

Applying Particle Filtering to Unsupervised Part-of-Speech Induction



Greg Dubbin
Wolfson College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2014

This thesis is dedicated to
my wife, Jen
for crossing an ocean to tackle the unknown with me.
You deserve this as much as I.

Acknowledgements

I could not have finished this thesis without the assistance and support of many people. These few words cannot adequately express my gratitude to all of my friends, family, and colleagues who helped me on this long journey.

I must first thank my supervisors, Phil Blunsom and Stephen Pulman. Phil patiently shared his knowledge and experience, helping me to learn and guiding me away from needless distractions. I am the researcher that I am today because of his instruction. Stephen introduced me to the field of natural language processing during my MSc, and provided a safety net during my DPhil.

I also thank all of my colleagues in the Computational Linguistics group at the University of Oxford for the interesting discussions and keeping my research enjoyable.

My mentor Jean De Barros has been an amazing source of encouragement and advice. I thank him for helping me stay organised and on track during the final stretch of my thesis.

I would also like to thank my family for their emotional and financial support. I am especially grateful to my parents, Susan and David Dubbin; my grandparents, Marjorie and Arthur Hockey and Helene and Murray Dubbin; my mother and father in law, Debora and Jeffrey Meyerson; and my wife's grandparents, Tillie and Herbert Meyerson.

Additionally, I would like to thank all of my friends from Oxford and the States. They helped keep me sane and expressed an interest in my thesis. Best of all, they always made sure to take me out to grab a pint.

Finally, I am eternally grateful to my wife, Jen, who has supported me from the beginning. Thank you for listening to me endlessly talk about my research and for being a second pair of eyes when I needed them. Thank you for staying by my side through the good times and the bad.

Abstract

Statistical Natural Language Processing (NLP) lies at the intersection of Computational Linguistics and Machine Learning. As linguistic models incorporate more subtle nuances of language and its structure, standard inference techniques can fall behind. One such application is research on the unsupervised induction of part-of-speech tags. It has the potential to improve both our understanding of the plausibility of theories of first language acquisition, and Natural Language Processing applications such as Speech Recognition and Machine Translation. Sequential Monte Carlo (SMC) approaches, i.e. particle filters, are well suited to approximating such models.

This thesis seeks to determine whether one application of SMC methods, particle Gibbs sampling, is capable of performing inference in otherwise intractable NLP applications. Specifically, this research analyses the benefits and drawbacks to relying on particle Gibbs to perform unsupervised part-of-speech induction without the flawed one-tag-per-type assumption of similar approaches.

Additionally, this thesis explores the affects of type-based supervision with tag-dictionaries extracted from annotated corpora or from the wiktionary. The semi-supervised tag dictionary improves the performance of the local Gibbs PYP-HMM sampler enough to nearly match the performance of the particle Gibbs type-sampler.

Finally, this thesis also extends the Pitman-Yor HMM tagger of Blunsom and Cohn (2011) to include an explicit model of the lexicon which encodes those tags from which a word-type may be generated. This has the effect of both biasing the model to produce fewer tags per type and modelling the tendency for open class words to be ambiguous between only a subset of the available tags. Furthermore, I extend the type based particle Gibbs inference algorithm to simultaneously resample the ambiguity class as well as tags for

all of the tokens of a given word type. The result is a principled probabilistic model of part-of-speech induction that achieves state-of-the-art performance.

Overall, the experiments and contributions of this thesis demonstrate the applicability of the particle Gibbs sampler and particle methods in general to otherwise intractable problems in NLP.

Contents

1	Introduction	1
1.1	Thesis Structure	3
1.1.1	Problem Description	3
1.1.2	Contributions	6
2	Parts of Speech	9
2.1	Introduction	9
2.2	History of Part-of-Speech	10
2.2.1	Traditional Parts of Speech	10
2.2.2	The American Structuralism	16
2.2.3	Modern Linguistics	20
2.3	Part-of-Speech Tagging	25
2.3.1	Automatic Part-of-Speech Tagging and Corpora	25
2.3.2	Supervised Part-of-Speech Tagging	30
2.3.3	Part-of-Speech Disambiguation	32
2.3.4	Unsupervised Part-of-Speech Induction	33
2.4	Defining Part-of-Speech	37
3	Hierarchical Non-Parametric Models and the PYP-HMM	39
3.1	Introduction	39
3.2	The Part-of-Speech Hidden Markov Model	40

3.3	Bayesian Inference	43
3.4	Non-Parametric Models	48
3.4.1	Dirichlet Processes	48
3.4.2	Pitman-Yor Processes	51
3.4.3	Hierarchical Non-Parametric Models	53
3.5	The PYP-HMM	53
3.5.1	Inference with the PYP-HMM	56
4	Unsupervised Bayesian Part of Speech Inference with Particle	
	Gibbs	59
4.1	Introduction	59
4.2	The Pitman-Yor Hidden Markov Model	62
4.3	Sequential Monte Carlo	62
4.3.1	Sentence Sampling	64
4.3.2	Type Sampling	66
4.4	Experiments and Results	69
4.4.1	Data	70
4.4.2	SMC Analysis	70
4.4.3	Unsupervised Part-of-Speech Tagging	77
4.5	Discussion	80
5	Type-based Semi-supervision	83
5.1	Introduction	83
5.2	Background	85
5.3	Labelled Semi-Supervised Part-of-Speech Discrimination	86
5.3.1	Traditional Semi-Supervision	87
5.3.2	Type-Based semi-supervision	91
5.3.3	Combined Supervision	93

5.4	Wikily-Supervised Part-of-Speech Induction	96
5.5	Discussion	100
6	Modelling the Lexicon in Unsupervised Part-of-Speech Induc-	
	tion	105
6.1	Introduction	105
6.2	Background	107
6.3	The Pitman-Yor Lexicon Hidden Markov Model	108
6.3.1	PYP-HMM	108
6.3.2	The Lexicon HMM	111
6.4	Inference	114
6.5	Experiments and Results	118
6.5.1	Unsupervised Part-of-Speech Tagging	118
6.5.2	Lexicon Analysis	122
6.6	Discussion	125
7	Conclusion	127
7.1	Future Work	131
	Bibliography	136

List of Figures

2.1	Anderson’s (1997) primary categories for English.	23
4.1	Example of the sentence particle filter	65
4.2	Posterior Log-Likelihood of PYP-HMM with exact as well as PG sent sampler	72
4.3	Posterior Log-Likelihood of PYP-HMM with with the type sam- pler as well as the 1HMM approximation	74
4.4	Posterior Log-Likelihood of PYP-HMM-LM with the sent , type , and 1HMM samplers	76
4.5	Posterior Log-Likelihood of PYP-HMM with mixed samplers . .	78
5.1	Sentence with the Penn Treebank and the Universal tagsets . .	97
6.1	Lex-HMM Structure	112
6.2	Ambiguity Class Distribution	113

List of Tables

2.1	Lexical categories proposed by Jackendoff	22
2.2	Parts of speech throughout history	26
4.1	PYP-HMM M-1 accuracies on multiple languages and samplers	79
5.1	Fully supervised and semi-supervised PoS tagging accuracy . . .	88
5.2	Semi-supervised accuracy with weighted labelled data	90
5.3	Weighted semi-supervised learning	90
5.4	Type-based supervision accuracy	92
5.5	Combined semi-supervision with varying weights	94
5.6	Combined semi-supervision with local and type samplers	95
5.7	Accuracy of wikily-supervised part-of-speech taggers over 9 lan- guages.	99
6.1	M-1 Accuracy of Lex-HMM and other samplers on the WSJ Corpus	119
6.2	M-1 Accuracy of Lex-HMM and PYP-HMM models on various languages	120
6.3	Selection of ambiguity classes predicted by Lex-HMM	123

Chapter 1

Introduction

With the introduction of statistical methods to natural language processing (NLP), researchers have been steadily developing new models and applications. Many of these applications rely on the analysis of the syntactic categories of the words in a sentence. The word-level syntactic categories, often called *parts of speech* (PoS), provide a considerable benefit to many applications without requiring deeper, more complicated analysis of a sentence. The part-of-speech of a word describes its syntactic function, allowing many models to generalise over many different words with similar behaviour. Intricate, hierarchical non-parametric models have been developed in an attempt to model these and other linguistic phenomena. These models incorporate long-range correlations that complicate standard inference techniques. This thesis explores *particle filters* as a new technique for performing inference on these intricate models, focusing on unsupervised induction of parts of speech.

The task of *supervised part-of-speech tagging* was one of the first successful applications of statistical machine-learning techniques to linguistic problems. A part-of-speech tagger assigns a part-of-speech label, or *tag*, to each word in a sentence. Supervised part-of-speech taggers rely on manually annotated

corpora to learn common patterns for each part-of-speech. These approaches are now capable of correctly labelling 97% of the words in an English corpus with their parts of speech (Manning, 2011).

On the other hand, unsupervised part-of-speech taggers have largely relied on manually created dictionaries listing the possible tags of each word. As NLP spreads to new languages and domains, even this small amount of supervision is undesirable in many applications. Furthermore, research into language acquisition relies on models of language that do not assume such specific knowledge. However, without these tag-dictionaries, a word’s part-of-speech was too ambiguous for standard models.

In the last decade, considerable improvement has been made on the related task of *unsupervised part-of-speech induction*. Non-parametric models are responsible for much of this improvement, using Bayesian priors to more closely reflect natural language. The introduction of hierarchical Pitman-Yor process priors with the PYP-HMM lead to further improvements (Blunsom and Cohn, 2011). This model reflects intricate correlations between similar words in multiple sentences, making inference difficult. These correlations forced Blunsom and Cohn (2011) to resort to a *one-tag-per-type* approximation, which assumes all occurrences of a word have the same part-of-speech tag.

An alternative approach explored in this thesis leverages the *Particle Gibbs* (PG) sampler to perform tractable inference without any such restrictions. The Particle Gibbs sampler incorporates particle filters into a Gibbs sampler by relying on their capacity to generate samples for each iteration of a Gibbs sampler. Particle filters generate multiple, independent sequences of random variables, accounting for global dependencies after they are satisfied. Thus, they are able to tractably perform inference on hierarchical, non-parametric models like the PYP-HMM. In fact, the Particle Gibbs sampler can perform

inference on even more intricate models, enabling the development of models reflecting more of the subtlety of language, which is what is done in this thesis.

1.1 Thesis Structure

The thesis is structured into two chapters (2 and 3) reviewing different aspects of the problems addressed by this thesis in the literature and three chapters (4, 5, and 6) describing solutions to the problem and their extensions. The final chapter (7) discusses the results from this research and possible future extensions to this work.

Section 1.1.1 summarizes the two aspects to the problem of unsupervised part-of-speech tagging with particle filters. The first is the history of the concept of part-of-speech from ancient Greek philosophy, to modern linguistics, to recent computational approaches to automatic part-of-speech tagging. The other aspect is the development of the hierarchical, non-parametric Bayesian models, including the PYP-HMM extended in this thesis.

The contributions of this thesis represent the key results of the chapters described in section 1.1.2. The content chapters first explore the application of the Particle Gibbs algorithm to unsupervised PoS induction with the PYP-HMM. The remaining chapters extend this approach to address questions brought up by the initial results. These extensions include semi-supervised tag-dictionaries and the development of a new unsupervised PoS model that incorporates a more explicit model of the lexicon.

1.1.1 Problem Description

Chapter 2 describes the history of part-of-speech and part-of-speech tagging, resulting in the definition of part-of-speech as word-level grammatical cate-

gories that allow a deeper syntactic analysis. The traditional parts of speech can be traced back to ancient Greeks at the Library of Alexandria attempting to describe classes of words that behave similarly. From then on, part-of-speech has been an integral part of describing a language’s grammar. The American Structuralists refined the concept of part-of-speech to incorporate the concept of *substitutability*. They defined a part-of-speech as a class of words that appear in interchangeable contexts (Harris, 1946).

Automatic part-of-speech taggers were developed with this definition, allowing the creation of part-of-speech annotated corpora and simplifying the task of grammatical parsing. These annotated corpora enabled supervised part-of-speech taggers to use statistical machine learning techniques, resulting in excellent accuracy. While other techniques were successfully applied to case of part-of-speech disambiguation with a tag-dictionary, they did not transfer well to the much more ambiguous task of unsupervised part-of-speech induction. This ambiguity was recently addressed with modern hierarchical non-parametric Bayesian models, as demonstrated by the PYP-HMM (Blunsom and Cohn, 2011). However, while these models are more adapted to modelling linguistic phenomena, they are more difficult for standard inference.

Chapter 3 explores the motivation and formulation of Bayesian inference with hierarchical non-parametric models. While supervised and semi-supervised techniques were relatively successful with a simple bigram HMM model, it has too much ambiguity for unsupervised PoS induction. Bayesian inference defines a prior distribution over the model parameters to capture the structure of the parameters and define their ambiguity. Non-parametric models like the Dirichlet and Pitman-Yor processes are particularly well adapted to modelling the linguistic structure with parameters of an unknown number or order. These models can be organized into hierarchies that connect related parameters, like

those of similarly spelled words or parts-of-speech with similar context.

Chapter 3 concludes with a description of the PYP-HMM model that serves as the basis for the experiments and extensions in the remainder of this thesis. This is a hierarchical non-parametric HMM that shows promising results. A non-parametric hierarchy allows the model to back off from trigram to bigram and ultimately unigram transitions. A further extension uses a hierarchical non-parametric character language model to connect similarly spelled words and model morphology. This morphological extension has too many long-range correlations, making it unable to mix properly without simultaneously sampling all word-tokens of the same word-type. Blunsom and Cohn (2011) were forced to resort to an approximation of these samplers that assumes all tokens of the same word-type share the same tag, called the one-tag-per-type assumption.

Problem Statement: “Can Particle Gibbs enable or improve unsupervised part-of-speech induction tractably, without the one-tag-per-type constraint?” As a fundamental unit of syntax, part-of-speech is a critical task for many NLP applications. Furthermore, the one-tag-per-type assumption is unsatisfying, as many words that might have multiple parts of speech are critical to the grammatical structure of a sentence. For example, mislabelling *bank* might lead to a sentence that is missing a subject or a main verb. Successfully applying Particle Gibbs sampling to unsupervised PoS induction without relying on the one-tag-per-type assumption both proves its viability as a solution to otherwise intractable non-parametric inference problems and improves the viability of unsupervised part-of-speech induction.

1.1.2 Contributions

Success on this problem is evaluated according to three metrics: posterior likelihood of the resulting model, many-to-one accuracy of the tag assignments across multiple languages, and the speed of inference. Posterior likelihood of two different inference techniques on the same model provides a direct comparison between them; if the PG sampler can find more likely assignments, then it is a superior technique for that model. Many-to-one (M-1) accuracy enables comparison of different models across multiple languages; unsupervised PoS induction should perform well in a variety of languages. Christodoulopoulos et al. (2010) found that M-1 accuracy is an intuitive PoS evaluation metric that reflects performance as long as the number of tags is held constant. Finally, speed addresses the issue of tractability, the potential improvements from Particle Gibbs are of little benefit if they come at too great of a computational cost. This thesis shows that Particle Gibbs inference finds more likely tag assignments without the one-tag-per-type assumption. It then develops a new model of part-of-speech that matches or exceeds the accuracy of the one-tag-per-type assumption across multiple languages and twice as quickly.

The contributions of the research presented in this thesis are described over three chapters. The first chapter describes two different implementations of the Particle Gibbs algorithm to the PYP-HMM model, analysing their behaviour and comparing them to previous approaches. This results in the discovery that the one-tag-per-type assumption compensates for a flaw in the model. The second chapter addresses this flaw with semi-supervised tag-dictionaries, comparing their effects with and without the Particle Gibbs algorithm. The last content chapter proposes an extension to the unsupervised model, the Lex-HMM, that improves the base model and compares favourably to the one-tag-per-type approximation.

