	Script	Full	Parts
acu	Achuar-Shiwiar	Jivaroan			5000	Latin	N	New Testament
afr	Afrikaans	Indo-European	Germanic	West	5,000,000	Latin	Y	
agr	Aguaruna	Jivaroan			38300	Latin	N	New Testament
ake	Akawaio	Carib	Northern	East-West Guiana	4,500	Latin	N	New Testament
als	Albanian	Indo-European	Albanian	Tosk	3,000,000	Latin	Y	
amh	Amharic	Afro-Asiatic	Semitic	South	17,500,000	Ethiopic	N	New Testament
amu	Amuzgo	Oto-Manguean	Amuzgoan		23000	Latin	N	New Testament
arb	Arabic	Afro-Asiatic	Semitic	Central	206,000,000	Arabic	Y	
hye	Armenian	Indo-European	Armenian		64,00,000	Armenian	N	Gen. Exod. Gosp.
djk	Aukan	Creole	English based	Atlantic	15,500	Latin	N	New Testament
bsn	Barasana-Eduria	Tucanoan	Eastern Tucanoan	Central	1,890	Latin	N	New Testament
eus	Basque	Basque			700000	Latin	N	New Testament
bul	Bulgarian	Indo-European	Slavic	South	9,000,000	Cyrillic	Y	

Table C.1: (continued)

ISO 639-3	Language	Family	Genus	Subgenus	Speakers	Script	Full	Parts
cjp	Cabcar	Chibchan	Talamanca		8,840	Latin	N	New Testament
cak	Cakchiquel	Mayan	Quichean	Greater Quichean	132,000	Latin	N	New Testament
cni	Campa (Ashninka)	Arawakan	Maipuran	Southern Maipuran	26,100	Latin	N	New Testament
kbh	Cams	Equatorial (?)			4770	Latin	N	New Testament
ceb	Cebuano	Austronesian	Malayo-Polynesian	Phillipine	15,800,000	Latin	Y	
cha	Chamorro	Austronesian	Malayo-Polynesian	Chamorro	92,000	Latin	N	Psalm Gosp. Acts
chr	Cherokee	Iroquoian	Southern Iroquoian		16,400	Cherokee	N	New Testament
chq	Chinantec (Quiotepec)	Oto-Manguean	Chinantecan		8,000	Latin	N	New Testament
cmn	Chinese	Sino-Tibetan	Sinitic	Chinese	840,000,000	Chinese	Y	
cop	Coptic	Afro-Asiatic	Egyptian		Extinct	Coptic	N	New Testament
hrv	Croatian	Indo-European	Slavic	South	5,500,000	Latin	Y	
ces	Czech	Indo-European	Slavic	West	9,500,000	Latin	Y	
dan	Danish	Indo-European	Germanic	North	5,500,000	Latin	Y	
dik	Dinka	Nilo-Saharan	Eastern Sudanic	Nilotic	450,000	Latin	N	New Testament
eng	English	Indo-European	Germanic	West	32,800,0000	Latin	Y	
epo	Esperanto	Constructed			1000	Latin	Y	
est	Estonian	Uralic	Finno-Ugric	Finno-Permic	1,000,000	Latin	Y	
ewe	Ewe	Niger-Congo	Atlantic-Congo	Volta-Congo	2,250,000	Latin	N	New Testament
pes	Farsi (Persian)	Indo-European	Indo-Iranian	Iranian	22,000,000	Arabic	Y	
fin	Finnish	Uralic	Finno-Ugric	Finno-Permic	5,000,000	Latin	Y	
fra	French	Indo-European	Italic	Romance	58,000,000	Latin	Y	
gla	Gaelic (Scottish)	Indo-European	Celtic	Insular	67,000	Latin	N	Gospel of Mark

Table C.1: (continued)

ISO 639-3	Language	Family	Genus	Subgenus	Speakers	Script	Full	Parts
gbi	Galela	West Papuan	North Halmahera	Galela-Loloda	79,000	Latin	N	New Testament
deu	German	Indo-European	Germanic	West	90,300,000	Latin	Y	
ell	Greek	Indo-European	Greek	Attic	13,000,000	Greek	Y	
guj	Gujarati	Indo-European	Indo-Iranian	Indo-Aryan	45,500,000	Gujarati	N	New Testament
hat	Haitian Creole	Creole			7,700,000	Latin	Y	
heb	Hebrew	Afro-Asiatic	Semitic	Central	5,300,000	Hebrew	Y	
hin	Hindi	Indo-European	Indo-Iranian	Indo-Aryan	180,000,000	Devanagari	Y	
hun	Hungarian	Uralic	Finno-Ugric	Ugric	12,500,000	Latin	Y	
isl	Icelandic	Indo-European	Germanic	North	230,000	Ethiopic	Y	
ind	Indonesian	Austronesian	Malayo-Polynesian	Malayo-Sumbawan	2,3100,000	Latin	Y	
ita	Italian	Indo-European	Italic	Romance	61,700,000	Latin	Y	
jai	Jakalteko	Mayan	Kanjobalan-Chujean	Kanjobalan	77,700	Latin	N	New Testament
jpn	Japanese	Japonic			122,000,000	Kanjii	Y	
quc	K'iche'	Mayan	Quichean-Mamean	Greater Quichean	1900,000	Latin	N	New Testament
kab	Kabyle	Afro-Asiatic	Berber	Northern	3,100,000	Latin	N	New Testament
kan	Kannada	Dravidian	Southern	Tamil-Kannada	35,300,000	Kannada	Y	
kor	Korean	Altaic(?)			6,6300,000	Hangul	Y	
lat	Latin	Indo-European	Italic	Latino-Faliscan	Extinct	Latin	Y	
lav	Latvian	Indo-European	Baltic	Eastern	1,500,000	Latin	N	New Testament
lit	Lithuanian	Indo-European	Baltic	Eastern	3,100,000	Latin	Y	
dop	Lukpa	Niger-Congo	Atlantic-Congo	Volta-Congo	50,000	Latin	N	New Testament
plt	Malagasy	Austronesian	Malayo-Polynesian	Greater Barito	7,520,000	Latin	Y	

Table C.1: (continued)

ISO 639-3	Language	Family	Genus	Subgenus	Speakers	Script	Full	Parts
mal	Malayalam	Dravidian	Southern	Tamil-Kannada	35,400,000	Malayalam	Y	
mam	Mam	Mayan	Quichean-Mamean	Greater Mamean	200,000	Latin	N	New Testament
glv	Manx	Indo-European	Celtic	Insular	7,7000	Latin	N	Esth. Jonah Gosp.
mri	Maori	Austronesian	Malayo-Polynesian	Central-Eastern	60,000	Latin	Y	
mar	Marathi	Indo-European	Indo-Iranian	Indo-Aryan	68,000,000	Devanagari	Y	
mya	Myanmar (Burmese)	Sino-Tibetan	Tibeto-Burman	Lolo-Burmese	32,300,000	Myanmar	Y	
nhg	Nahuatl (Tetelcingo)	Uto-Aztecan	Southern Uto-Aztecan	Aztecan	3,500	Latin	N	New Testament
nep	Nepali	Indo-European	Indo-Iranian	Indo-Aryan	11,100,000	Devanagari	Y	
nor	Norwegian	Indo-European	Germanic	North	4,600,000	Latin	Y	
ojb	Ojibwa	Algic	Algonquian	Central	20,000	Aboriginal Syllabics	N	New Testament
pck	Paite (Chin)	Sino-Tibetan	Tibeto-Burman	Kuki-Chin-Naga	78,800	Latin	Y	
pol	Polish	Indo-European	Slavic	West	36,600,000	Latin	Y	
por	Portuguese	Indo-European	Italic	Romance	178,000,000	Latin	Y	
pot	Potawatomi	Algic	Algonquian	Central	1,300,000	Latin	N	Matthew Acts
kek	Q'eqchi'	Mayan	Quichean-Mamean	Greater Quichean	400,000	Latin	Y	
quw	Quichua	Quechuan	Quechua II	B	20,000	Latin	N	New Testament
rmn	Romani	Indo-European	Indo-Iranian	Indo-Aryan	710,000	Latin	N	New Testament
ron	Romanian	Indo-European	Italic	Romance	23,400,000	Latin	Y	
rus	Russian	Indo-European	Slavic	East	143,000,000	Cyrillic	Y	
srp	Serbian	Indo-European	Slavic	South	7,000,000	Latin	Y	
jiv	Shuar (Jivaro)	Jivaroan			46,700	Latin	N	New Testament
slk	Slovak	Indo-European	Slavic	West	4,610,000	Latin	Y	

Table C.1: (continued)

ISO 639-3	Language	Family	Genus	Subgenus	Speakers	Script	Full	Parts
slv	Slovene	Indo-European	Slavic	South	1,730,000	Latin	Y	
som	Somali	Afro-Asiatic	Cushitic	East	8,340,000	Latin	Y	
spa	Spanish	Indo-European	Italic	Romance	328,000,000	Latin	Y	
swh	Swahili	Niger-Congo	Atlantic-Congo	Volta-Congo	788,000	Latin	N	New Testament
swe	Swedish	Indo-European	Germanic	North	8,300,000	Latin	Y	
arc	Syriac	Afro-Asiatic	Semitic	Central	Extinct	Syriac	N	New Testament
shi	Tachelhit	Afro-Asiatic	Berber	Northern	3,000,000	Arabic	N	New Testament
tgl	Tagalog	Austronesian	Malayo-Polynesian	Phillipine	23,900,000	Latin	Y	
ttq	Tamajaq (Tuareg)	Afro-Asiatic	Berber	Tamasheq	640,000	Latin	N	Portions
tel	Telugu	Dravidian	South-Central	Telugu	69,600,000	Telugu	Y	
tha	Thai	Tai-Kadai	Kam-Tai	Be-Tai	20,300,000	Thai	Y	
tur	Turkish	Altaic	Turkic	Southern	50,000,000	Latin	Y	
ukr	Ukranian	Indo-European	Slavic	East	37,000,000	Cyrillic	N	New Testament
ppk	Uma	Austronesian	Malayo-Polynesian	Celebic	20,000	Latin	N	New Testament
usp	Uspanteco	Mayan	Quichean-Mamean	Greater Quichean	3,000	Latin	N	New Testament
vie	Vietnamese	Austro-Asiatic	Mon-Khmer	Viet-Muong	68,600,000	Latin	Y	
wal	Wolaytta	Afro-Asiatic	Omotic	North	1,230,000	Ethiopic	N	New Testament
wol	Wolof	Niger-Congo	Atlantic-Congo	Atlantic	4,000,000	Latin	N	New Testament
xho	Xhosa	Niger-Congo	Atlantic-Congo	Volta-Congo	7,800,000	Latin	Y	
dje	Zarma	Nilo-Saharan	Songhai	Southern	2,350,000	Latin	Y	
zul	Zulu	Niger-Congo	Atlantic-Congo	Volta-Congo	998,0000	Latin	N	New Testament

Bibliography

List of conference name abbreviations

ACL:	Annual Meeting of the Association for Computational Linguistics
CogSci:	Annual Meeting of the Cognitive Science Society
COLING:	International Conference on Computational Linguistics
CoNLL:	Conference on Computational Natural Language Learning
EACL:	Conference of the European Chapter of the Association for Computational Linguistics
EMNLP:	Conference on Empirical Methods in Natural Language Processing
HLT:	Human Language Technologies
LREC:	International Conference on Language Resources and Evaluation
IJCAI:	International Joint Conference on Artificial Intelligence
IJCNLP:	International Joint Conference on Natural Language Processing
IWPT:	International Conference on Parsing Technologies
NAACL:	Conference of the North American Chapter of the Association for Computational Linguistics
NODALIDA:	Nordic Conference on Computational Linguistics
TLT:	International Workshop on Treebanks and Linguistic Theories

Abney, S. T. (1987). *The English Noun Phrase in its Sentential Aspect*. Ph.D. thesis, MIT.