The first contribution from Chapter 4 is the description of two different implementations of the Particle Gibbs sampler for the PYP-HMM. The first implementation performs block sampling over sentences, allowing comparison with an existing block sampling algorithm based on the forward-backward algorithm. Analysis of the sentence samplers reveal that the Particle Gibbs algorithm improves with the addition of more particles only up to a point. Furthermore, this sampler can match the performance of the efficient forward-backward sampler in less time.

The second sampler implemented in Chapter 4 samples all word-tokens of the same word-type, mimicking the one-tag-per-type approximation, called the PYP-1HMM, without the restriction. Results show that the type-sampler finds more likely tag assignments under the PYP-HMM model than the PYP-1HMM, though it is considerably slower. The final analysis of the accuracy of each sampler reveals that while the PG type sample finds more likely tag assignments, they are less accurate than those found by the approximation. These results show that the PYP-HMM assigns a high probability to tag assignments with many tags per type, and the one-tag-per-type restriction achieved its high accuracy by ignoring those assignments. The main contribution of Chapter 4 is demonstrating empirically that the Particle Gibbs sampler is capable of performing inference on otherwise intractable models.

Chapter 5 addresses the issue of too many tags per type with the inclusion of semi-supervised tag-dictionaries. The first tag-dictionaries are extracted from a subset of labelled training data and applied to the traditional semi-supervised discrimination task. These experiments found that the inclusion of tag-dictionary minimizes the benefit of type sampling with or without Particle Gibbs. These results were then reinforced during experiments with tag-dictionaries extracted from the Wiktionary. The Particle Gibbs nearly matches

the performance of the slower model of Li et al. (2012). However, the standard local Gibbs sampler also achieves nearly the same performance while being considerably faster still. Another contribution of these results was the discovery that the 1HMM underperforms with a tag-dictionary because it cannot take full advantage of the reduced ambiguity.

The final contribution of this thesis is the creation of a new model of part-of-speech, the Lex-HMM, in Chapter 6. The Lex-HMM extends the PYP-HMM with an unsupervised model of the lexicon. This lexicon directly models the generation of each word-type’s ambiguity class, the set of tags that might be assigned to a word. The Lex-HMM implements a basic model of ambiguity classes that allows a soft restriction on the number of tags assigned to each word-type. This soft constraint allows the Lex-HMM to address the flaw of the PYP-HMM without resorting to the one-tag-per-type assumption.

Furthermore, Chapter 6 describes an extension of the type-based Particle Gibbs sampler from Chapter 4 to perform inference on the Lex-HMM. The Particle Gibbs sampler is necessary for PoS induction with the Lex-HMM because the model has even more long-range constraints than the standard PYP-HMM. With the Particle Gibbs sampler, the Lex-HMM is able to match or exceed the accuracy of the 1HMM on most of 10 different languages. Beyond that, the Lex-HMM achieves this performance in half the time as the 1HMM.

Chapter 7 concludes the thesis with an overview of these contributions. The conclusion also discusses interesting questions and areas for expansion for future research in unsupervised part-of-speech induction and the application of particle filters to NLP problems. The overall contribution of the thesis is the establishment of the utility of particle filters in applications with long-range correlations and the creation of the Lex-HMM model that relies on such particle filters.

Chapter 2

Parts of Speech

2.1 Introduction

Part-of-speech, also referred to as word-class or lexical category, is a word-based unit of syntax. As such, part-of-speech tagging is often the first step in the NLP pipeline. It provides a surface-level understanding of sentence structure and an obvious clustering to allow generalisation in machine learning.

Many applications have come to rely on part-of-speech tagging because supervised PoS tagging is considered a “solved” problem (Manning, 2011). The large amounts of labelled data in several major languages has resulted in very high accuracies. Similarly, the relatively limited search results in a fast first pass for a deeper analysis. However, these results are generally language and domain dependent, and there is less labelled data in many important or rising domains and languages, e.g. Twitter and Arabic or Indonesian.

On the other hand, unsupervised PoS tagging is still an open problem. Early results depended on difficult to replicate, finely tuned dictionaries (Banko and Moore, 2004). Recent techniques aim to eliminate the involvement of expensive experts.

This chapter aims to define part-of-speech and the problem of part-of-speech tagging. To provide context for the part-of-speech tagging problem and the definition of part of speech, section 2.2 covers the history of the concept of parts of speech. Then, section 2.3 explores the literature of supervised, semi-supervised, and unsupervised part-of-speech tagging. Finally, section 2.4 explains the specific definition of part-of-speech that this thesis uses and some theoretical and practical considerations for this definition.

2.2 History of Part-of-Speech

To better motivate a definition of part-of-speech, it helps to understand the history of the term. Through most of the history of the term, the concept of part-of-speech has been explained only by the distinctions made in the individual definitions of the distinct parts of speech. Therefore, the motivation of the creator in distinguishing these parts of speech is the only tool for understanding their purpose. This purpose has changed over time, and the defining features of the parts of speech have changed with it. A common theme across the centuries is that, once defined, the parts of speech are treated as the basis of more complicated grammatical structure. This results in the conflict between defining parts-of-speech as categories of word-types (word classes) and as categories of syntactic context (syntactic categories). This conflict was not resolved until modern linguists recognized the two as separate, but related concepts.

2.2.1 Traditional Parts of Speech

Most linguists credit the ancient Alexandrians Aristarchus and his pupil Dionysius Thrax with the creation of what would become the traditional parts of

speech in the second century B.C.E. (Robins, 1966). The Alexandrians were chiefly concerned with the preservation and critical analysis of poetry from earlier Greek writers (Rauh, 2010, p. 14). In this pursuit, they noticed similarities in the inflections of these words. Thrax codified these similarities in his grammatical analysis, *Tékhne grammatikē* (*The Art of Grammar*). Thrax identified 8 parts of speech: noun, verb, participle, article, pronoun, preposition, adverb, and conjunction. In his grammar, Thrax stressed that his parts of speech were determined by “analogy”, meaning categories represented words that behaved in the same way (Rauh, 2010, p. 16). This is a precursor to the distributional approach proposed by the American Structuralists. A translation (by Davidson (1874)) of the definitions Thrax gives is presented below.

1. A Noun is a declinable part of speech, signifying something either concrete or abstract (concrete, as stone; abstract, as education); common or proper (common, as man, horse; proper, as Socrates, Plato). It has five accidents: genders, species, forms, numbers, and cases.
2. A Verb is an indeclinable word, indicating time, person, and number, and showing activity or passivity. The verb has eight accidents: Moods, Dispositions (voices!), Species, Forms, Numbers, Tenses, Persons, Conjugations.
3. A Participle is a word partaking of the nature both of nouns and verbs. It has all the accidents which belong to nouns as well as those which belong to verbs, except mood and person.
4. An Article is a declinable part of speech prefixed or subjoined to the various cases of nouns.... It has three accidents: Gender, Number, and Case.
5. A Pronoun is a word assumed instead of a noun, and indicating definite persons. It has six accidents: Person, Gender, Number, Case, Form, and Species.
6. An Adverb is an indeclinable part of speech, said of a verb or added to a verb. Of the Adverbs, some are Simple, and others Compound – ...Some are indicative of time.... Some indicate manner...; some, quality...; some, quantity...; some, number...; some, place.... Some Adverbs signify a wish...; some express horror...; some, denial or negation.

7. A Preposition is a word placed before any of the parts of speech, both in Composition and Syntax.
8. A Conjunction is a word binding together a thought in order and filling up the hiatuses of speech. Of conjunctions, some are copulative, some disjunctive, some conjunctive, some præter-conjunctive, some causative, some dubitative, some conclusive, and some expletive.

These definitions focus primarily on the morphological features of the words, making them properties of the word-types. For example, nouns are distinguished by the fact that they can be inflected in case, gender, and number, resulting in the inclusions of adjectives which behave the same way in Greek. Later, Apollonius Dyscolus used these parts of speech in his description of Greek syntax, first connecting the concept of part-of-speech with syntax (Robins, 1966). This approach to part-of-speech inspired the Romans Remnius Palaemon and later Priscian to use almost the same 8 parts of speech ¹ in their descriptions of Latin grammar. This is a natural extension, as Latin grammar is largely inflectional, so when a part-of-speech dictates what inflections a word can take, it also dictates that word's grammatical function. However, this approach conflates the word-type based word classes and the context based syntactic categories, starting a trend that would continue for millennia.

The concept of part-of-speech continued with Lindley Murray's *English Grammar*, first published in 1795 (1809). This grammar defined 9 parts of speech, including both the Greek article and the Roman interjection, but forsaking the participle in favour of distinguishing adjectives from nouns. The adjective is described as modifying a noun, syntactically motivated by the fact that an adjective requires a noun while the reverse is not true. The descriptions of Murray's (1809) parts of speech are similar to Thrax's, with less of a

¹Replacing the non-existent articles with interjections.

focus on morphology and more on notional and prototypical definitions, they are (emphasis added on the part-of-speech name):

1. An *article* is a word prefixed to substantives to point them out, and to show how far their signification extends: as, *a* garden, *an* eagle, *the* woman.
2. A *Substantive* or *noun* is the name of any thing that exists or of which we have any notion: as, *London*, *man*, *virtue*.

A substantive may, in general, be distinguished by its taking an article before it, or by its making sense of itself: as, a *book*, the *sun*, an *apple*; *temperance*, *industry*, *chastity*.

3. An *Adjective* is a word added to a substantive, to express its quality: as, “An *industrious* man; *a virtuous* woman.”

An adjective may be known by its making sense with the addition of the word *thing*: as, a *good* thing; a *bad* thing: or of any particular substantive; as, a *sweet* apple, a *pleasant* prospect, a *lively* boy.

4. A *Pronoun* is a word used instead of a noun, to avoid the too frequent repetition of the same word: as, “The man is happy; *he* is benevolent; *he* is useful.”
5. A *Verb* is a word which signifies to BE, to DO, or to SUFFER: as, “I *am*; I *rule*; I *am ruled*.”

A verb may generally be distinguished by its making sense with any of the personal pronouns, or the word to before it: as, I *walk*, he *plays*, they *write*; or, to *walk*, to *play*, to *write*.

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: as, “He reads *well*; a *truly* good man; he writes *very correctly*.”

An adverb may generally be known, by its answering to the question, How? how much? when? or where? as, in the phrase, “He reads *correctly*,” the answer to the question, How does he read? is, *correctly*.

7. *Prepositions* serve to connect words with one another, and to show the relation between them: as, “He went *from* London *to* York.” “she is *above* disguise;” “they are supported *by* industry.”

A preposition may be known by its admitting after it a personal pronoun, in the objective case: as, *with*, *for*, *to*, &c.

8. A *Conjunction* is a part of speech that is chiefly used to connect sentences; so as, out of two or more sentences, to make but one. it sometimes connects only words: as “Thou *and* he are happy, *because* you are good.” “Two *and* three are five.”
9. *Interjections* are words thrown in between the parts of a sentence, to express the passions or emotions of the speaker: as “O virtue! how amiable though art!”

Each notional definition is accompanied by examples demonstrating the part-of-speech’s syntactic distribution. Murray (1809) goes on to describe English Syntax as a set of rules applying to the syntactic behaviour of the parts-of-speech and their subcategories.

This tradition of morphologically defining parts of speech to aid in describing a languages syntax continued into the 19th century with Diez’s *Grammatik der romanischen Sprachen* (Diez, 1836-38). Diez described several Romance languages in a similar style to Priscian’s, with primarily morphological descriptions of the parts of speech that are used to describe the grammar of each language. Diez describes 10 parts of speech, using syntactic differences to distinguish adjectives from nouns, adding a part-of-speech for numerals, reintroducing the article, and ignoring participles.

The morphological definitions used in defining the Greek and Romance parts of speech are less useful in languages with less inflection. Therefore, when Mätzner published his Modern English grammar, his definitions focused on semantic features of the words (Mätzner, 1880-85, vol. 1). Mätzner describes the same 10 parts of speech as Diez and discusses their inflectional properties with respect to Old English. However, these inflectional properties no longer exist in Modern English for the most part. Still, having defined his parts of speech, Metzer proceeds to describe their syntactic properties to define a grammar for simple and complex sentences (Rauh, 2010, p. 23).

Rauh (2010, p. 24) argues that Mätzner was influenced by a second, older tradition in his definition of the parts of speech, that of Plato and the Stoics. The Greek philosopher Plato used the term “*mérē lógou*” (sentence parts) when he separated the sentence (*lógos*) into *ónoma* and *rhēma*. These terms are now translated into noun and verb, respectively, but Plato intended them to split the sentence into topic and comment, or argument and predicate (Robins, 1966). These terms were used more like the broad syntactic categories of Noun Phrase (*ónoma*) and Verb Phrase (*rhēma*) (Robins, 1966). Plato was attempting to analyse speech as logical statements, and his parts of speech were semantic and syntactic in nature (Rauh, 2010, p. 25). This is most apparent in Plato’s categorisation of adjectives and adjectival phrases as *rhēmata*, because they could be used as a predicate without a copula in ancient Greek. Aristotle extended Plato’s parts of speech to include a third, lesser category, the *sýndesmoi* (conjunctions), which were all of the other words in a sentence that connected the key elements, the *onómata* and the *rhēmata*.

This tradition of analysing sentences as statements of logic was continued by the Stoics (Robins, 1966). The Stoics introduced morphology into the definition of the parts of speech by distinguishing the *árthron* (a class representing articles and pronouns) that can be inflected from the rest of the *sýndesmoi*. Robins (1966) claims that by defining classes according to grammatical categories, the Stoics represent the beginning of the European grammatical tradition. The Stoics used the existence of case inflection to distinguish the grammatical purpose of the parts of speech. The fact that *rhēmata*(verbs) lacked case distinguished them from the *onómata*(nouns), and similarly the *árthra* from the *sýndesmoi*. Later, Diogenes and Chrysippus split the nouns into proper and common nouns. Other Stoics added a class of adverbs, so that the Stoic system defined 6 parts of speech (Robins, 1966). While the Stoic system

was used to describe syntactic as well as semantic function, Rauh (2010, p. 25) stresses that, like Plato and Aristotle, the intention was to better understand logical argument while the Alexandrians sought to analyse written language.

The Stoics' logical approach to grammatical analysis eventually inspired the creation of the Port Royal grammar by the linguist Claude Lancelot and philosopher Antoine Arnauld (1975). The grammar was published in a book titled *Grammaire générale et raisonnée* (*General and Rational Grammar*) in 1660. The goal was to focus on purely semantic definitions of the 8 Alexandrian parts of speech plus the interjection, so that they would be universal to all languages. Rauh (2010, p. 25) argues that this grammar fails to be universal because it attempts to force all words into 9 semantic categories that were originally distinguished according to their morphology.

The morphological basis of the original parts of speech and the semantic and syntactic similarity of the words they described lead grammarians to equate each of these aspects. This historical tradition of classifying words and then using those classes as the basis of a sentence's grammatical structure persists to this day. Even as linguists of the first half of the twentieth century eschewed the prescriptive traditions of European grammarians, they still operated under the assumption that words could be assigned to one of roughly ten classes representing their syntactic behaviour.

2.2.2 The American Structuralism

The continued reliance on conflating grammatical categories with word classes originally designed for morphological languages, as well as infusion of semantic meaning into these same categories has lead to confusion over the definition and use of parts of speech in modern linguistics starting with the American Structuralists. This movement was inspired by anthropologist Franz Boas and

codified by Leonard Bloomfield. Boas found that the Native American languages he was studying could not be adequately explained by the western grammatical tradition (Rauh, 2010, p. 27-28). Bloomfield advocated a scientific approach to language description that involved the transcription and analysis of spoken utterances (Rauh, 2010). These utterances would be used as data that would support and be explained by linguistic theories.