Ahrenberg, L. (2007). LinES: An English-Swedish parallel treebank. In *Proceedings of NODALIDA*.

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., & Yarowsky, D. (1999). Statistical machine translation. In *Final Report, JHU Summer Workshop*, vol. 30.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, (pp. 1152–1174).

Attias, H. (2000). A variational Bayesian framework for graphical models. *Advances in neural information processing systems*, *12*, 209–215.

Atwell, E., Hughes, J., & Souter, C. (1994). AMALGAM: Automatic mapping among lexico-grammatical annotation models. In *Proceedings of ACL workshop on The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, (pp. 11–20).

Augustine (398). Confessions. Source: James J. O'Donnell (1992), *Confessions, Vol. 1: Introduction and text*.

Auli, M., & Lopez, A. (2011). A comparison of loopy belief propagation and dual decomposition for integrated CCG supertagging and parsing. In *Proceedings of ACL-HLT*, (pp. 470–480).

Baayen, H., Piepenbrock, R., & Gulikers, L. (1995). The CELEX lexical database (release 2 ed.): Linguistic data consortium.

Baker, J. K. (1979). Trainable grammars for speech recognition. In J. J. Wolf, & D. H. Klatt (Eds.) *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, (pp. 547–550). Acoustical Society of America.

Baker, J. P. (1997). Consistency and accuracy in correcting automatically tagged data. In R. Garside, G. Leech, & T. McEnery (Eds.) *Corpus annotation : linguistic information from computer text corpora*, (pp. 243–250). London, United Kingdom: Longman.

Beal, M. J., Ghahramani, Z., & Rasmussen, C. E. (2002). The infinite hidden Markov model. *Advances in neural information processing systems*, *14*, 577–584.

Berg-Kirkpatrick, T., Burkett, D., & Klein, D. (2012). An empirical investigation of statistical significance in NLP. In *Proceedings of EMNLP-CoNLL*, (pp. 995–1005).

Berg-Kirkpatrick, T., Côté, A. B., DeNero, J., & Klein, D. (2010). Painless unsupervised learning with features. In *Proceedings of NAACL*, (pp. 582–590).

Biemann, C. (2006). Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of COLING-ACL*, (pp. 7–12).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Bisk, Y., & Hockenmaier, J. (2012). Simple robust grammar induction with combinatory categorial grammars. *Association for the Advancement of Artificial Intelligence*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Blevins, J. P. (2013). Word-based morphology from Aristotle to modern WP. In K. Allan (Ed.) *The Oxford handbook of the history of linguistics*, (pp. 375–396). Oxford, United Kingdom: Oxford University Press.

Blunsom, P., & Cohn, T. (2010). Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of EMNLP*, (pp. 1204–1213).

Blunsom, P., & Cohn, T. (2011). A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. In *Proceedings of ACL-HLT*, (pp. 865–874).

Böhmová, A., Hajič, J., Hajičová, E., & Hladká, B. (2001). The Prague dependency treebank: Three-level annotation scenario. In A. Abeillé (Ed.) *Treebanks: Building and Using Syntactically Annotated Corpora*, (pp. 103 – 126). Kluwer Academic Publishers.

Boonkwan, P., & Steedman, M. (2011). Grammar induction from text using small syntactic prototypes. In *Proceedings of IJCNLP*, (pp. 438–446).

Box, G. E., & Muller, M. E. (1958). A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, *29*, 610–611.

Brants, S., Dipper, S., Hansen, S., Lezius, W., & Smith, G. (2002). The TIGER treebank. In *Proceedings of TLT*, (pp. 24–41).

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, (pp. 224–231).

Brend, R. M., & Pike, K. L. (1977). *The Summer Institute of Linguistics: its works and contributions*. The Hague, Netherlands: Mouton.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, (pp. 112–116).

Brown, P. F., Della Pietra, V. J., Della Pietra, S. A., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*, 263–311.

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

Brown, R. (1958). *Words and things*. New York, NY, USA: Free Press.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*, (pp. 149–164).

Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, *2*, 159–225.

Calvet, L.-J. (1987). *La guerre des langues et les politiques linguistiques*. Paris: Payot.

Carlberger, J., & Kann, V. (1999). Implementing an efficient part-of-speech tagger. *Software-Practice and Experience*, *29*(9), 815–32.

Catholic Church (2001). *Liturgiam authenticam: fifth instruction on vernacular translation of the Roman liturgy*. Washington, D.C., USA: United States Conference of Catholic Bishops.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.) (2006). *Semi-supervised learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: The MIT Press.

Charniak, E. (1991). Bayesian networks without tears. *AI magazine*, *12*, 50–63.

Charniak, E. (1997). Statistical techniques for natural language parsing. *AI magazine*, *18*, 33–43.

Chen, K.-J., Luo, C.-C., Chang, M.-C., Chen, F.-Y., Chen, C.-J., Huang, C.-R., & Gao, Z.-M. (2003). Sinica treebank: Design criteria, representational issues and implementation. In A. Abeillé (Ed.) *Treebanks: Building and using syntactically annotated corpora*, (pp. 231–248). Kluwer Academic Publishers.

Chomsky, N. (1957). *Syntactic structures*. The Hague, Netherlands: Mouton.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA, USA: MIT Press.

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2010). Two decades of unsupervised PoS induction: How far have we come? In *Proceedings of EMNLP*, (pp. 575–584).

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2011). A Bayesian mixture model for PoS induction using multiple features. In *Proceedings of EMNLP*, (pp. 638–647).

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2012). Turning the pipeline into a loop: Iterated unsupervised dependency parsing and PoS induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, (pp. 96–99).

Chrupała, G. (2011). Efficient induction of probabilistic word classes with LDA. In *Proceedings of IJCNLP*, (pp. 363–372).

Chrupała, G. (2012). Hierarchical clustering of word class distributions. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, (pp. 100–104).

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*, (pp. 136–143).

Cinque, G. (1999). *Adverbs and functional heads: A cross-linguistic perspective*. Oxford, United Kingdom: Oxford University Press.

Civit, M., Martí, M., & Bufí, N. (2006). Cat3LB and Cast3LB: From constituents to dependencies. In T. Salakoski, F. Ginter, S. Pyysalo, & T. Pahikkala (Eds.) *Advances in natural language processing*, vol. 4139 of *Lecture Notes in Computer Science*, (pp. 141–152). Berlin, Germany: Springer Berlin / Heidelberg.

Clark, A. (2000). Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL*, (pp. 91–94).

Clark, A. (2003). Combining distributional and morphological information for part of speech induction. In *Proceedings of EACL*, (pp. 59–66).

Clark, A. (2010). Efficient, correct, unsupervised learning of context-sensitive languages. In *Proceedings of CoNLL*, (pp. 28–37).

Clark, A., & Lappin, S. (2010). Unsupervised learning and grammar induction. In A. Clark, C. Fox, & S. Lappin (Eds.) *The handbook of computational linguistics and natural language processing*, Blackwell Handbooks in Linguistics. Hoboken, NJ, USA: John Wiley & Sons.

Cohen, J. (1994). The earth is round ($p<.05$). *American Psychologist*, *49*(12), 997–1003.

Cohen, S. B., Das, D., & Smith, N. A. (2011). Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*, (pp. 50–61).

Cohen, S. B., & Smith, N. A. (2009). Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL-HLT*, (pp. 74–82).

Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Creutz, M., & Lagus, K. (2005). Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, (pp. 106–113).

Creutz, M., & Lagus, K. (2006). Morfessor in the Morpho challenge. In *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, (pp. 12–17).

Croft, W. (1991). *Syntactic categories and grammatical relations: The cognitive organization of information*. Chicago, IL, USA: University of Chicago Press.

Croft, W. (2000). Parts of speech as language universals and as language-particular categories. In S. Vogel, P. M., & B. Comrie (Eds.) *Approaches to the typology of word classes*, (pp. 65–102). Berlin, Germany: Mouton de Gruyter.

Daelemans, W., Zavrel, J., Berck, P., & Gillis, S. (1996). MBT: A memory-based part of speech tagger generator. In *Proceedings of the Fourth Workshop on Very Large Corpora*, (pp. 14–27).

Das, D., & Petrov, S. (2011). Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL-HLT*, (pp. 600–609).

Déjean, H. (2000). How to evaluate and compare tagsets? a proposal. In *Proceedings of LREC*.

Demberg, V. (2007). A Language-Independent unsupervised model for morphological segmentation. In *Proceedings of ACL*, (pp. 920–927).

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

DeNero, J., & Macherey, K. (2011). Model-based aligner combination using dual decomposition. In *Proceedings of ACL*, (pp. 420–429).

Dixon, R. M. W. (1977). Where have all the adjectives gone? *Studies in Language*, *1*, 19–80.

Dubbin, G., & Blunsom, P. (2012). Unsupervised bayesian part of speech inference with particle gibbs. In P. Flach, T. Bie, & N. Cristianini (Eds.) *Machine Learning and Knowledge Discovery in Databases*, vol. 7523 of *Lecture Notes in Computer Science*, (pp. 760–773). Springer Berlin Heidelberg.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64.

Džeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Žabokrtsky, Z., & Žele, A. (2006). Towards a Slovene dependency treebank. In *Proceedings of LREC*, (pp. 1388–1391).

Eco, U. (1995). *The search for the perfect language*. Cambridge, MA, USA: Blackwell Publishers.

Eisner, J. (2000). Bilexical grammars and their cubic-time parsing algorithms. *Advances in Probabilistic and Other Parsing Technologies*, *16*, 29–61.

Erjavec, T. (2004). MULTEXT-East version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of LREC*, (pp. 1535–1538). Paris, France.

Eryiğit, G., & Adalı, E. (2004). An affix stripping morphological analyzer for Turkish. In *Proceedings of the International Conference Artificial Intelligence and Applications*, (pp. 299–304).

Eyigöz, E., Gildea, D., & Oflazer, K. (2013). Simultaneous word-morpheme alignment for statistical machine translation. In *Proceedings NAACL*.

Finkel, J. R. (2010). *Holistic Language Processing: Joint Models of Linguistic Structure*. Ph.D. thesis, Stanford University.

Fisher, R. A. (1925). *Statistical methods for research workers*. Edinburgh, United Kingdom: Oliver and Boyd.

Francis, W. N. (1964). A standard sample of present-day English for use with digital computers. Tech. rep., Dept. of Linguistics, Brown University, Providence, RI, USA. Report to the US Office of Education on Co-operative Research Project no. E-007.

Francis, W. N., & Kučera, H. (1964). Manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. Tech. rep., Dept. of Linguistics, Brown University, Providence, RI, USA.

Frank, S., Goldwater, S., & Keller, F. (2009). Evaluating models of syntactic category acquisition without using a gold standard. In *Proceedings of CogSci*, (pp. 2576–2581).

Ganchev, K., Graça, J. a., Gillenwater, J., & Taskar, B. (2009). Posterior regularization for structured latent variable models. Tech. rep., University of Pennsylvania.

Ganchev, K., Graça, J. a. V., & Taskar, B. (2008). Better alignments = better translations? In *Proceedings of ACL-HLT*, (pp. 986–993).

Gao, J., & Johnson, M. (2008). A comparison of Bayesian estimators for unsupervised hidden Markov model POS taggers. In *Proceedings of EMNLP*, (pp. 344–352).

Garrette, D., & Baldridge, J. (2013). Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of NAACL*, (pp. 138–147).

Garrette, D., Mielens, J., & Baldridge, J. (2013). Real-world semi-supervised learning of POS-taggers for low-resource languages. In *Proceedings of ACL*, (pp. 583–592).