The Structuralist movement would lead to the distributional approaches described by linguists like Zellig Harris and Charles Carpenter Fries (Rauh, 2010, p. 33). Harris described the distributional approach as relying on the principle of *substitutability*. This approach relied on the collection of utterances into “corpora” and comparing the distribution of elements within these corpora. He used the principle of substitutability to create morpheme classes and morpheme sequence classes (Harris, 1951). Harris’s (1951, p. 363) original description of his substitutability principle was:

The morphemes are grouped into morpheme classes, or classes of morphemes-in-environments, such that the distribution of one member of a class is similar to the distribution of any other member of that class. These morpheme classes and any sequences of morpheme classes which are substitutable for them within the utterance, are now grouped into larger classes (called position or resultant classes) in such a way that all the morpheme sequences (including sequences of one morpheme) in a position class substitute freely for each other in those positions in the utterance within which that class occurs.

Harris described two methods for identifying morpheme classes using the substitutability principle. The first compared all environments that occur with each morpheme, identifying each set of morpheme with identical environments as a morpheme class. The second method alternates between expanding the set of environments in which a morpheme can be substituted and expanding the set of morphemes which can be substituted in those environments. Harris

claims that the second method is equivalent to the first on the assumption that any word in the class that can substituted in one environment can also be substituted in any environment in which one of the other words occurs. Harris combined his morpheme classes into morpheme sequence classes, built of multiple morpheme classes. The distributional behaviour of these morpheme sequence classes define a language's grammar.

Recognizing that both of these methods are limited by the lack of all possible utterances existing within a corpus, Harris relaxed the principle to only require 80% environment overlap (Harris, 1951, p. 245). Applying the second method to a variety of corpora led Harris to describe 35 morpheme classes. 18 classes represented syntactic categories, 7 of which are subcategories of the greater verb category. Harris's method did not distinguish between morphological categories and syntactic categories, so 1 class was a neutral stem class that could be combined with the remaining 16 affix classes to form a syntactic class (Harris, 1946). Rauh (2010, p. 48-53) suggests that Harris's use of the names for the traditional parts of speech affected his application of his own rules, resulting in far fewer morpheme classes than the rigorous application of substitutability would find. This is because many words can occur in different syntactic contexts, like the word *feels*, which can be substituted with *kicks* in *The boy feels the ball*, but not in *The sand feels warm*. Similarly *is* can be substituted with *feels* in *The sand is warm*, but not in the interrogative *Is the sand warm?* The combination of the principle of substitutability of context with the assignment of a single morpheme class to each morpheme results in a much larger number of classes than described by the traditional parts of speech.

Another Structuralist was by Charles Fries, a contemporary of Harris with a slightly different approach to syntactic classification. Unlike Harris, Fries

approached syntactic categories as separate from morphological ones. Fries assigned words to classes according to their substitutability into the words in 3 *test frames*, short clauses covering a range of syntactic structures (Fries, 1952, p. 75). Fries’s (1952) test frames were:

Frame A: The concert was good (always)
 Frame B: The clerk remembered the tax (suddenly)
 Frame C: The team went there

Fries identified 19 categories, 4 of which he referred to as “parts of speech” or “form classes”, because they could be identified by their morphology. He referred to the remaining 15 classes as “function words” (Fries, 1952, p. 110). Fries took care to avoid reusing the names of the traditional parts of speech, instead naming his categories with letters and numbers. Still, the limited number of test frames fails to distinguish words that cannot be substituted in other contexts. Rauh (2010, p. 49) suggests the words *is* and *feels* for the frame *The concert --- good*. Note that only *is* works in the interrogative *--- the concert good?*

The above example can be solved by splitting the verbs like *feels* and *is* into *verbs of perception* and *verbs of being*, respectively. The necessity of splitting word-classes into smaller and more specific subclasses is known as the *subclass problem* (Haspelmath, 2001). The problem with the subclass strategy is that it becomes difficult to distinguish when a word is distributed differently because of its semantics². While the categories still represent common context, they become more semantic in nature and less syntactic as they get smaller.

This is a similar problem to the one faced by Harris (1946) with substitutability. However, dividing the verbs *kicks* and *feels* into *transitive* and *intransitive* or *perception* verbs is not a universal solution. The word *feels* acts

²The word *sounds* is a similar perception verb to *feels*, yet it is unlikely to be substituted in the phrase *The sand --- warm*.

as either a transitive or intransitive verb in different contexts. Another perception word, e.g. *looks*, fits in the context frame *The sand --- warm*. However, *looks* cannot replace *feels* as a transitive verb in the frame *The boy --- the ball*. Words like *banks* can occur in even more distinct contexts, e.g. *She went to several banks* and *She banks her money*. Harris (1946) reconciled this problem with the goal of assigning one morpheme class to each morpheme by considering these cases as homophones or *homographs*, different words that have the same pronunciation or spelling. This solution leads to the *homograph problem*, where it is unclear how to determine which homograph is present in a given environment or if it even represents a new homograph.

The homograph problem and the subclass problem can often exacerbate each other. As the granularity of the subclasses becomes more fine, more homographs must be distinguished. On the other hand, if the same word in two different environments or frames is actually two different words, then what links the subclasses to their superclass? A major cause of the problems with the Harris's morpheme classes and Fries's categories was that both attempted to uniquely define a word-class that completely described the grammatical environments in which each word could occur. The homograph and subclass problems persist into modern linguistics, but they are mitigated as linguists begin to distinguish between word-classes and syntactic categories.

2.2.3 Modern Linguistics

Franz Boas and the Structuralists represent the start of a more scientific, evidence-based approach to the study of language and the predecessor of the field of modern Linguistics. Following the “Chomksyan Revolution” (Newmeyer, 1980, p. 19), modern linguists rejected the materialistic proposal of Bloomfield (1933, p. 22) that the study of language must be based solely on a finite number

of observable utterances. Instead, modern theories of language include models of how new words and utterances might be understood or generated (Rauh, 2010). A common feature of these models is the distinction of word-classes and their features, often contained within a lexicon, and the categories that make up a sentence’s grammatical structure.

After the Structuralists, linguists were divided regarding how best to describe a language’s grammar. Chomsky (1975b, 1957, 1965) proposed a Generative Grammar made up of phrases named after some of the parts of speech: Noun Phrases, Verb Phrases, etc. These phrases are made up of words divided into *lexical categories*, representing the “major” semantic parts of speech: noun, verb, and adjective, and *functional categories* representing word-classes that only represent syntactic function (Chomsky, 1975a). Chomsky (1975a) distinguished the lexical categories according to the presence of nominal ($\pm N$) and verbal ($\pm V$) features. This feature-based approach was extended by Jackendoff (1977, p. 31) to include the preposition category ($-N, -V$). The grammar contain rules describing the sub-categorical properties of the lexical categories and these properties determine a word’s grammatical function.

Jackendoff (1977) then went on to further describe the remaining functional categories with additional features for Subject, Object, Complement, and Determiner. These features represent whether or not a word can have a subject, a complement that is an NP, any complement, or whether the word is an article or word of degree, respectively. Table 2.1 shows the specific categories proposed by Jackendoff (1977), note that they are also named after the traditional parts of speech, but are not necessarily the same, especially for non-Indo-European languages (Haspelmath, 2001).

Pollard and Sag (1987) developed a rule-based model of grammar in which words and phrases are combined according to complex *feature structures*. In

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

Table 2.1: The lexical categories proposed by Jackendoff (1977, p. 33)

this approach, a syntactic category is the complete feature structure of a word, which ultimately achieves the substitutability Harris set out to accomplish (Rauh, 2010, p. 199). In the grammar of Sag et al. (2003), a lexicon pairs a word with all of its relevant syntactic, phonetic, and semantic features. The part-of-speech of a word represents its highest-level feature, determining what other features the lexicon needs to describe in order to fully determine each word’s syntactic category (Pollard and Sag, 1994, p. 22).

Langacker (1987) proposed the Cognitive Grammar as a semantic-focused response to Chomsky’s Generative Grammar. This grammar is built of symbols representing semantic and phonetic pairs, with no explicit distinction between morphemes, words, phrases, and sentences. Despite the lack of any explicit reference to syntactic categories, Langacker (1987, p. 189,214,242) still distinguishes between word-level *basic grammatical categories* and phrasal *grammatical categories*. Every grammatical element in Langacker’s Cognitive Grammar has some semantic meaning, so the basic grammatical categories are defined notionally, by their general meaning (Rauh, 2010, p. 237).

Anderson (1971) described his Localist Case Grammar as a dependency grammar that is based on word classified according to the notional features N and P. This approach was further clarified in the book *A Notional Theory of*

{P}	{P N}	{P N}	{N P}	{N}	{}
aux	verb	adjective	noun	name	functor

Figure 2.1: Anderson’s (1997) primary categories for English.

Syntactic Categories (Anderson, 1997). The Localist Case Grammar uses the ability and prevalence of a word as describing a participant (N) and as an event or situation (P) to describe *primary categories* or *basic word classes* (Anderson, 1997, p. 105). The six categories used to describe English are shown in figure 2.1, with a “;” indicating that the feature on the left is more prevalent and a “:” indicating a balance between the two. There are further intermediary classes with head features (notional and inflectional) from one word class and complement selection properties of another (Anderson, 1997, p. 82). These classes have further *secondary categories* that describe the complements a class can take, for nouns these represent Case, Gender, Definiteness/Specificity, and Number (Anderson, 1997, p. 106). These primary and secondary categories, in addition to syntactic derivation rules, are stored in a lexicon and are used to derive the specific syntactic categories of a word. Note that this approach explicitly distinguishes syntactic categories from the word classes/lexical classes from which they derive (Anderson, 1997, p. 2).

A different approach to notional categories comes from research by cognitive psychologist Rosch and her team into prototypical categories (Rosch 1977, p. 36; Mervis 1980, p. 286). This research showed that cognitive categories are often described by prototypical members with fuzzy boundaries where members have some features common to categories with different prototypes (Rosch, 1977). Lakoff (1965) found that the traditional parts of speech exhibit prototype effects. This research was incorporated into a description of lexical categories by Dixon (1977). While these prototypical categories are still used to describe a language’s grammar, they are only grammatical hints,

as some words might not behave like any of the prototypes.

Table 2.2 shows key examples of the evolution of the traditional parts of speech. These examples show that, while the identifying features have varied, part-of-speech has historically been used to identify the surface-level components of grammar. In an attempt to avoid confusion, linguists have proposed other terminology for word-level grammatical categories. Haspelmath (2001) refers to *word classes*, a much broader term referring to any classification that applies to word-types rather than word-tokens, and compares the intersection of word classes in different languages with the traditional parts of speech.

The term *lexical category* describes categories of words in a lexicon. Usage of the term varies over their relation to grammatical categories. Van Valin, Jr. (2001) names his lexical categories after the traditional parts of speech, but insists that they be defined morphosyntactically. In this definition, a word's lexicon entry defines all of the possible lexical categories a word might have, with the actual category determined individually for each occurrence according to its syntactic context. On the other hand, Chomsky distinguishes meaning-bearing lexical categories (noun, verb, adjective, preposition) and functional categories (Chomsky, 1975a), and these categories are used to determine the sub-categorical features that control syntax. Furthermore, Rauh (2002) defines lexical categories as cognitive, prototypical categories used for creating a lexicon that aid in determining a word's grammatical category, which is used in the construction of the grammar.

All of these approaches note the need to distinguish between grammatical categories of words for deeper grammatical analysis. Regardless of terminology, every language has groups of words that have very similar syntactic distributions (Haspelmath, 2001). Modern linguistics distinguishes between the type-level categories that describe possible behaviour and the specific token-level

categories that describe actual syntactic function. Both of these categories are often referred to as parts of speech because there is often considerable overlap between either of them and the traditional parts of speech. The definition of part-of-speech for any given application depends on its purpose and practical considerations in implementation.

2.3 Part-of-Speech Tagging

The work of the American Structuralists like Harris and Fries coincided with the growing access to mainframe computers at research institutions. The ability of computers to store and analyse large amounts of data was a natural fit to the rigorous corpus-based approach they proposed. One of the first successful applications of computers to language processing was automatic part-of-speech tagging, or *grammatical coding* (Klein and Simmons, 1963). With the growing body of corpora and exponential rise in processing power, the field of Computational Linguistics formed.

Later on, the creation of annotated corpora greatly simplified the task of evaluating part-of-speech taggers and enabled the use of supervised approaches. These supervised approaches were either fully supervised tagging or semi-supervised part-of-speech disambiguation using a supplied tag dictionary. Over time, researchers began to question the feasibility of the tag dictionary approach and started the tasks of minimally supervised (Hasan and Ng, 2009) and fully unsupervised part-of-speech induction.

2.3.1 Automatic Part-of-Speech Tagging and Corpora

As computers became available to researchers in the early 1960's, linguists began to use them to analyse text. Klein and Simmons (1963) created one of

Yāska	Plato	Aristotle	Stoics	Thrax	Priscian	Varo	Murray	Diez and Mätzner	Harris	Jackendoff	Croft
nāma	ónoma	ónoma	common-noun	noun	noun	noun	noun	noun	noun (N)	noun	noun
ākhyāta	rhēma	rhēma	verb	verb	verb	verb	verb	verb	verb (V)	verb	verb
upasarga		syndesmos	conjunction article adverb proper-name	conjunction article adverb	conjunction adverb	adverb	conjunction article adverb	conjunction article adverb	conjunction(&) article (T) adverb (D)	article	
				participle pronoun preposition	participle pronoun preposition interjection	participle	pronoun preposition interjection adjective	pronoun preposition interjection adjective numeral	pronoun (I) preposition (P) adjective (A)	preposition	adjective
nipāta									particle (B) modal (R) <i>have, be</i>	particle modal	
									degree quantifier		

Table 2.2: The parts of speech or lexical/syntactic categories of various grammatical systems throughout history, extended from table 2.2 presented in Christodouloupoulos (2013, p. 16).

the first automatic part-of-speech taggers, called the "computational grammar coder," or CGC. Like most of its successors, the purpose of the CGC was to assign grammatical codes to individual words as a precursor to a total parsing system. The alternative of the time was to utilize large dictionaries of words, which slowed the parsing process because they had to be stored on magnetic tape due to their size. The CGC was system of hand-crafted rules that systematically reduced the ambiguity of a word's tag assignment through a series of tests. The first test used a dictionary of 2000 common and idiosyncratic words to determine a word's ambiguity class. If a word is not in the dictionary or has an ambiguous entry, a series of suffix tests narrow the ambiguity class further. If the word still has no ambiguity class, it is assigned to the ambiguity class representing the set of open grammatical codes. The final Context Frame Test attempts to replicate Harris's (1951) substitutability property by limiting the codes in ambiguous strings of up to three words between unambiguously coded words (Klein and Simmons, 1963). A major flaw with this hand-coded rule based approach is that the rules can never truly be exhaustive and must be extended as gaps are found.

The ambiguity classes represented a combination of homograms, like Harris (1946), and words that were ambiguous because they could not fit in the limited dictionary. The CGC used a tagset of 30 codes, around twice as many as the categories proposed by contemporary linguists like Harris (1951) and Fries (1952). The extra codes are to distinguish between subcategories which behave differently syntactically, like coding nouns for number and verbs with tense, simplifying the work for the parser. Other words have specialized grammatical codes if their syntactic behaviour requires deeper analysis than the Context Frame Tests can provide. Two examples are the prepositions *to* and *of*, which have the additional grammatical functions of distinguishing infinitive verbs

and possession, respectively (Klein and Simmons, 1963).