Garside, R., Leech, G., & McEnery, T. (Eds.) (1997). *Corpus annotation: Linguistic information from computer text corpora*. London, United Kingdom: Longman.

Garside, R., Leech, G., & Sampson, G. (Eds.) (1987). *The computational analysis of English – A corpus-based approach*. London, United Kingdom: Longman.

Geertzen, J., & van Zaanen, M. (2004). Grammatical inference using suffix trees. *Grammatical Inference: Algorithms and Applications*, (pp. 163–174).

Gelling, D., Cohn, T., Blunsom, P., & Graça, J. (2012). The PASCAL challenge on grammar induction. In *The NAACL-HLT Workshop on the Induction of Linguistic Structure*, (pp. 64–80).

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (pp. 721–741).

Germann, U. (2001). Aligned Hansards of the 36th parliament of Canada. *Natural Language Group of the USC Information Sciences Institute*.

Gigerenzer, G. (2004). Mindless statistics. *The Journal of Socio-Economics*, *33*(5), 587–606.

Gillenwater, J., Ganchev, K., Graça, J. a., Pereira, F., & Taskar, B. (2010). Sparsity in dependency grammar induction. In *Proceedings of ACL*, (pp. 194–199).

Gillick, L., & Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, (pp. 532–535).

Gimpel, K., & Smith, N. A. (2012). Concavity and initialization for unsupervised dependency parsing. In *Proceedings of NAACL-HLT*, (pp. 577–581).

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

Goldsmith, J. A. (2010). Segmentation and morphology. In A. Clark, C. Fox, & S. Lappin (Eds.) *The handbook of computational linguistics and natural language orocessing*. Chichester, United Kingdom: Wiley-Blackwell.

Goldwater, S. (2006). *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Goldwater, S., & Griffiths, T. (2007). A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of ACL*, (pp. 744–751).

Goldwater, S., Griffiths, T., & Johnson, M. (2006a). Interpolating between types and tokens by estimating power-law generators. *Advances in neural information processing systems*, *18*, 459–466.

Goldwater, S., Griffiths, T. L., & Johnson, M. (2006b). Contextual dependencies in unsupervised word segmentation. In *Proceedings CoLing-ACL*.

Goodman, N. (1983). *Fact, fiction, and forecast*. Cambridge, MA, USA: Harvard University Press.

Graca, J. a., Pardal, J. P., Coheur, L., & Caseiro, D. (2008). Building a golden collection of parallel multi-language word alignments. In *Proceedings of LREC*, (pp. 986–993).

Graça, J. a., Ganchev, K., Taskar, B., & Pereira, F. (2009). Posterior vs parameter sparsity in latent variable models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta (Eds.) *Advances in neural information processing systems 22*, (pp. 664–672). NIPS Foundation.

Graves, F. P. (1912). *Peter Ramus and the educational reformation of the sixteenth century*. New York, NY, USA: Macmillan.

Greenbaum, S. (1993). The tagset for the international corpus of English. In C. Souter, & E. Atwell (Eds.) *Corpus-based computational linguistics*. Amsterdam, Netherlands: Rodopi.

Greene, B. B., & Rubin, G. M. (1971). Automatic grammatical tagging of english. Tech. rep., Brown University, Providence, RI, USA.

Haghighi, A., Blitzer, J., DeNero, J., & Klein, D. (2009). Better word alignments with supervised ITG models. In *Proceedings of ACL-IJCNLP*, (pp. 923–931).

Haghighi, A., & Klein, D. (2006). Prototype-driven learning for sequence models. In *Proceedings of NAACL*, (pp. 320–327).

Hajičová, E. (2002). Theoretical description of language as a basis of corpus annotation: The case of Prague Dependency Treebank. *Prague Linguistic Circle Papers*, *4*, 111–127.

Hajič, J., Panevová, J., Urešová, Z., Bémová, A., & Pajas, P. (2003). PDTVALLEX: creating a large-coverage valency lexicon for treebank annotation. In *Proceedings of TLT*, (pp. 57–68).

Hajič, J., Vidová-Hladká, B., & Pajas, P. (2001). The Prague Dependency Treebank: Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, (pp. 105–114).

Hana, J., & Feldman, A. (2012). Resource-light approaches to computational morphology part 1: Monolingual approaches. *Language and Linguistics Compass*, *6*, 622–634.

Harris, Z. S. (1946). From morpheme to utterance. *Language*, *22*, pp. 161–183.

Harris, Z. S. (1951). *Methods in structural linguistics*. Chicago, IL, USA: University of Chicago Press.

Harris, Z. S. (1954). Distributional structure. *Word*, (pp. 146–162).

Haspelmath, M. (2001). Word classes and parts of speech. In P. B. Baltes, & N. J. Smelser (Eds.) *International encyclopedia of the aocial and behavioral sciences*, (pp. 16,538–16,545). Oxford, United Kingdom: Elsevier Science Ltd. / Pergamon.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Headden, W. P., Johnson, M., & McClosky, D. (2009). Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of NAACL-HLT*, (pp. 101–109).

Headden, W. P., McClosky, D., & Charniak, E. (2008). Evaluating unsupervised part-of-speech tagging for grammar induction. In *Proceedings of ACL*, (pp. 329–336).

Henkel, R. E. (1976). *Tests of significance*, vol. 4 of *Quantitative applications in the social sciences*. Beverly Hills, CA, USA and London, United Kingdom: Sage Publications.

Hockett, C. F. (1958). *A course in modern linguistics*. New York, NY, USA: Macmillan.

Holmqvist, M., & Ahrenberg, L. (2011). A gold standard for English–Swedish word alignment. In *Proceedings of NODALIDA*, (pp. 106–113).

Hopper, P. J., & Thompson, S. A. (1984). The discourse basis for lexical categories in universal grammar. *Language*, (pp. 703–752).

Huang, Z., Harper, M., & Petrov, S. (2010). Self-training with products of latent variable grammars. In *Proceedings of EMNLP*, (pp. 12–22).

Ide, N. (1998). Encoding linguistic corpora. In *Proceedings of the sixth Workshop on Very Large Corpora*, (pp. 9–17).

Jackendoff, R. (1977). *X-Bar syntax: A study of phrase structure*. Cambridge, MA, USA: MIT Press.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA*, (pp. 105–112). Tartu, Estonia.

Johnson, M. (2007). Why doesn't EM find good HMM POS-taggers? In *Proceedings of EMNLP-CoNLL*, (pp. 296–305).

Johnson, M., Griffiths, T., & Goldwater, S. (2007). Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of NAACL-HLT*, (pp. 139–146).

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*, *10*, 136–163.

Kaalep, H.-J. (1997). An Estonian morphological analyser and the impact of a corpus on its development. *Computers and the Humanities*, *31*, 115–133.

Kanji, G. K. (2006). *100 statistical tests*. London, United Kingdom: SAGE Publications.

Kanungo, T., Resnik, P., Mao, S., Kim, D., & Zheng, Q. (2005). The Bible and multilingual optical character recognition. *Communications of the ACM*, *48*, 124–130.

Kawata, Y., & Bartels, J. (2000). Stylebook for the Japanese treebank in VERMOBIL. Tech. rep., Universität Tüubingen.
URL `http://www.sfs.uni-tuebingen.de/resources/stylebook_vm_jap.pdf`

Kemp, J. A. (1986). The tekhne grammatike of Dionysius Thrax: Translated into English. *Historiographia Linguistica*, *13*, 343–363.

Klein, D. (2005). *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Klein, D., & Manning, C. D. (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of ACL*, (pp. 128–135).

Klein, D., & Manning, C. D. (2004). Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of ACL*.

Kline, R. B. (2013). *Beyond significance testing: statistics reform in the behavioral sciences*. Washington, DC, USA: American Psychological Association.

Kneser, R., & Ney, H. (1993). Forming word classes by statistical clustering for statistical language modelling. In R. Köhler, & B. B. Rieger (Eds.) *Contributions to quantitative linguistics: Proceedings of quantitative linguistics conference*, (pp. 221–226). Dordrecht, Netherlands: Springer.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, vol. 4, (pp. 388–395).

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, vol. 5.

Koo, T., Carreras, X., & Collins, M. (2008). Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL: HLT*, (pp. 595–603). Columbus, Ohio: Association for Computational Linguistics.

Kruijff-Korbayová, I., Chvátalová, K., & Postolache, O. (2006). Annotation guidelines for Czech–English word alignment. In *Proceedings of LREC*, (pp. 1256–1261).

Kurimo, M., Creutz, M., & Varjokallio, M. (2006). Unsupervised segmentation of words into morphemes Morpho Challenge 2005: Application to automatic speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, (pp. 1–4).

Kurimo, M., Virpioja, S., Turunen, V., & Lagus, K. (2010). Morpho challenge competition 2005–2010: evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, (pp. 87–95).

Kučera, H. (1992). The odd couple: The linguist and the software engineer. In J. Svartvik (Ed.) *Directions in corpus linguistics: Proceedings of Nobel symposium 82, Stokholm, 4-8 August 1991*. Berlin, Germany: De Gruyter.

Lamar, M., Maron, Y., Johnson, M., & Bienenstock, E. (2010). SVD and clustering for unsupervised POS tagging. In *Proceedings of ACL*, (pp. 215–219).

Lambert, P., De Gispert, A., Banchs, R., & Mariño, J. B. (2005). Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, *39*, 267–285.

Lancelot, C., & Arnauld, A. (1975). *The Port-Royal grammar: general and rational grammar*. The Hague, Netherlands: Mouton.

Langacker, R. W. (1987). *Foundations of cognitive grammar, Volume I, Theoretical prerequisites*. Stanford, CA, USA: Stanford University Press.

Lee, Y. K., Haghighi, A., & Barzilay, R. (2010). Simple type-level unsupervised POS tagging. In *Proceedings of EMNLP*, (pp. 853–861).

Lee, Y. K., Haghighi, A., & Barzilay, R. (2011). Modeling syntactic context improves morphological segmentation. In *Proceedings of CoNLL*, (pp. 1–9).

Leech, G. (1992). 100 million words of English: the British national corpus. *Language Research*, *28*, 1–13.

Leech, G. (1997). Grammatical tagging. In R. Garside, G. Leech, & T. McEnery (Eds.) *Corpus annotation : linguistic information from computer text corpora*. London, United Kingdom: Longman.

Li, S., Graa, J., & Taskar, B. (2012). Wiki-ly supervised part-of-speech tagging. In *Proceedings of EMNLP-CoNLL*, (pp. 1389–1398).

Li, Z., Zhang, M., Che, W., Liu, T., Chen, W., & Li, H. (2011). Joint models for chinese POS tagging and dependency parsing. In *Proceedings of EMNLP*, (pp. 1180–1191).

Lin, J. (1991). Divergence measures based on the Shannon entropy. *Information Theory, IEEE Transactions on*, *37*, 145–151.

Luhtala, A. (2000). *On the origin of syntactical description in Stoic logic*. Münster, Germany: Nodus Publikationen.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, (pp. 281–297).

Magerman, D. M. (1994). *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Stanford University.

Manning, C. (2011). Part-of-speech tagging from 97% to 100%: is it time for some linguistics? *Computational Linguistics and Intelligent Text Processing*, (pp. 171–189).

Marcus, M. (2011). A brief history of the penn treebank. Johns Hopkins University CLSP Seminar.
URL `http://www.clsp.jhu.edu/documents/mmarcus-2011.pptx`

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, *19*, 331–330.

Marshall, I. (1983). Choice of grammatical word-class without global syntactic analysis: Tagging words in the LOB corpus. *Computers and the Humanities*, *17*, 139–150.

Martineau, J. (1866). *Essays philosophical and theological*. Cambridge, MA, USA: William V. Spencer.