The TAGGIT part-of-speech tagger was a similar rule-based approach by Greene and Rubin (1971). Like the CGC, TAGGIT uses a word dictionary and a suffix dictionary to determine ambiguity classes. Additionally, it has manually designed context rules based on unambiguous words within a context window of 2 words on either side (Greene and Rubin, 1971). However, the TAGGIT system was designed to automatically tag the much larger and more diverse data of the newly created Brown Corpus (Francis, 1964). The Brown corpus is comprised of around 1 million words divided into 15 categories of American fiction and non-fiction published in 1961 (Francis, 1964). Completed in 1971, the part-of-speech annotation of the Brown corpus used 82 simple tags and 135 compound tags, further distinguishing syntactically distinct subcategories of the traditional parts of speech (Francis and Kučera, 1979). TAGGIT used a dictionary of 3,000 words and a suffix list of 450 suffixes (Greene and Rubin, 1971). Additionally, TAGGIT had 3,300 context frame rules (Greene and Rubin, 1971), compared to the CGC’s 500 (Klein and Simmons, 1963). TAGGIT successfully tagged 77% of the Brown corpus, and human annotators corrected the remaining errors (Greene and Rubin, 1971).

The 1 million annotated words of the Brown Corpus enabled a new approach to PoS tagging, supervision. The first probabilistic part-of-speech tagger, CLAWS (described in detail next section), took advantage of the tag transition frequencies in the annotated Brown Corpus to assist in the annotation of the Lancaster-Oslo/Bergen (LOB) Corpus (Marshall, 1983). The use of the supervised frequency information enabled the automatic tagging of 95-96% and reducing the number of corrections needed to 35,000, compared to the 230,000 corrections in the construction of the Brown Corpus (Beale, 1985). The LOB Corpus further expanded the tagset to 135 tags in an effort to ap-

proach “the ideal of providing distinct codings for all classes of words having distinct grammatical behaviour” (Garside et al., 1987, p. 167). This stated ideal reveals the continued attempt to reconcile the two notions of classifying word-types and categorizing grammatical behaviour. Other corpora followed suit with the Lancaster UCREL group expanding to 165 tags and the London-Lund Corpus of Spoken English using 197 (Marcus et al., 1993a). The growth in size of tagsets during this time shows the consequences of this reconciliation and exemplifies the subcategory problem.

The trend of steadily increasing the size and specificity of tagsets stopped in the 1990’s with the creation of the Penn Treebank (Marcus et al., 1993a) and the British National Corpus (BNC) (Garside, 1996). The BNC was annotated with an updated version of CLAWS (CLAWS 4) using a basic tagset of 62 tags for the full corpus of 100 million words and a more extensive tagset of 152 tags for a smaller “core” of 2 million words (Garside, 1996). The Penn Treebank (PTB) went even further to create a tagset by reducing the 87 tags of the Brown tagset to 36 tags for words and 12 more for punctuation and symbols, for a total of 48 tags (Marcus et al., 1993a). The creators of the Penn Treebank were concerned with the problem of sparsity in probability estimation, so by reducing the number of tags they aimed to ensure that each tag was frequent enough to for trustworthy estimates (Marcus et al., 1993a). Other reasons for reducing the size of the tagset are the reduction in errors and inconsistency in the human corrections and in the uncertainty for the automatic taggers (Leech, 1993).

The common goal of all of these part-of-speech annotated corpora has continued to be the goal set out by the GCG of simplifying the process of full grammatical analysis. These corpora, as well as those in other languages, allow the use of supervised machine learning techniques to greatly improve

the accuracy of PoS taggers. Furthermore, the annotated corpora provide a gold-standard against which to evaluate the results of both supervised and unsupervised PoS taggers.

2.3.2 Supervised Part-of-Speech Tagging

As mentioned previously, the (Constituent Likelihood Automatic word-tagging System) CLAWS tagger was one of the first probabilistic tagger, taking advantage of the supervised tags of the Brown Corpus to improve its estimation of ambiguous sequences. The improved CLAWS4 assisted in the tagging of the BNC (Garside, 1996). This system was motivated by the observation that 80% of the actual applications of TAGGIT’s context frame rules only used one word in the immediate context (Beale, 1985). Accordingly, CLAWS uses a matrix of tag-tag transition probabilities, similar to the modern HMM approach (Beale, 1985). Unlike a HMM, CLAWS uses a conditional random field (CRF) to incorporate rough word-tag frequency markers in its dictionary, with “@” marking rare tags for a word and “%” for very rare tags, lowering the probability of paths through those tags by one half and one eighth. respectively (Beale, 1985). The system uses similar dictionary and suffix tests to assign a set of possible tags to each word, but augments a limited set of language-specific context frame tests with the transition probabilities (Garside, 1996). The transition probabilities and the frequency markers combine with the Viterbi algorithm choosing the most likely path through a sequence of ambiguous words (Garside, 1996). The resulting algorithm achieves around 96% accuracy.

The Penn Treebank was tagged with the PARTS tagger of Church (1988), which used a Hidden Markov Model (HMM) to model part-of-speech. The HMM generates the next tag according to the transition probability from the

last tag (or the last two tags in the case of a trigram HMM) and that tag outputs a word according to an emission probability. Many other subsequent part-of-speech taggers relied on the HMM, learning the transition and emission distributions either from supervised data or with a semi-supervised dictionary and the Baum-Welch algorithm, an instance of Expectation-Maximisation (Merialdo, 1993, Weischedel et al., 1993, Schütze and Singer, 1994, Brants, 2000).

Another approach was proposed by Brill (1994) to automatically learn transformation rules from a supervised corpus. The approach initialises the tags to the most frequent tag available for each word, and then performs a series of transformations that change one tag to another according to various context features (Brill, 1994). Each transformation is greedily chosen to minimise the number of errors relative to the supervised corpus (Brill, 1994). Once learned, the transformations can be reapplied in the same order they were learned to achieve competitive performance with contemporary approaches (Brill, 1994).

The state-of-the-art results in supervised PoS tagging were achieved by the Stanford Tagger of Manning (2011) and Shen et al. (2007). These taggers use linear and log-linear feature-based models to disambiguate tags according to a variety of orthographic and contextual features (Manning, 2011). Collobert et al. (2011) show close performance with a supervised neural-network and word-embedding features learned from a large unsupervised corpus. Manning (2011) achieved 97.3% accuracy and the inter-annotator agreement is about 98% (Baker, 1997). All of these techniques require a supervised corpus to achieve these results, and there are still many resource-poor languages and domains for which the supervised approach is infeasible.

2.3.3 Part-of-Speech Disambiguation

Considering the cost of manually annotating the large corpora used for supervised training³, many researchers in the 1990’s also experimented with unsupervised PoS tagging. Most of these “unsupervised” taggers assumed the availability of a lexicon of available tags, or tag-dictionary, for most words (Brill, 1995, Kupiec, 1992, Merialdo, 1993). These dictionaries simplified the inference problem by reducing ambiguity and evaluation by roughly defining the tagset. They are abstractions of the more complex lexica described by modern linguistics in section 2.2.3. The task of learning tags with a lexicon and an unlabelled corpus is referred to as part-of-speech *disambiguation* (Brill, 1995). The most popular approach modelled the generation of the corpus with an HMM and learned the maximum likelihood parameters through the Baum-Welch algorithm, now referred to as Expectation-Maximisation (EM) (Baum, 1972, Kupiec, 1992, Merialdo, 1993, Elworthy, 1994). To reduce the total number of parameters, Kupiec (1992) treated words with the same possible tag assignments, or *ambiguity class*, as equivalence classes, sharing emission probabilities⁴. On the other hand, Brill (1995) was able to adapt his rule-based transformation tagger to tag discrimination by replacing the minimum errors score with a one that prefers to reuse tags that have already been unambiguously assigned.

While the early part-of-speech disambiguation approaches achieved 95-96% accuracies, nearly matching the fully supervised counterparts, they assumed the existence of dictionaries of dubious quality. The tag-dictionaries themselves were extracted from the combined training and test sets, resulting in initial accuracies of 90% before training (Brill, 1995). Furthermore, Banko

³Marcus et al. (1993a) states that the proposal for the Penn Treebank estimated the total cost at \$10 million

⁴Kupiec (1992) gave the 100 most frequent words their own equivalence classes

and Moore (2004) found that the results of these taggers depended strongly on preprocessing the dictionary to remove all word-tag pairs that represent less than some threshold proportion of the total occurrences of that word. Ravi and Knight (2009) show that when these uncommon entries are not removed from the dictionary, the EM algorithm assigns them more evenly than the actual highly skewed distribution.

More recent research on the part-of-speech disambiguation task tends to focus on minimising the amount of supervised information necessary or mitigating the effects of a non-processed dictionary. Goldwater and Griffiths (2007) use a diluted dictionary representing the most frequent words. Haghighi and Klein (2006) learn from a small number of prototypical words for each part-of-speech. Garrette and Baldrige (2013), Garrette et al. (2013) learn taggers for low-resource languages with only two hours of annotation of a corpus or dictionary. Ravi and Knight (2009) use Integer Programming to eliminate rare or erroneous entries from the lexicon. Finally, Toutanova and Johnson (2008) model the generation of ambiguity classes to predict new ambiguity classes for unseen words.

2.3.4 Unsupervised Part-of-Speech Induction

Brown et al. (1992) takes a different approach from its contemporaries. Rather than part-of-speech disambiguation with a lexicon, Brown et al. (1992) performed fully unsupervised part-of-speech *induction*. Unsupervised part-of-speech induction groups word-tokens into unlabelled clusters representing syntactic categories. Brown et al. (1992) models part-of-speech with an HMM like Kupiec (1992) and Merialdo (1993). Brown et al. (1992) introduces the one-tag-per-type assumption because fully unsupervised induction has considerably more ambiguity than the disambiguation class. The one-tag-per-type

assumption means that all tokens of the same word-type are placed in the same cluster, or word-class. This approach initially places each word in its own class and then greedily merges classes to maximize the average mutual information of adjacent word-classes until it reaches the desired number of clusters (Brown et al., 1992).

Another method by Schütze (1993) also assigns one-tag-per-type, though it attempts to predict and discard ambiguous words that violate this assumption. This approach uses a series of clustering dimension reducing techniques for word context vectors to create a vector space that can be used as inputs and outputs for a neural network. First, four matrices representing the cooccurrence frequencies of the 5,000 most frequent word-types are combined into a 5,000x20,000 matrix. The 5,000 words are then mapped to vectors in 15 dimensional space via Singular Value Decomposition (SVD) (Schütze, 1993). The 5,000 words are combined into 500 clusters, 222 clusters from the 15 dimensional space and 278 high-frequency "closed-class words" (Schütze, 1993). These clusters are incorporated into a new 2,000x22,771 cooccurrence matrix of all the words that occur at least 100 times, which are then mapped to a 10 dimensional vector space again with SVD (Schütze, 1993). These vectors are used to train a birecurrent neural network, a combination of two recurrent neural networks: one of which predicts the context vector given the left word and context, the other does the same from the right. The output of the first 5,000 vectors from the network are clustered into 200 clusters, labelled with the two words closest to their centroid (Schütze, 1993). A word is assigned to the cluster with the highest correlation coefficient for its closest member to the network's output accumulated over all occurrences of the word. If no cluster scores above a certain threshold for a given word, that word is declared ambiguous and discarded. Schütze (1993) argues that these clusters represent

parts of speech, and uses human annotators to label the clusters according to which of first 5,000 words are in them.

The scepticism toward tag-dictionaries of the early 2000's revitalised the part-of-speech induction task, with Clark (2000) describing an HMM model for syntactic categories. Clark (2003) expands this model to include a morphology (or more accurately orthography) model. This model uses letter HMMs (compared to the word HMMs described before) for each cluster, such that the prior probability of a cluster is the product of the probability of that cluster's HMM generating each of the words within it (Clark, 2003). Clark (2003) recognized that a key advantage of unsupervised PoS induction over more supervised approaches is its language independence, this motivated his choice of letter HMMs over other possible morphology models. Like the approaches of Brown et al. (1992) and Schütze (1993), this approach takes the one-tag-per-type assumption, greatly improving the efficiency of inference Clark (2003). Inference in this approach is iterative, starting with the most frequent words in their own clusters, and another cluster for all of the remaining words (Clark, 2000). Then, each iteration calculates the change in likelihood for the overall model for moving each word from its current cluster to each other cluster, choosing the move that maximizes the likelihood (Clark, 2003).

While pragmatic in nature, the one-tag-per-type assumption is similar to the assumptions of the Structuralists. The result is word-classes that approximate general syntactic behaviour. The move to lexicon-based models in modern linguistics and the trend toward small tagsets in supervised PoS tagging are reactions to flaws in this approach. The one-tag-per-type assumption assigns the same word-class to different homograms, limits the ability to generalize over subcategories, and requires large numbers of classes to encode fine grammatical distinctions. In the spirit of eliminating this assumption, Clark (2003)

explored the possibility of allowing ambiguity with a simple HMM and the EM algorithm used in the discriminative approaches from section 2.3.3, finding that without any restrictions on the number of tags-per-type, the emission distribution allows too many tags to emit words of the same type.

Goldwater and Griffiths (2007) attempt to alleviate the tags-per-type problem of the traditional HMM noted by Clark (2003) with the introduction of non-parametric Bayesian priors. The approach of Goldwater and Griffiths (2007) used the non-parametric *Dirichlet process distributions*⁵ as priors to encourage sparsity of the transition and emission distributions. Goldwater and Griffiths (2007) used Gibbs sampling to approximate the marginal posterior distribution for the tag assignment for each individual token. A similar approach is used by Johnson (2007), performs Variational Bayes inference (an approximation of the posterior from a more easily computed lower bound) on the same model. Another solution to the tags-per-type sparsity problem relies on posterior-regularisation (Ganchev et al., 2010) to directly constrain the posterior distribution of the Bayesian HMM to the desired sparsity (Ganchev et al., 2009). Additionally, the system of Berg-Kirkpatrick et al. (2010) makes the HMM emission and transition distributions logistic to allow the incorporation of manually selected features like morphology.

State-of-the-art results were achieved by the *Pitman-Yor Hidden Markov Model* (PYP-HMM) of Blunsom and Cohn (2011). This model is the base of the work demonstrated in this thesis, and it will be explored in more detail in Chapter 3. The PYP-HMM extends the context of the HMM from a bigram HMM to a trigram HMM. A hierarchical non-parametric *Pitman-Yor Process* prior accounts for the sparsity of tag trigrams by smoothly backing off to smaller contexts. The PYP-HMM+LM is an extension to the PYP-HMM that

⁵examined in more detail in Chapter 3

incorporates morphology into the emission distribution based on a character language model, similar to the letter HMM of Clark (2003). However, the standard collapsed Gibbs sampler is unable to mix, getting trapped in local maxima. Therefore, Blunsom and Cohn (2011) used a type-based block Gibbs sampler, resorting to an approximation based on the flawed one-tag-per-type assumption to make calculating the posterior of this type sampler tractable.

2.4 Defining Part-of-Speech

Section 2.2 explored the history of part-of-speech and found that while the specific definitions and names have changed, part-of-speech has consistently been an important step in defining a language’s grammatical structure. While the traditional parts of speech conflate type-level word classes with token-level grammatical categories, modern linguists distinguish two related but distinct categories: grammatical categories that specify a word’s grammatical function and word-classes that define which of these functions a word might fulfil. Section 2.3 showed that computational approaches to part-of-speech have consistently treated them as grammatical categories. Therefore, this thesis defines part-of-speech as word-level syntactic categories revealing the syntactic position and function of a word-token.

Chapter 3

Hierarchical Non-Parametric Models and the PYP-HMM

3.1 Introduction

As explained in Chapter 2, unsupervised PoS induction results considerably improved with the development of Bayesian non-parametric models of part-of-speech. The work in this thesis is an extension of those approaches, based on the successful PYP-HMM model of Blunsom and Cohn (2011). This chapter explores the benefits and techniques of Bayesian non-parametric inference. The discussion will ground the explanations by applying these frameworks to the problem of unsupervised part-of-speech induction with a Hidden Markov Model.

Section 3.2 starts with an overview of the bigram HMM in non-Bayesian part-of-speech induction. This is followed by section 3.3 motivating and explaining the techniques of Bayesian inference. Then, section 3.4 covers the Dirichlet and Pitman-Yor Processes, common non-parametric priors, as well as their hierarchical extensions. Finally, section 3.5 concludes with the de-

scription of the Hierarchical Pitman-Yor Process Hidden Markov Model of part-of-speech (PYP-HMM), which will provide a base for the experiments and extensions introduced in the rest of this thesis.

3.2 The Part-of-Speech Hidden Markov Model

Since the early 1990's, the Hidden Markov Model has been a popular model of part-of-speech (Kupiec, 1992, Merialdo, 1993). The HMM captures Harris's (1946) idea of substitutability by having the part-of-speech tag¹ determine the word assigned to it and having the tag itself determined by its context. Furthermore, the HMM is a *generative model*, jointly modelling both the observed words, $\mathbf{w}$, and the latent tags $\mathbf{t}$. The generative nature of the HMM makes it easily adaptable to the task of unsupervised induction.

A bigram HMM has a multinomial tag-tag transition distribution, $B_j, 1 \leq j \leq K$ for each of K tags, with parameters Θ_j^B . Each tag also has another multinomial tag-word emission distribution, $E_j, 1 \leq j \leq K$, with multinomial parameters Θ_j^E . Each tag is generated sequentially given the previous tag in the sentence, so the probability of tag t_n following tag t_{n-1} is $P(t_n|t_{n-1}, \Theta) = \Theta_{t_{n-1}, t_n}^B$. The tags each generate the word assigned to them, so for word w_n and tag t_n , $P(w_n|t_n, \Theta) = \Theta_{t_n, w_n}^E$. The conditional dependencies defined by these probabilities are written as

$$t_n|t_{n-1}, B \sim B_{t_{n-1}} \quad (3.1)$$

$$w_n|t_n, E \sim E_{t_n} \quad (3.2)$$

Where for each tag t , B_t is a multinomial distribution over all other tags and

¹For the sake of brevity, part-of-speech and tag will be used interchangeably from now on

E_t is a multinomial distribution over the vocabulary. The parameters, Θ_t^B , of the transition multinomial, B_t , are vectors of probabilities for transitions from t to each of the K other tags, so $|\Theta_t^B| = K$, $\sum_{k=1}^K \Theta_{t,k}^B = 1$, and $\Theta_{t,k}^B \geq 0$ for $1 \leq k \leq K$. The parameters, Θ_t^E , of the emission multinomial, E_t , are probability vectors for the emission from tag t of each of the $|\mathbf{V}|$ words in the vocabulary, $\mathbf{V}$, such that $|\Theta_t^E| = |\mathbf{V}|$ and $\sum_{v=1}^{|\mathbf{V}|} \Theta_{t,v}^E = 1$. Placing a sentence boundary tag, $t_0 = t_{N+1} = \$$, before and after each sentence allows them to be treated as one long string, so that the likelihood of the sequence of all sentences, $\mathbf{w}$, and their tags, $\mathbf{t}$, with $|\mathbf{w}| = N$ is:

$$P(\mathbf{t}, \mathbf{w} | \Theta) = P(t_0 | \Theta) \prod_{n=1}^{N+1} P(t_n | t_{n-1}, \Theta) P(w_n | t_n, \Theta) \quad (3.3)$$

In the fully supervised case, both $\mathbf{w}$ and $\mathbf{t}$ are observed, and inference finds the parameters, Θ , that maximise the likelihood function $P(\mathbf{w}, \mathbf{t} | \Theta)$ from equation (3.3). In this case, the Θ s that maximise the likelihood are simply the observed tag-tag and tag-word frequencies. However, these frequencies represent only the transitions and emissions observed in the training data, and these data are inherently incomplete. This means that rare tag-tag and tag-word bigram frequencies are not representative of the “true” probabilities. Furthermore, unseen words and transitions have frequencies of zero, preventing the model from making predictions on new test data.

This problem is traditionally solved with *smoothing* techniques, connecting rare or unseen frequencies with the more common ones. For example, add-one smoothing adjusts the frequencies as if each possible occurrence was observed once more than it actually was. Alternatively, low-frequency estimates can be replaced by “backing-off” to a representative, more frequent occurrence. For example, the bigram transitions can back-off to unigram probabilities,

so that the model prefers common tags when it encounters rare transitions. This approach ties various parameters to each other reflecting some baseline understanding of how the model should generally behave.

Once the optimal parameters Θ are found, they are used to predict the tags $\tilde{\mathbf{T}}$ assigned to the new sentences $\tilde{\mathbf{w}}$. These tags are the optimal solutions to the probability $P(\tilde{\mathbf{t}}|\tilde{\mathbf{w}}, \Theta)$, calculated with the application of Bayes' rule:

$$P(\tilde{\mathbf{t}}|\tilde{\mathbf{w}}, \Theta) \propto P(\tilde{\mathbf{w}}|\tilde{\mathbf{t}}, \Theta)P(\tilde{\mathbf{t}}|\Theta) \quad (3.4)$$

These tag assignments are traditionally found either through application of the Viterbi algorithm, which finds the most likely sequence of tags for each sentence, or by choosing the marginally most likely tag as calculated by the forward-backward algorithm.

Without supervision, the tags, $\mathbf{t}$, are not observed so the likelihood, $P(\mathbf{w}|\Theta)$ marginalizes them out and the parameters must be estimated from the words alone. In this case, the *Maximum Likelihood Estimate* (MLE) of the parameters can be found according to the Baum-Welch algorithm (Baum, 1972), a variant of Expectation-Maximization designed for HMMs. The algorithm alternates between calculating the expected frequencies of the transition and emission data via the forward-backward algorithm and hill climbing the parameters to maximize the likelihood of those frequencies. This approach is guaranteed to find the parameters of a local maximum of the likelihood function.

As Clark (2003) found, this approach struggles because it evenly distributes the probability over the ambiguities, allowing many tags to transition to each other and many different tags to emit the same word. This is because the parameters are not tied together as they are in the smoothed, supervised case.

The Bayesian solution to this problem is to incorporate a distribution over the parameters called a prior. Furthermore, in unsupervised part-of-speech induction, the training data is the test data, so the goal is to find the most likely tags over all of the parameters. The Bayesian prior allows inference to integrate over the possible parameters, marginalizing them out and instead finding the marginal likelihood of the tags. These concepts are explored in the next section.

3.3 Bayesian Inference

The goal of Bayesian inference is twofold. The first goal is to encode what is certain about the structure of the parameter space with a prior distribution over it. The second goal is to marginalize the parameters out entirely, focusing only on those latent values that are of interest.

Bayesian inference assigns a *prior* probability, $P(\Theta|\boldsymbol{\alpha})$ to the parameters of the model. This allows the decomposition of the *posterior* distribution of the parameters, $P(\Theta|\boldsymbol{w}, \boldsymbol{\alpha})$ given the observed data according to Bayes' Rule:

$$P(\Theta|\boldsymbol{w}, \boldsymbol{\alpha}) \propto P(\boldsymbol{w}|\Theta, \boldsymbol{\alpha})P(\Theta|\boldsymbol{\alpha}) \quad (3.5)$$

The parameters which maximize $P(\Theta|\boldsymbol{w}, \boldsymbol{\alpha})$ in equation (3.5) are the *Maximum A Posteriori* (MAP) estimate of the parameters. As with maximum likelihood estimation, the inference of the MAP parameters marginalizes over the possible tag assignments. Additionally, the MAP parameters can then be used to predict the tag assignments according to their marginal probabilities or the probability of the entire sequence with the Viterbi algorithm.

A natural choice of prior distribution for the transition and the emission distributions is the Dirichlet distribution. Both the transition and emission

distributions are multinomial distributions over the tags and vocabulary, respectively. The Dirichlet distribution, $\mathcal{D}(\boldsymbol{\alpha})$ with parameter vector $\boldsymbol{\alpha}$ of size K , with $\alpha_i > 0, 1 \leq i \leq K$, is a distribution over multinomial distributions with K categories. The Dirichlet is the *conjugate prior* of the multinomial distribution, meaning that after N draws, $d_1, \dots, d_N$, from the multinomial distribution M with K categories defined as:

$$M | \boldsymbol{\alpha} \sim \mathcal{D}(\boldsymbol{\alpha}) \quad (3.6)$$

$$d_1, \dots, d_N | M \sim M \quad (3.7)$$

With the term $M | \boldsymbol{\alpha} \sim \mathcal{D}(\boldsymbol{\alpha})$ indicating that the multinomial, or its parameters, were drawn from the Dirichlet prior $\mathcal{D}(\boldsymbol{\alpha})$. If $\mathbf{c} = (c_1, \dots, c_K)$ is the vector of counts for each class of the draws, $\mathbf{d}$, then

$$M | \mathbf{c}, \boldsymbol{\alpha} \sim \mathcal{D}(\mathbf{c}_1 + \alpha_1, \dots, \mathbf{c}_K + \alpha_K) \quad (3.8)$$

Therefore, the posterior distribution of a multinomial distribution with a Dirichlet prior is also a Dirichlet distribution. This leads to a new, Bayesian version of the HMM from section 3.2 based on Dirichlet priors. This Dirichlet HMM (DHMM) has each of the multinomial transition and emission distributions generated from a Dirichlet prior. These priors have transition parameters $\boldsymbol{\alpha}_j^B, 1 \leq j \leq K$ and emission parameters $\boldsymbol{\alpha}_j^E, 1 \leq j \leq K$ for K tags in the

tagset. The DHMM is thus defined as:

$$B_j | \boldsymbol{\alpha}_j^T \sim \mathcal{D}(\boldsymbol{\alpha}_j^T) \quad (3.9)$$

$$E_j | \boldsymbol{\alpha}_j^E \sim \mathcal{D}(\boldsymbol{\alpha}_j^E) \quad (3.10)$$

$$t_n | t_{n-1}, B \sim B_{t_{n-1}} \quad (3.11)$$

$$w_n | t_n, E \sim E_{t_n} \quad (3.12)$$

Where, as before, the t_n and w_n are each generated sequentially. The prior probability used to calculate the posterior in equation (3.5), $P(\Theta | \boldsymbol{\alpha})$, is the probability of $\mathcal{D}(\boldsymbol{\alpha})$ generating the multinomial distributions with parameters Θ .

These approaches are still based around finding point estimates for the optimal values of Θ , which are then used to predict the tags $\mathbf{t}$. The Bayesian alternative is to simply predict the most likely tags, $\mathbf{t}$, given the words, $\mathbf{w}$, over all the possible values of the parameters, Θ . This is achieved by maximizing over the *posterior predictive distribution*, $P(\mathbf{t} | \mathbf{w}, \boldsymbol{\alpha})$, which is the posterior distribution marginalized over Θ :

$$\text{argmax}_{\mathbf{t}} P(\mathbf{t} | \mathbf{w}, \boldsymbol{\alpha}) = \int_{\Theta} P(\mathbf{t} | \mathbf{w}, \Theta, \boldsymbol{\alpha}) P(\Theta | \boldsymbol{\alpha}) d\Theta \quad (3.13)$$

This integral is generally not solvable analytically and must be approximated. Variational Bayesian methods approximate this integral by exactly optimising a simpler approximation of the posterior (Beal, 2003). The alternative approach explored in this thesis is to use Markov chain Monte Carlo (MCMC) methods, like Gibbs sampling, to approximate the integral with a series of samples from the true posterior (Neal, 2003, Goldwater and Griffiths, 2007).

Markov chain Monte Carlo methods generate a sequence of samples such

that they form a Markov chain with the target distribution as its equilibrium distribution. For integration with MCMC, the integrand is calculated at each step in a random walk, and the average value of the integrand approaches the true value as the number of steps increases without bound.

Gibbs sampling is one MCMC technique for joint distributions where each step samples a subset of the variables from the posterior conditioned on the remaining, unchanged variables. A simple implementation of Gibbs sampling for the DHMM model would be to sample each of the tag assignments, t_n individually, conditioned on all of the other tag assignments, $\mathbf{t}_{-n}$, words, $\mathbf{w}$ and parameters, Θ, α :

$$P(t_i|\mathbf{t}_{-n}, \mathbf{w}, \Theta, \alpha) = P(t_n|t_{n-1}, t_{n+1}, w_n, \Theta) \quad (3.14)$$

$$\propto P(t_n|t_{n-1}, \Theta)P(t_{n+1}|t_n, \Theta)P(w_n|t_n, \Theta) \quad (3.15)$$

$$= \Theta_{t_{n-1}, t_n}^B \Theta_{t_n, t_{n+1}}^B \Theta_{t_n, w_n}^E \quad (3.16)$$

After sampling each tag assignment, the sampler would use the frequency counts to sample a new set of parameters according to the posterior shown in equation (3.8).

Alternatively, the tag t_n can be sampled jointly with the relevant transition and emission parameters, Θ^B and $\Theta_{t_n}^E$, respectively. The resulting probability distribution depends only on the counts of the variables other than those at the index of the tag being sampled. In fact, those tags are sufficient to marginalize out the parameters, Θ , entirely. This integral has an analytical solution and is referred to as *collapsed Gibbs sampling*. This approach replaces the factorized

probabilities in equation (3.14) with the marginalized probabilities:

$$P(t_n|t_{n-1}, \mathbf{c}_{t_{n-1},-}^{B-n}, \boldsymbol{\alpha}) \propto c_{t_{n-1},t_n}^{B-n} + \alpha_{t_{n-1},t_n}^B \quad (3.17)$$

$$P(t_{n+1}|t_n, \mathbf{c}_{-,t_n}^{B-n}, \boldsymbol{\alpha}) \propto c_{t_n,t_{n+1}}^{B-n} + \alpha_{t_n,t_{n+1}}^B \quad (3.18)$$

$$P(w_n|t_n, \mathbf{c}_{-,w_n}^{E-n}, \boldsymbol{\alpha}) \propto c_{t_n,w_n}^{E-n} + \alpha_{t_n,w_n}^E \quad (3.19)$$

Where $\mathbf{c}^{B-n}$ and $\mathbf{c}^{E-n}$ are the tag and word bigram counts, respectively, of the remaining variables not at index n .

While this approach marginalizes over the parameters Θ and allows encoding of the uncertainty of Θ , it ultimately pushes that uncertainty on to the Dirichlet parameters $\boldsymbol{\alpha}$. There are as many parameters in $\boldsymbol{\alpha}$ as there are in Θ . One solution is to set all of the values of the transition priors to some scalar α^B , so $\alpha_{i,j}^B = \alpha^B, 1 \leq i, j \leq K$, and all of the emission parameters to another, α^E , so $\alpha_{i,v}^E = \alpha^E, 1 \leq i \leq k$ and $1 \leq v \leq |\mathbf{V}|$. This prior is known as the symmetric Dirichlet prior, $\mathcal{D}(K, \alpha)$, with parameter K dictating the number of classes in the resultant multinomial and concentration parameter $\alpha > 0$, such that

$$\mathcal{D}(K, \alpha) = \mathcal{D}(\underbrace{\alpha, \dots, \alpha}_{K \text{ times}}) \quad (3.20)$$

From equation (3.8), it is clear that the add-one smoothing from section 3.2 is equivalent to the posterior distribution of a symmetric Dirichlet prior with $\alpha_j = 2, 1 \leq j \leq K$. However, the HMM priors should encode the sparsity and skewness of part-of-speech transitions and emissions. Setting some of the values for each $\boldsymbol{\alpha}_t^T$ would bias toward sparse transitions, but it would favour specific transitions arbitrarily because the parameters are ordered. On the other hand, section 3.4 shows how non-parametric priors encode limited knowledge about a domain when the number or order of parameters is unknown.