Matilal, B. K. (1990). *The word and the world: India's contribution to the study of language*. Oxford, United Kingdom: Oxford University Press.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning theory and kernel machines*, (pp. 173–187). Springer.

Melamed, I. D. (1998). Manual annotation of translational equivalence: The Blinker project. Tech. rep., University of Pennsylvania.

Melamed, I. D. (2008). Annotation style guide for the blinker project, version 1.0.4. Tech. rep., University of Pennsylvania.

Melčuk, I. A. (1988). *Dependency syntax: theory and practice*. New York, NY, USA: State University Press of New York.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, *20*, 155–172.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*, 1087.

Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American statistical association*, *44*, 335–341.

Milin, P., Kuperman, V., Kostic, A., & Baayen, R. H. (2009). Paradigms bit by bit: An information theoretic approach to the processing of paradigmatic structure in inflection and derivation. *Analogy in grammar: Form and acquisition*, (pp. 214–252).

Mingqin, L., Juanzi, L., Zhendong, D., Zuoying, W., & Dajin, L. (2003). Building a large Chinese corpus annotated with semantic dependency. In *Proceedings of the second SIGHAN workshop on Chinese language processing*, (pp. 84–91).

Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.

Moscoso del Prado Martín, F. (in press). The universal 'shape' of human languages: spectral analysis beyond speech. *PLoS One*.

Moscoso del Prado Martín, F., Kostić, A., & Baayen, R. H. (2004). Putting the bits together: An information theoretical perspective on morphological processing. *Cognition*, *94*(1), 1–18.

Murray, L. (1798). *English grammar, adapted to the different classes of learners: with an appendix, containing rules and observations, for assisting the more advanced students to write with perspicuity and accuracy*. York: Wilson, Spence and Mawman, 4th ed.

Naseem, T., Chen, H., Barzilay, R., & Johnson, M. (2010). Using universal linguistic knowledge to guide grammar induction. In *Proceedings of EMNLP*, (pp. 1234–1244).

Naseem, T., Snyder, B., Eisenstein, J., & Barzilay, R. (2009). Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, *36*, 341–385.

Neal, R. M. (2003). Slice sampling. *Annals of statistics*, *31*, 705–741.

Nevins, A., Pesetsky, D., & Rodrigues, C. (2009). Pirahã exceptionality: A reassessment. *Language*, *85*, 355–404.

Nida, E. A., & Taber, C. R. (1969). *The theory and practice of translation*. Helps for Translators. Leiden, Netherlands: E. J. Brill.

Och, F. J. (1999). An efficient method for determining bilingual word classes. In *Proceedings of EACL*, (pp. 71–76).

Och, F. J., & Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of COLING*, (pp. 1086–1090).

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*, 19–51.

Oflazer, K., Say, B., Hakkani-Tür, D. Z., & Tür, G. (2003). *Building A Turkish Treebank*, chap. 1, (pp. 1–17). Kluwer Academic Publishers.

Petrov, S., Das, D., & McDonald, R. (2011). A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Petrov, S., & McDonald, R. (2012). Overview of the 2012 shared task on parsing the web. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).

Pitman, J., & Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, *25*(2), 855–900.

Plato (360 BCE.). Sophist. Trans. Harold N. Fowler (1921) as *Plato in Twelve Volumes, Vol. 12*.

Polguáere, A., & Melčuk, I. A. (2009). *Dependency in linguistic description*. Studies in Language Companion Series. Amsterdam, Netherlands: John Benjamins Pub.

Proctor, P. (Ed.) (1978). *Longman dictionary of contemporary English*. Harlow, United Kingdom: Longman Group.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, (pp. 257–286).

Rasmussen, C. E. (2000). The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, *12*, 554–560.

Ratnaparkhi, A., et al. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP*, (pp. 133–142).

Rauh, G. (2010). *Syntactic categories: Their identification and description in linguistic theories*. Oxford University Press.

Ravi, S., & Knight, K. (2009). Minimized models for unsupervised part-of-speech tagging. In *Proceedings of ACL-IJCNLP*, (pp. 504–512). Suntec, Singapore.

Redington, M., Chater, N., & Finch, S. (1998). Distributional information: a powerful cue for acquiring syntactic categories. *Cognitive Science*, *22*, 425–469.

Resnik, P., & Hardisty, E. (2010). Gibbs sampling for the uninitiated. Tech. Rep. CS-TR-4956, Institute for Advanced Computer Studies, Maryland University College Park.

Resnik, P., Olsen, M., & Diab, M. (1999). The Bible as a parallel corpus: Annotating the "Book of 2000 Tongues". *Computers and the Humanities*, *33*, 129–153.

Rimell, L., Clark, S., & Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of EMNLP*, (pp. 813–821).

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

Robins, R. H. (1969). *A short history of linguistics*. London, United Kingdom: Longmans, 2nd ed.

Robinson, J. J. (1970). Dependency structures and transformational rules. *Language*, (pp. 259–285).

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of EMNLP-CoNLL*, (pp. 410–420).

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., & Scheffczyk, J. (2006). Framenet II: Extended theory and practice.
URL `http://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf`

Rush, A. M., Sontag, D., Collins, M., & Jaakkola, T. (2010). On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, (pp. 1–11).

Sampson, G. (1995). *English for the computer: The SUSANNE corpus and analytic scheme*. Oxford, United Kingdom: Oxford University Press.

Sánchez-León, F., & Nieto-Serrano, A. F. (1997). Retargeting a tagger. In R. Garside, G. Leech, & T. McEnery (Eds.) *Corpus annotation: linguistic information from computer text corpora*. London, United Kingdom: Longman.

Santorini, B. (1990). Part-of-speech tagging guidelines for the penn treebank project (3rd revision). Tech. Rep. MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

Schachter, P. (1985). Parts of speech systems. In T. Shopen (Ed.) *Language typology and syntactic description*, (pp. 3–61). Cambridge, United Kingdom: Cambridge University Press.