3.4 Non-Parametric Models

The goal of non-parametric models is to create a prior over a model's parameters without knowing the number or order of those parameters. This allows modelling intricate distributions while defining only a limited number of *hyperparameters*. The symmetric Dirichlet prior, $\mathcal{D}(K, \alpha)$, described at the end of section 3.3 fails at these requirements because the number of parameters must be specified.

Instead, the non-parametric Dirichlet and Pitman-Yor Processes are generalizations of the Dirichlet prior to a potentially infinite number of classes. Developed to model biological populations, these priors generate size-based permutations of an infinite number of classes. Alternatively, they represent the proportion of a species in the order that it appears (Pitman, 1995).

3.4.1 Dirichlet Processes

A Dirichlet process, $\text{DP}(\alpha, P_0)$ is a generalisation of the Dirichlet prior over a potentially infinite number of classes (Ferguson, 1973). The parameter $\alpha > 0$ is similar to the concentration parameter in the symmetric Dirichlet prior. The second parameter, P_0 , is referred to as the *base distribution*, and it is a prior distribution over the support of multinomial distributions drawn from the Dirichlet process.

The base distribution is important because the Dirichlet process is exchangeable and the parameters of multinomial distributions drawn from it do not have a set order. Pitman (1995) demonstrates that the parameters of a multinomial drawn from a DP without a base distribution are equivalent to the proportions of each class from a random partitioning of the integers, ordered according to the smallest integer in each class. With this definition in mind,

consider the transition distributions of two tags, B^{t^1} and B^{t^2} , both drawn from a Dirichlet process prior. In this example, the first parameter of B^{t^1} is the parameter for the first class to come after tag t^1 , and similarly for B^{t^2} and t^2 . However, the first tag after t^1 is not necessarily the same as the first tag after t^2 , and there is no way to distinguish which of these tags is which. However, by assigning P_0 to be a distribution over each of the tags, each of the classes in the partition can be assigned to a tag according to the prior probability of that tag, P_0 .

The DP can be defined according to a stick-breaking model. In this model, a stick of size 1 is broken into two smaller pieces, a class, $k \sim P_0$, is drawn from the base distribution, and the first parameter is the size of the first piece multiplied by the base probability of the class it is assigned to. Subsequent parameters are determined by breaking the remainder from the previous parameter's break. Formally, the parameters, $\Theta \sim \text{DP}(\alpha, P_0)$ are assigned according to:

$$k_i | P_0 \sim P_0 \quad (3.21)$$

$$\beta'_i | \alpha \sim \text{Beta}(1, \alpha) \quad (3.22)$$

$$\beta_i = \beta'_i \prod_{j=1}^{i-1} (1 - \beta'_j) \quad (3.23)$$

$$\Theta_i = \beta_i \quad (3.24)$$

Where $\text{Beta}(a, b)$ is the conjugate prior of the binomial distribution. Of course, equation (3.21) works on the simplifying assumption that all of the classes k_i are different. In situations where P_0 is a multinomial distribution over a finite number of discrete classes², this assumption is faulty. Instead, the parameters

²This is the case in both the transition and emission distributions of the HMM

must sum over all of sticks assigned to the same class:

$$\Theta_k = \sum_{i=1}^{\infty} \beta_i \delta_k(k_i) \quad (3.25)$$

Where δ is an indicator function such that if $k_i = k$ then $\delta_k k_i = 1$ and otherwise $\delta_k k_i = 0$.

While the infinite sum in equation (3.25) is not constructive, there is an equivalent formulation of the Dirichlet process based on the predictive posterior distribution. The *Chinese Restaurant Process* (CRP) formulation of the Dirichlet process considers a sequence of random variables, $d_1, \dots, d_n$, representing customers entering a Chinese restaurant with an infinite number of tables. The first customer, d_1 sits at the first table $l_1 = 1$. A new customer, d_{n+1} , chooses to sit at a table, l_{n+1} , representing the clusters. Any finite subsequence of such a sequence could have been generated from a multinomial with a Dirichlet prior over unknown number of categories, with a similar posterior distribution. Thus the new customer either sits at one of the L^n existing tables, $l_{n+1} \leq L^n$, or chooses to sit at a new table, $l_n = L^n + 1$, with probability:

$$P(l_{n+1} = l | \mathbf{l}^n) = \begin{cases} \frac{c_l^n}{n+\alpha} & 1 \leq l \leq L \\ \frac{\alpha}{n+\alpha} & l = L^n + 1 \end{cases} \quad (3.26)$$

here c_l^n is the number of the n previous customers sitting at table l and $\mathbf{l}^n$ is the seating arrangement of the n previous customers.

The formulation from equation (3.26) does not account for the base distribution, P_0 . Each customer at a table, l_{n+1} shares a dish, d_{n+1} , representing the classes sampled from P_0 and the value of the random variable. Rather than calculating the infinite sum from the stick-breaking formulation in equation (3.25), the CRP only needs to track the number of customers sharing the

same dish:

$$P(d_{n+1} = k | \mathbf{d}^n, P_0) = \frac{c_k^n + \alpha P_0(k)}{n + \alpha} \quad (3.27)$$

where $\mathbf{d}^n$ is the history of dishes chosen by the previous n customers and c_k^n is the number of the previous n customers being served dish k . Equation (3.27) is essentially the combined probability of the new customer sitting at a table already serving dish k and the probability of the customer sitting at a new table and ordering k .

The exchangeability of samples from a DP prior allows a simple collapsed Gibbs algorithm for inference. Using the CRP formulation, a sequence of variables has the same probability regardless of its arrangement. Therefore the next variable to be sampled can be moved to the end of a sequence and then taken out and resampled. This approach is almost exactly the same as the collapsed Gibbs sampler described in equation (3.17).

3.4.2 Pitman-Yor Processes

Pitman and Yor (1997) describes a two-parameter generalisation of the Dirichlet process, called the two-parameter Poisson-Dirichlet process, or the *Pitman-Yor process* (PYP). In addition to the *strength* parameter b and the base distribution P_0 , the PYP has a *discount* parameter a . When $a = 0$, the PYP reduces to a DP, $\text{DP}(\alpha, P_0) = \text{PYP}(0, \alpha, P_0)$. When $a > 0$ and $b > -a$, however, $\text{PYP}(a, b, P_0)$ generates long-tailed power-law distributions found in word frequencies (Zipf, 1932) and other linguistic phenomena (Goldwater et al., 2011).

The Pitman-Yor process has both a stick-breaking and CRP formulation similar to that of the Dirichlet process. Unlike the DP, the stick-breaking formulation of $\text{PYP}(a, b, P_0)$ incorporates the order of the generation of the β'

values, modifying equations (3.21) and (3.25) as follows:

$$k_i | P_0 \sim P_0 \quad (3.28)$$

$$\beta'_i | \alpha \sim \text{Beta}(1 - a, b + ia) \quad (3.29)$$

$$\beta_i = \beta'_i \prod_{j=1}^{i-1} (1 - \beta'_j) \quad (3.30)$$

$$\Theta_k = \sum_{i=1}^{\infty} \beta_i \delta_k(k_i) \quad (3.31)$$

Similarly, the CRP formulation of PYP(a, b, P_0) accounts for the number of tables currently occupied, L^n :

$$P(l_{n+1} = l | \mathbf{l}^n) = \begin{cases} \frac{c_l^n - a}{n+b} & 1 \leq l \leq L \\ \frac{L^n a + b}{n+b} & l = L^n + 1 \end{cases} \quad (3.32)$$

This leads to the posterior probability of class k given the base distribution P_0 :

$$P(d_{n+1} = k | \mathbf{d}^n, P_0) = \frac{c_k^n - c_{L,k}^n + (L^n a + b) P_0(k)}{n + b} \quad (3.33)$$

where $c_{L,k}^n$ is the number of tables serving dish k and, as in equation (3.27), c_k^n is the number of customers served dish k .

Equations (3.30) and (3.33) show how the discount parameter a results in a longer tail of classes, as a gets larger and the number of tables increases, a new customer is more likely to choose to sit at a new table. The strength parameter b , effects the first few tables, but diminishes in relevance as the number of customers increases. Overall, the DP and PYP have a clustering effect, assigning a higher predictive probability to larger classes. This effect ultimately results in the desired sparse distributions.

3.4.3 Hierarchical Non-Parametric Models

While the standard PYP results in sparse distributions similar to those found in linguistics, there is still a problem when the data are not rich enough to learn rare frequencies. Increasing the context of the part-of-speech model from a bigram to a trigram HMM complicates inference because most trigrams are rare. The standard smoothing approach discussed in section 3.2 uses a back-off distribution with less context to fill in the gaps. An even better solution interpolates between the full distribution and the back-off distribution to connect related parameters with similar contexts.

Hierarchical Non-parametric models are the Bayesian equivalent to back-off smoothing (Teh, 2006a). A hierarchical PYP uses another PYP as its base distribution, backing off smoothly to the base when there is less data. For example, the PYP-HMM discussed in the next section generates a three-level hierarchy from trigrams to bigrams to unigrams for the transition distribution prior.

Hierarchical PYPs have a relatively straightforward CRP formulation that links each child restaurant to its parent. As can be seen from equation (3.33), the base-distribution of a PYP is only relevant when a customer sits at a new table in the CRP formulation. Therefore, when a customer chooses a new table in the child CRP, a new customer enters the parent CRP. When the parent customer determines which dish they are served, that is the dish that the child customer orders for her table.

3.5 The PYP-HMM

This section specifies the hierarchical PYP part-of-speech model that forms the base of the experiments and extensions examined in this thesis. This model

was first introduced by Blunsom and Cohn (2011) and achieved state-of-the-art results in several languages.

The PYP-HMM model of PoS applies a hierarchical Pitman-Yor process (PYP) prior to a trigram hidden Markov model (HMM) to jointly model the distribution of a sequence of latent word tags, $\mathbf{t}$, and word tokens, $\mathbf{w}$. The joint probability defined by the transition, $P(t_l|t_{n-1}, t_{n-2}, \Theta)$, and emission, $P(w_n|t_n, \Theta)$, distributions of a trigram HMM is

$$P(\mathbf{t}, \mathbf{w} | \Theta) = \prod_{n=1}^{N+1} P(t_n|t_{n-1}, t_{n-2}, \Theta) P(w_n|t_n, \Theta)$$

where $N = |\mathbf{t}| = |\mathbf{w}|$ and the special tag \$ is added to denote the sentence boundaries. The model defines the transition and emission distributions to be multinomial:

$$\begin{aligned} t_n | t_{n-1}, t_{n-2}, T & \sim T_{t_{n-1}, t_{n-2}} \\ w_n | t_n, E & \sim E_{t_n} \end{aligned}$$

The PYP-HMM draws the above multinomial distributions from a hierarchical Pitman-Yor Process prior. The hierarchical prior can be intuitively understood to smooth the trigram transition distributions with bigram and unigram distributions in a similar manner to an ngram language model (Teh, 2006b). This backoff structure greatly reduces sparsity in the trigram distributions and is

achieved by chaining together the PYPs through their base distributions:

$$\begin{aligned}
T_{ij}|a^T, b^T, B_i &\sim \text{PYP}(a^T, b^T, B_i) \\
B_i|a^B, b^B, U &\sim \text{PYP}(a^B, b^B, U) \\
U|a^U, b^U &\sim \text{PYP}(a^U, b^U, \text{Uniform}). \\
E_i|a^E, b^E, C &\sim \text{PYP}(a^E, b^E, C_i),
\end{aligned}$$

where T_{ij} , B_i , and U are trigram, bigram, and unigram transition distributions respectively and C_i is either a uniform distribution (PYP-HMM) or a bigram character language model distribution (PYP-HMM+LM, intended to model basic morphology). The character language model of the PYP-HMM+LM is another hierarchical PYP over the sequence of characters in a word. It is a bigram Markov model where the states are the observed characters and each character is dependent on the previous one:

$$w_{nm}|w_{nm-1}, t_n, C \sim C_{t_n w_{nm-1}} \quad (3.34)$$

$$C_{im}|a^C, b^C, D_i \sim \text{PYP}(a^C, b^C, D_i) \quad (3.35)$$

$$D_j|a^D, b^D \sim \text{PYP}(a^D, b^D, \text{Uniform}) \quad (3.36)$$

Where m is an index over the characters in a word and a representation of the character itself and the distribution Uniform is uniform over the set of all characters.

Draws from the posterior of the hierarchical PYP can be calculated with a variant of the Chinese Restaurant Process (CRP) called the Chinese Restaurant Franchise (CRF) (Teh, 2006b, Goldwater et al., 2006). In the CRP analogy, each latent variable (tag) in a sequence is represented by a customer entering a restaurant and sitting at one of an infinite number of tables. A

customer chooses to sit at a table in a restaurant according to the probability in equation (3.32). All customers at a table share the same dish, representing the value assigned to the latent variables. When customers sit at an empty table, a new dish is assigned to that table according to the base distribution of the PYP. To expand the CRP analogy to the CRF for hierarchical PYPs, when a customer sits at a new table, a new customer enters the restaurant representing the PYP of the base distribution.

3.5.1 Inference with the PYP-HMM

Blunsom and Cohn (2011) explored two Gibbs sampling methods for inference with the PYP-HMM model. The first individually samples tag assignments for each token. The second employs a tactic shown to be effective by earlier works by constraining inference to only one tag per word type (PYP-1HMM) (Brown et al., 1992, Lee et al., 2010). Because marginalizing over all possible table assignments for more than a single tag is intractable, this inference must be approximated. Blunsom and Cohn (2011) approximate the PYP-1HMM tag assignment probabilities for a particular sample according to heuristic fractional table counts. Additionally, the hyperparameters a and b for each PYP distribution are inferred and sampled between iterations, with a slice sampler and Beta and Gamma priors for a and b , respectively. Specific details of this model and associated samplers can be found in (Blunsom and Cohn, 2011).

Chapter 4 presents a principled SMC based sampler that allows for the simultaneous sampling of all the tags associated with a given word type without resorting to heuristics. This type based sampler achieves near state-of-the-art performance without the requirement that all words of a given type share the same tag. This one-tag-per-type assumption is clearly false for syntactic categories (e.g. *sample* as a verb and a noun), thus its elimination is a step on

the path to high performance unsupervised PoS induction.

Chapter 4

Unsupervised Bayesian Part of Speech Inference with Particle Gibbs

Chapter 4 introduces the basic Sequential Monte Carlo algorithm and describes the implementation of two Particle Gibbs samplers for the PYP-HMM model described in Chapter 2. These samplers are compared with those used in Blunsom and Cohn (2011) using a variety of metrics. Much of this research is based on work originally published in Dubbin and Blunsom (2012).

4.1 Introduction

Modern research is steadily revealing more of the subtle structure of natural language to create increasingly intricate models. Recent advances in Bayesian non-parametrics have been employed by Computational Linguistics researchers to create effective unsupervised models of the latent structure text (Goldwater

et al., 2006, Liang et al., 2007, Cohn et al., 2009, Blunsom and Cohn, 2011). Chapter 3 describes the motivation and application of these models. These models predominantly make use of the Dirichlet Process (DP), and its generalization the Pitman-Yor Process (PYP), in part due to the ease of deriving a collapsed Gibbs sampler for its inference. Unfortunately, this ease of inference comes at a cost; collapsed Gibbs samplers mix poorly due to the strong global dependencies present in the conditional distributions sampled from Liang et al. (2010).

Chapter 2 suggests that latent part-of-speech induction is one application with such non-local correlations. The correlations between PoS tags of word-tokens of the same word-type make part-of-speech induction a useful test for inference techniques.

Sequential Monte Carlo methods, like particle filters, are particularly well suited to estimating tightly coupled distributions (Andrieu et al., 2010). Particle filters sample sequences of latent variable assignments by concurrently generating several representative sequences consistent with a model’s conditional probability distribution. The sequential nature of the sampling simplifies inference by ignoring ambiguous correlations with future latent variables at the cost of sampling the sequence multiple times. Other applications of particle filters in computational linguistics generally focus on the online nature of SMC (Canini et al., 2009, Borschinger and Johnson, 2011). On the other hand, this chapter demonstrates that batch applications benefit from the power of SMC to generate samples from tightly coupled distributions that would otherwise need to be approximated. Furthermore, the time cost of the additional samples generated by SMC can be mitigated by generating them in parallel.

This chapter focuses on the task of unsupervised part-of-speech induction, which aims to label tokens in a corpus with the syntactic role they play. In

particular it describes and evaluates novel SMC inference algorithms for the Pitman-Yor Hidden Markov Model (PYP-HMM) first proposed in (Blunsom and Cohn, 2011). This model represents the state-of-the-art for unsupervised PoS induction, but the complex conditional dependencies present in the posterior led the authors of Blunsom and Cohn (2011) to resort to a heuristic inference strategy and unrealistic restrictions on the model to achieve their highest reported results.

Section 4.2 starts by introducing PYP-HMM model and its previously proposed inference algorithms the PYP-HMM itself and non-parametric models in general are explored in more detail in Chapter 3. Section 4.3 introduces the Sequential Importance Sampling (SIS) algorithm, a basic SMC method that generates samples from the model’s posterior. Using this approach Section 4.3 describes two novel inference algorithms for the PYP-HMM: a simple sentence-based block sampler (4.3.1) and a more complicated type-based sampler (4.3.2). Section 4.4 evaluates these algorithms, analysing their behaviour in comparisons to the previously proposed state-of-the-art approaches. In summary, this chapter shows that our SMC based algorithms are able to improve inference, finding higher probability modes, and task specific accuracies. The results indicate two clear areas for further improvements with semi-supervision or taking advantage of the Particle Gibbs sampler to extend the PYP-HMM model.

4.2 The Pitman-Yor Hidden Markov Model

4.3 Sequential Monte Carlo

Sequential Monte Carlo was introduced in 1993 as a Bayesian estimator for signal processing problems with strong non-linear conditional dependencies (Gordon et al., 1993). Since then, SMC methods have been adopted by many fields, including statistics, biology, economics, etc. (Jasra et al., 2008, Beaumont, 2003, Fernandez-Villaverde and Rubio-Ramirez, 2007). The SMC approach is the probabilistic analogue of the beam search heuristic, where the beam width can be compared to the number of particles and pruning is analogous to resampling. Two key advantages to SMC are its online nature and its ability to generate samples for sequences with non-local correlations.

The basic SMC approach serves as the basis for several variants. Many SMC implementations resample the population of particles from the existing population to minimize the effect of increasing sample variance with increasing sequence length (Kitagawa, 1996). Particle smoothing variants of SMC reduce the relative variance of marginals early in the sequence, as well improving the diversity of the final sample (Fearnhead et al., 2008). Particle Markov chain Monte Carlo (PMCMC) formally augments classic Markov chain Monte Carlo (MCMC) approaches, like Gibbs sampling, with samples generated by particle filters (Andrieu et al., 2010).

While MCMC approximates a distribution as the average of a sequence of samples taken from the posterior of the distribution, SMC approximates a distribution as the importance weighted sum of several sequentially generated samples, called particles. This section describes two SMC samplers that jointly sample multiple tag assignments: a sentence based block sampler (**sent**) and a word-type based block sampler (**type**). The basics of particle filtering are out-

lined below, while the implementation specifics of the **sent** and **type** particle filters are described in sections 4.3.1 and 4.3.2, respectively. The hyperparameters are inferred with a slice sampler as mentioned in Chapter 3 and explained by Blunsom and Cohn (2011).

SMC can be understood as a probabilistic analogue of the beam search heuristic. SMC stores P sequences, analogous to beam width, and extends each incrementally according to a proposal distribution q_n , similar to the heuristic cost function in beam search. Many particle filtering implementations also include a resampling step which acts like pruning by reducing the number of unlikely sequences.

I implemented Sequential Importance Sampling (SIS), detailed by Doucet and Johansen (2009), to approximate joint samples from the sentence and word-type distributions. This approach approximates a target distribution, $\pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n}$, of the sequence, $x_{1:n}$, of n random variables, that is $\gamma_n(x_{1:n})$ calculates the unnormalized density of $x_{1:n}$.

SIS initialises each particle $p \in [1, P]$ by sampling from the initial proposal distribution $q_1(x_1^p)$, where x_1^p is the value assigned to the n -th latent variable for particle p . The algorithm then sequentially extends each particle according to the conditional proposal distribution $q_n(x_n^p | x_{1:n-1}^p)$, where $x_{1:n}^p$ is the sequence of values assigned to the first n latent variables in particle p . After extending a particle p , SIS updates the importance weight $\omega_n^p = \omega_{n-1}^p * \alpha_n(x_{1:n}^p)$. The weight update, defined as

$$\alpha_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n | x_{1:n-1})}, \quad (4.1)$$

accounts for the discrepancy between the proposal distribution, q_n , and the target distribution, π_n , without normalizing over $x_{1:n}$, which becomes intractable

for longer sequences even in discrete domains. The normalizing constant of the target distribution is approximately $Z_n \approx \sum_{p=1}^P \omega_n^p$ and the unnormalized density is $\gamma_n(x_{1:n}) \approx \sum_{p=1}^P \omega_n^p \delta_{x_{1:n}}(x_{1:n}^p)$, where $\delta_{x_{1:n}}(x_{1:n}^p) = 1$ if $x_{1:n}^p = x_{1:n}$ and 0 otherwise. The particles can also be used to generate an unbiased sample from π_n by choosing a particle p proportional to its weight ω_n^p .

Andrieu et al. (2010) shows that to ensure the samples generated by SMC for a Gibbs sampler has the target distribution as the invariant density, the particle filter must be modified to perform a *conditional SMC update*. This means that the particle filter guarantees that one of the final particles is assigned the same values as the previous Gibbs iteration. This implementation of the conditional SMC update reserves one special particle, 0, for which the proposal distribution always chooses the previous iteration’s value at that site.

4.3.1 Sentence Sampling

The **sent** particle filter samples blocks of tag assignments $\mathbf{t}_{1:n}^S$ for a sentence, S , composed of tokens, $\mathbf{w}_{1:n}^S$. Sampling an entire sentence minimizes the risk of assigning a tag with a high probability given its local context, but minimal probability given the entire sentence. Sentences can be sampled by ignoring table counts while sampling a proposal sentence, incorporating them after the fact with a Metropolis-Hastings (MH) acceptance test (Gao and Johnson, 2008). The MH acceptance test accepts or rejects the new sample after accounting for the associated changes in the seating arrangement. The Metropolis-Hastings step simplifies the sentence block particle filter further by not requiring the conditional SMC update, with the acceptance test ensuring ergodicity.

While there is already a tractable dynamic programming approach to sampling an entire sentence based on the Forward-Backward algorithm, particle filtering the sentences PYP-HMM model should prove beneficial. For the tri-

sentence:	The	new	rate	will	be	payable	Feb.	15	.	
GS:	DT	JJ	NN	MD	VB	JJ	NNP	CD	.	
P1:	DT	JJ	NN	VB	VB	JJ	NN	CD	.	.9
P2:	NN	JJ	NNS	VB	VB	JJS	JJ	NN	.	.1

Figure 4.1: A hypothetical example of the sentence particle filter with two particles (P1 and P2) as well as the gold standard (GS). The final number at the end is the normalized weight assigned to each particle. The final sample will choose one of these two particles with probability equal to each weight.

gram HMM defined by the model, the forward-backward sampling approach has time complexity in $O(NT^3)$ for a sentence of length N with T possible tag assignments at each site. Particle filters with P particles can approximate these samples in $O(NTP)$ time, which becomes much faster as T increases.

```

At iteration  $i = 0$  randomly initialise PoS assignments;
for  $i \leftarrow 1$  to Iterations do
  foreach sentence  $S$  in the corpus do
    Remove table counts from  $\mathbf{z}_{1:N}^S$ ;
    for  $n \leftarrow 1$  to  $N$  do
      for  $p \leftarrow 1$  to  $P$  do
        Sample  $t_n^{S,p} \sim q_n^S(\circ | \mathbf{t}_{1:n-1}^{S,p}, w_n^S)$  ; // equation (4.2)
        Update weight  $\omega_n^{S,p} \leftarrow \omega_{n-1}^{S,p} \cdot \alpha_n(t_{1:n}^{S,p})$  ; // equation (4.1)
      end
    end
    Sample  $p \propto \omega_N^{S,p}$ ;
    Sample new table counts  $\mathbf{z}_{1:N}^S$  based on  $\mathbf{t}_{1:N}^{S,p}$ ;
    if Metropolis-Hastings accepts then
      Set  $\mathbf{t}_{1:N}^S \leftarrow \mathbf{t}_{1:N}^{S,p}$ ;
    else
      Keep old  $\mathbf{t}_{1:N}^S$ ; // from iteration i-1
      Sample new table counts  $\mathbf{z}_{1:N}^S$  based on  $\mathbf{t}_{1:N}^S$ ;
    end
  end
end

```

Algorithm 1: sent Particle Gibbs Sampler

Figure 4.1 shows a hypothetical example of this algorithm for one sentence. The pseudocode for the Particle Gibbs **sent** sampler is shown in Algorithm 1.

Sampling of sentence S begins by removing all of the transitions and emissions in S from the table counts, $\mathbf{z}$, resulting in the table counts $\mathbf{z}^{-S}$ of tag assignments $\mathbf{t}^{-S}$ the values assigned to the variables outside of S . For each site index $n \in [1, N]$ in the sentence, the particle filter chooses the new tag assignment, $t_n^{S,p}$, for each particle $p \in [1, P]$ from the sentence proposal distribution,

$$\begin{aligned} q_n^S(t_n^{S,p} | \mathbf{t}_{1:n-1}^{S,p}, w_n^S) &\propto P(t_n^{S,p} | t_{n-2}^{S,p}, t_{n-1}^{S,p}, \mathbf{t}^{-S}, \mathbf{z}^{-S}) \\ &\times P(w_n^S | t_n^{S,p}, \mathbf{t}^{-S}, \mathbf{z}^{-S}, \mathbf{w}^{-S}). \end{aligned} \quad (4.2)$$

Where w_n^S is the word-token at site index n in sentence S and $\mathbf{w}^{-S}$ are the word-token variables outside of sentence S . After each new tag is assigned, the particle's weight is updated according to equation (4.1). The simplicity of the proposal density hints at the advantage of particle filtering over forward-backward sampling: it tracks only P histories and their weights rather than tracking the probability of over all possible histories. Once each particle has assigned a value to each site in the sentence, one tag sequence is chosen proportional to its particle weight, $\omega_N^{S,p}$.

4.3.2 Type Sampling

The type sampling case for the PYP-HMM is more complicated than the **sent** sampler. The long-range couplings defined by the hierarchical PYP priors strongly influence the joint distribution of tags assigned to tokens of the same word-type (Liang et al., 2010). For example, in the calculation of the posterior of the emission probability, the associated Chinese Restaurant will start with no tables associated with the sampled word-type. Therefore, the effects of the seating decisions of new customers cannot be postponed during filtering as in sentence sampling. To account for this, the **type** particle filter samples se-

quences of seating arrangements and tag assignments jointly, $\mathbf{x}_{1:n}^W = (\mathbf{t}_{1:n}^W, \mathbf{z}_{1:n}^W)$, for the word-type, W . The final table counts are resampled once a tag assignment has been chosen from the particles. Algorithm 2 shows the pseudocode of the **type** Particle Gibbs sampler.

```

At iteration  $i = 0$  randomly initialise PoS assignments;
for  $i \leftarrow 1$  to Iterations do
  foreach word-type  $W$  observed in the corpus do
    Remove table counts from  $\mathbf{z}_{1:N}^W$ ;
    for  $n \leftarrow 1$  to  $N$  do
       $t_n^{W,0} \leftarrow t_n^W$  ; // from iteration  $i - 1$ 
      for  $p \leftarrow 1$  to  $P$  do
        Sample  $t_n^{W,p} \sim q_n^W(\circ | \mathbf{t}_{1:n-1}^{W,p}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{w}_{1:n}^W)$  ; // equation (4.3.2)
        Sample  $z_n^{W,p}$  Update weight  $\omega_n^{W,p} \leftarrow \omega_{n-1}^{W,p} \cdot \alpha_n(t_{1:n}^{W,p})$  ; // equation (4.1)
      end
    end
    Sample  $p \propto \omega_N^{W,p}$ ;
    Set  $\mathbf{t}_{1:N}^W \leftarrow \mathbf{t}_{1:N}^{W,p}$ ;
    Sample new table counts  $\mathbf{z}_{1:N}^W$  based on  $\mathbf{t}_{1:N}^W$ ;
  end
end

```

Algorithm 2: type Particle Gibbs Sampler

Tracking the seating arrangement history for each particle adds an additional complication to the **type** particle filter. The exchangeability of seating decisions means that only counts of customers are necessary to represent the history. Each particle represents both a tag sequence, $\mathbf{t}_{1:n}^{W,p}$, and the count deltas, $\mathbf{z}_{1:n}^{W,p}$. The count deltas of each particle are stored in a hash table that maps a dish in one of the CRF restaurants to the number of tables serving that dish and the total number of customers seated at those tables. The count delta hash table ensures that it has sufficient data to calculate the correct probabilities (per equation (3.32)) by storing any counts that are different from the

base counts, $\mathbf{z}^{-W}$, and deferring to the base counts for any counts it does not have stored.

At each token occurrence n , the next tag assignment, $t_n^{W,p}$ for each particle $p \in [1, P]$ is chosen first according to the word-type proposal distribution

$$\begin{aligned} q_n^W(t_n^{W,p} | \mathbf{t}_{1:n-1}^{\mathbf{W},\mathbf{p}}, \mathbf{z}_{1:n-1}^{\mathbf{W},\mathbf{p}}, \mathbf{w}_{1:n}^{\mathbf{W}}) &\propto \\ &P(t_n^{W,p} | c_n^{-2}, c_n^{-1}, \mathbf{t}_{1:n-1}^{-\mathbf{W},\mathbf{p}}, \mathbf{z}_{1:n-1}^{-\mathbf{W},\mathbf{p}}) \\ &\times P(c_n^{+1} | c_n^{-1}, t_n^{W,p}, \mathbf{t}_{1:n-1}^{-\mathbf{W},\mathbf{p}}, \mathbf{z}_{1:n-1}^{-\mathbf{W},\mathbf{p}}) \\ &\times P(c_n^{+2} | t_n^{W,p}, c_n^{+1}, \mathbf{t}_{1:n-1}^{-\mathbf{W},\mathbf{p}}, \mathbf{z}_{1:n-1}^{-\mathbf{W},\mathbf{p}}) \\ &\times P(w_n^W | t_n^{W,p}, \mathbf{t}_{1:n-1}^{-\mathbf{W},\mathbf{p}}, \mathbf{z}_{1:n-1}^{-\mathbf{W},\mathbf{p}}, \mathbf{w}_{1:n-1}^{-\mathbf{W},\mathbf{p}}). \end{aligned}$$

In this case, $c_n^{\pm o}$ represents a tag in the context of site t_n^W offset by o , while $\mathbf{t}_{1:n-1}^{-W,p}$, $\mathbf{z}_{1:n-1}^{W,p}$, and $\mathbf{w}_{1:n-1}^{-W,p}$ represent the tag assignments, table counts, and word token values chosen by particle p as well as the values at all of the sites where a word token of type W does not appear. This proposal distribution ignores changes to the seating arrangement between the three transitions involving the site n . The specific seating arrangement of a particle is chosen after the tag choice, at which point the weights are updated by the result of equation (4.1). As with the **sent** sampler, once all of the particles have been sampled, one of them is sampled with probability proportional to its weight. This final sample is a sample from the true target probability asymptotically as P increases without bound.