Schmerling, S. F. (1983). Two theories of syntactic categories. *Linguistics and Philosophy*, *6*, 393–421.

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of EACL*, (pp. 141–148). San Francisco, CA, USA.

Schütze, H., & Singer, Y. (1994). Part-of-speech tagging using a variable memory Markov model. In *Proceedings of ACL*, (pp. 181–187).

Schwartz, R., Abend, O., Reichart, R., & Rappoport, A. (2011). Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *Proceedings of ACL-HLT*, (pp. 663–672).

Sgarbas, K., Fakotakis, N., & Kokkinakis, G. (1995). A PC-KIMMO-based morphological description of Modern Greek. *Literary and Linguistic Computing*, *10*, 189–201.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3–4), 591–611.

Shen, L., Satta, G., & Joshi, A. K. (2007). Guided learning for bidirectional sequence classification. In *Proceedings of ACL*, (pp. 760–767).

Simov, K., Osenova, P., Simov, A., & Kouylekov, M. (2004). Design and implementation of the Bulgarian HPSG-based treebank. *Research on Language & Computation*, *2*, 495–522.

Sirts, K., & Alumäe, T. (2012). A hierarchical Dirichlet process model for joint part-of-speech and morphology induction. In *Proceedings of NAACL-HLT*, (pp. 407–416). Montréal, Canada.

Smith, K., Kirby, S., & Brighton, H. (2003). Iterated learning: a framework for the emergence of language. *Artificial Life*, *9*, 371–386.

Smith, N. A. (2012). Adversarial evaluation for models of natural language. *CoRR*, *abs/1207.0245*.

Smith, N. A., & Eisner, J. (2005a). Contrastive estimation: training log-linear models on unlabeled data. In *Proceedings of ACL*, (pp. 354–362).

Smith, N. A., & Eisner, J. (2005b). Guiding unsupervised grammar induction using contrastive estimation. In *Proceedings of IJCAI Workshop on Grammatical Inference Applications*, (pp. 73–82).

Smrž, O., & Pajas, P. (2004). MorphoTrees of Arabic and their annotation in the TrEd environment. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, (pp. 38–41).

Snyder, B., & Barzilay, R. (2008). Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-HLT*.

Snyder, B., Naseem, T., & Barzilay, R. (2009). Unsupervised multilingual grammar induction. In *Proceedings of ACL-IJCNLP*, (pp. 73–81).

Souter, C. (1989). *A Short handbook to the polytechnic of Wales corpus*. Norway: ICAME, Norwegian Computing Centre for the Humanities, Bergen University.

Spitkovsky, V. I., Alshawi, H., Chang, A. X., & Jurafsky, D. (2011a). Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of EMNLP*, (pp. 1281–1290).

Spitkovsky, V. I., Alshawi, H., & Jurafsky, D. (2010a). From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Proceedings of NAACL-HLT*, (pp. 751–759).

Spitkovsky, V. I., Alshawi, H., & Jurafsky, D. (2011b). Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction. In *Proceedings of EMNLP*, (pp. 1269–1280).

Spitkovsky, V. I., Alshawi, H., & Jurafsky, D. (2011c). Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of CoNLL*, (pp. 19–28).

Spitkovsky, V. I., Alshawi, H., Jurafsky, D., & Manning, C. D. (2010b). Viterbi training improves unsupervised dependency parsing. In *Proceedings of CoNLL*, (pp. 9–17).

Starke, M. (2009). Nanosyntax: A short primer to a new approach to language. *Nordlyd*, *36*, 1–6.

Steedman, M. (2001). *The syntactic process*. Cambridge, MA, USA: MIT press.

Täckström, O., McDonald, R., & Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of NAACL-HLT*, (pp. 477–487).

Teh, Y. W. (2006). A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of COLING-ACL*, (pp. 985–992).

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Paris, France: Libraire C. Klincksieck.

Toutanova, K., & Johnson, M. (2007). A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL-HLT*, (pp. 173–180).

Toutanova, K., & Manning, C. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference On Empirical Methods In Natural Language Processing And Very Large Corpora*, (pp. 63–70).

United Bible Societies (2013). Bible translation.
URL `http://www.unitedbiblesocieties.org/sample-page/bible-translation`

Vadas, D. (2009). *Statistical Parsing of Noun Phrase Structure*. Ph.D. thesis, University Of Sydney.

Van Gael, J., Vlachos, A., & Ghahramani, Z. (2009). The infinite HMM for unsupervised PoS tagging. In *Proceedings of EMLNP*, (pp. 678–687).

Vauvenargues, L. d. C. (1747). *Introduction a la connoissance de lesprit humain, suivie de reflexions et de maximes.*. Paris, France: Antoine-Claude Briasson.

Virpioja, S., Väyrynen, J. J., Creutz, M., & Sadeniemi, M. (2007). Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. *Machine Translation Summit XI*, *2007*, 491–498.

Vlachos, A., Korhonen, A., & Ghahramani, Z. (2009). Unsupervised and constrained Dirichlet process mixture models for verb clustering. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, (pp. 74–82).

Vogel, S., Ney, H., & Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proceedings of COLING*, (pp. 836–841).

Weischedel, R., Schwartz, R., Palmucci, J., Meteer, M., & Ramshaw, L. (1993). Coping with ambiguity and unknown words through probabilistic models. *Computational linguistics*, *19*, 361–382.

Wierzbicka, A. (2001). *What did Jesus mean?: explaining the Sermon on the Mount and the parables in simple and universal human concepts*. Oxford University Press on Demand.

Wikipedia (2013). List of literary works by number of translations.
URL `http://en.wikipedia.org/wiki/List_of_literary_works_by_number_of_translations`

Wren, P. C., & Martin, H. (1995). *High school English grammar & composition*. New Delhi, India: S.Chand & Company.

Yamada, H., & Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*.

Yarowsky, D., & Ngai, G. (2001). Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of NAACL*, (pp. 1–8).

Zwicky, A. M. (1985). Heads. *Journal of Linguistics*, *21*, 1–29.