As mentioned earlier, the sequence of particle approximations do not have the target distribution as invariant unless they use the conditional SMC update. Therefore, a special 0^{th} particle is automatically assigned the value from the prior iteration of the Gibbs sampler at each site n , though the proposal probability $q_n^W(t_n^{W,0} | \mathbf{t}_{1:n-1}^{W,p}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{w}_{1:n}^{\mathbf{W}})$ still has to be calculated to update the

weight $\omega_n^{W,p}$ properly. This ensures that the `type` sampler has a non-zero probability of reverting to the prior iteration’s sequence.

4.4 Experiments and Results

This section aims to explore several aspects of both the PG sampler and the specifics of the PoS inference task. Firstly, the motivation of the PG implementations is to allow better inference on tightly coupled models. This suggests that the PG samplers should mix better than the local sampler, finding the mode of the model in fewer iterations. The `type` sampler should perform especially better with the character-LM, which assigns a higher likelihood when words of the same type are labelled with the same tag than unigram-emission distribution. If the PG approach does find more likely modes, does that correspond to gains in practical measures like accuracy? Finally, how does the number of particles in PG samplers influence the inference? The hypothesis is that increasing the number of particles decreases the variance of the posterior likelihoods as the chance of generating a population dominated by likely modes increases. However, the marginal improvement from additional particles should exhibit diminishing returns.

Section 4.4.1 describes the corpora on which these hypotheses are tested. It evaluates the performance of the samplers with two approaches. The first approach is an analysis of the samplers as inference algorithms. Each approach should tend toward a mode in the distribution as it mixes, resulting in more likely restaurant configurations. Section 4.4.2 analyses the particle filter based samplers with various numbers of particles in an effort to understand how they behave. Then, section 4.4.3 evaluates each of the proposed approaches on PoS inference tasks from several languages. These results allow a practical

comparison with other PoS inference approaches.

4.4.1 Data

Section 4.4.2 tests and compares several different approaches with a number of parameters. To simplify the procedures, all of the tests in the analysis are performed on a reduced version of the Penn Wallstreet Journal (WSJ) treebank, similar to Gao and Johnson (2008) and Goldwater and Griffiths (2007). This reduced corpus is composed of the first 10,000 sentences in the Penn WSJ treebank, with 240,236 tokens. Additionally, this corpus uses the same, 17 tag, reduced tagset as Goldwater and Griffiths (2007), developed by Smith and Eisner (2005).

Section 4.4.3 compares the many-to-one (M-1) accuracy of the induced tag assignments on the full Penn WSJ treebank corpus, as well as the corpora from the CoNLL-X shared language task (Buchholz and Marsi, 2006). M-1 accuracy assigns the induced syntactic categories to the PoS of the most tokens of that category.

4.4.2 SMC Analysis

Before comparing the performance of the PG samplers to other inference methods, their behaviour should be analysed. It is not obvious how well the benefits of block sampling transfer to SMC based approaches. Both the **sent** and **type** samplers are novel approaches to computational linguistics, and many of their properties are unclear. For example, the samples generated from the particle filter should have a higher variance than the target distribution. If the variance is too high, the sampler will be slower to converge. While additional particles lower the relative variance, they also increase the run time linearly. It seems that there is a threshold of particles necessary to ensure that some are high

likelihood sequences, beyond which inference gains are minimal the additional computational expense is wasted.

Like other MCMC methods, Particle Gibbs generates a sequence of samples from the posterior distribution of the model in question. The PG sampler should mix more quickly than a local sampler because it takes larger steps. These larger steps should move the PG sampler to a more likely mode in fewer iterations. The results in this section measure the power of the inference as the rate at which the posterior likelihood of the restaurant configuration increases.

The sentence based sampler, **sent**, samples from a distribution that can be almost exactly computed¹, facilitating comparisons between the exact sampler and the SMC approach. Figure 4.2 compares the posterior log-likelihoods of the **sent** sampler and the exact sentence sampler over 200 iterations. As expected, the likelihoods of the particle filters approach that of the exact sentence sampler as the number of particles increases from 10 to 100.

Figure 4.3 compares the table configuration log-likelihood of the 1HMM approximation implemented by Blunsom and Cohn (2011) with the **type** particle filter based sampler as well as the local sampler and the exact block sentence sampler. Unlike the sentence based block sampler, **type** sampler cannot be exactly calculated, even with the 1HMM approach of constraining inference to only consider sequences that assign the same tag to every token of the same word-type. The 1HMM sampler approximates these probabilities using expected table counts. Theoretically, the **type** sampler should be a better approximation, being guaranteed to approach the true distribution as the number of particles increases. The results suggest that the **type** sampler can perform better than the 1HMM sampler with few particles. Unlike the **sent** sampler, the **type** sampler performs well even with fewer particles than the number of

¹Except for the minimal effect of intrasentence seating decisions accounted for with a Metropolis-Hastings acceptance test

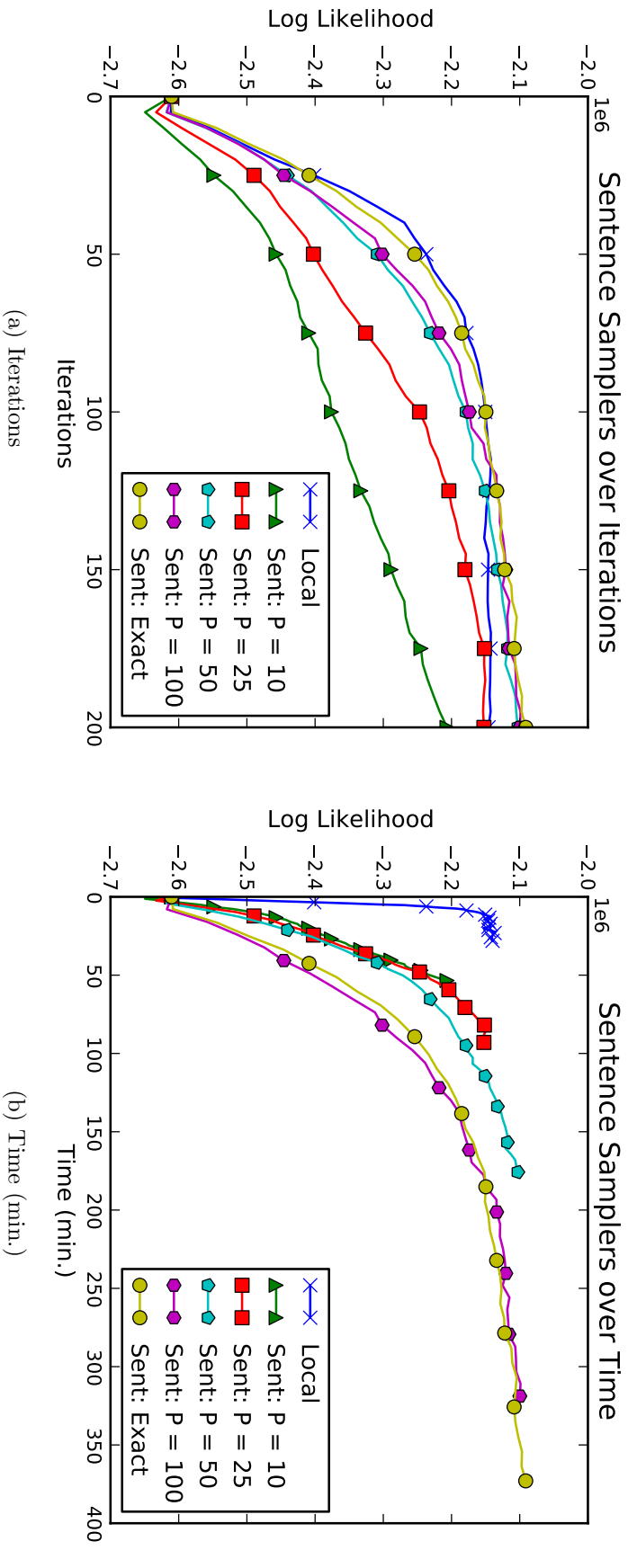


Figure 4.2: Posterior Log-Likelihood of PYP-HMM inference over iterations (4.2a) and time (4.2b) with exact as well as PG **sent** sampler with various numbers of particles. Except for the local sampler, which ran for 300 iterations in the time graph, the samplers were run for 200 iterations each, the end of a line represents the time to finish those 200 iterations.

PoS categories in the tagset. These results suggest that the choice of block has a strong influence on the resulting sampler.

Interestingly, the local sampler reaches a higher mode than the 1HMM approach before the iteration 50. However, earlier work by Blunsom and Cohn (Blunsom and Cohn, 2011) found that the 1HMM approximation achieved a consistently higher M-1 accuracy than the local sampler. Section 4.4.3 confirms the same result. This reveals a disconnect between the likelihood under the PYP-HMM model and the M-1 accuracy. Additionally, both the **type** and 1HMM samplers reach likely modes within 30 iterations, after which they plateau and the additional cost of these approaches is wasted.

An interesting variant of the basic PYP-HMM model uses replaces the unigram base of the emission distribution with a bigram character language model (PYP-HMM-LM) (Blunsom and Cohn, 2011). In addition to allowing the PYP-HMM to recognize basic word morphology, the character language model creates a strong bias toward only one tag per word-type. Figure 4.4a shows that the PYP-HMM-LM model is too tightly coupled for either the local or the **sent** samplers to mix quickly as they did with the simpler PYP-HMM model. The PYP-HMM-LM model reveals a weakness of the simple local Gibbs sampler approach, because the probability of a word being emitted by a new state is so low that sampler is highly likely to choose the same PoS tag assignment as other occurrences of that word. The additional context of the **sent** sampler results in bigger steps and faster mixing than the local sampler, but most of the variables influencing probability of and particular tag assignment are in different sentences.

On the other hand, the **type** and 1HMM samplers sample all of the tags assignments for words of the same type simultaneously. Figure 4.4b shows that the result is drastically faster mixing. The **type** sampler is just as strongly

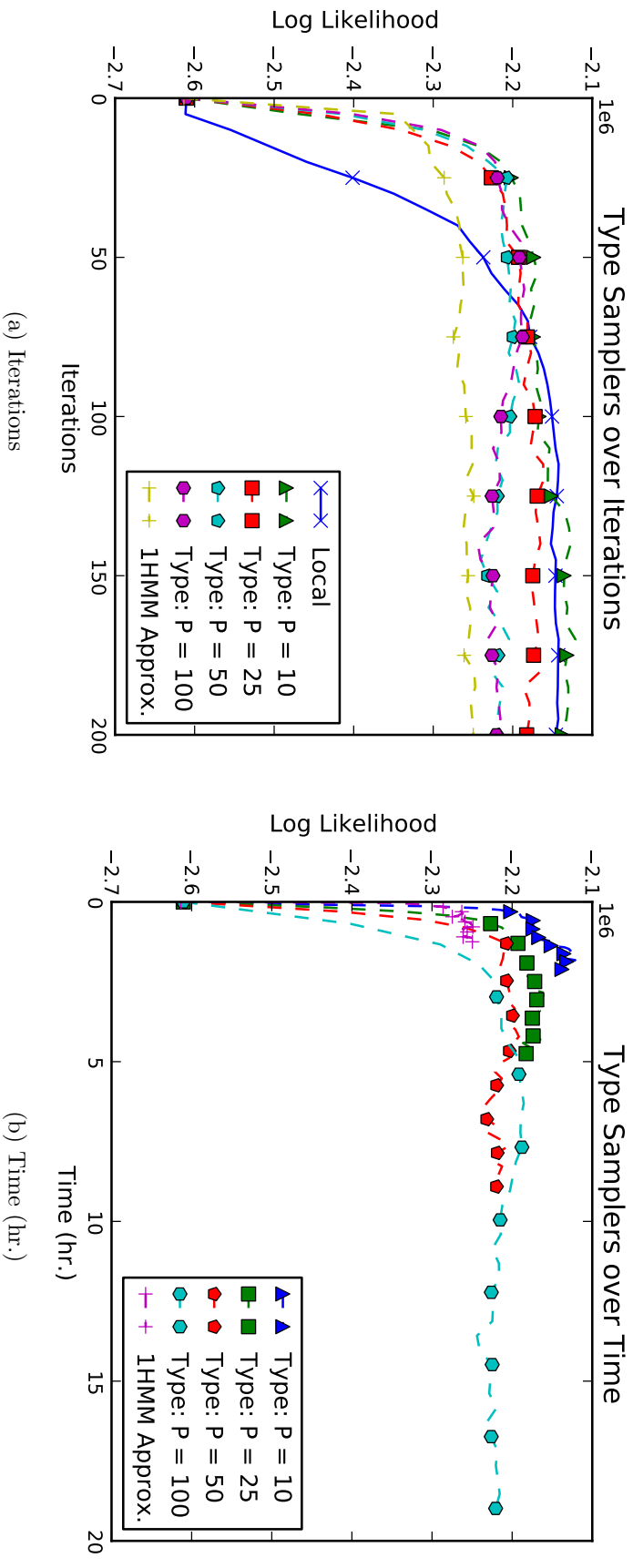


Figure 4.3: Posterior Log-Likelihood of PYP-HMM inference over iterations (4.3a) and time (4.3b) with the type sampler as well as the 1HMM approximation proposed by Blunsom and Cohn (2011) with various numbers of particles. Except for the local sampler, which ran for 300 iterations in the time graph, the samplers were run for 200 iterations each, the end of a line represents the time to finish those 200 iterations.

influenced to assign all words of the same type to the same PoS category, but there are no same-word-type assignments to bias the choice of which one tag they will all get. This strong correlation between the tags assigned to words of the same type may explain the **type** sampler’s strong performance with so many fewer particles than necessary for the **sent** sampler: once a tag has been assigned to a sufficient number of words of the same type, that same tag will be progressively more likely to be chosen again. Perhaps a one-tag-per-type particle filter can take advantage of this fact to simplify the proposal distribution.

The fact that the **type** sampler requires so few particles for inference relative to the **sent** sampler suggests that the choice of block can heavily influence the performance of a block Particle Gibbs sampler. The optimal block for a given model or dataset is not obvious and may be difficult to determine empirically. On the other hand, the large and expensive steps taken by a Particle Gibbs sampler are likely only necessary when faster methods reach modes from which they are slow to escape. A mixed sampler that only occasionally takes large steps with Particle Gibbs might achieve similar results with less computational cost. Such an approach need not even rely on any specific block partitioning, particle filters do not place many restrictions on the distributions from which they generate samples.

There are a multitude of possible combinations of mixed samplers, figure 4.5 demonstrates the performance of a few instantiations of one simple approach. Each of the mixed samplers in figure 4.5 randomly chooses on of either the local, **sent**, or **type** samplers (the numbers in the legend describe the ratio of each sampler in the mixture.) For the sake of simplicity, each mixed sampler uses the same number of particles for the **sent** and **type** samplers, the mixed samplers in figure 4.5 all use ten particles. While there is a high degree of

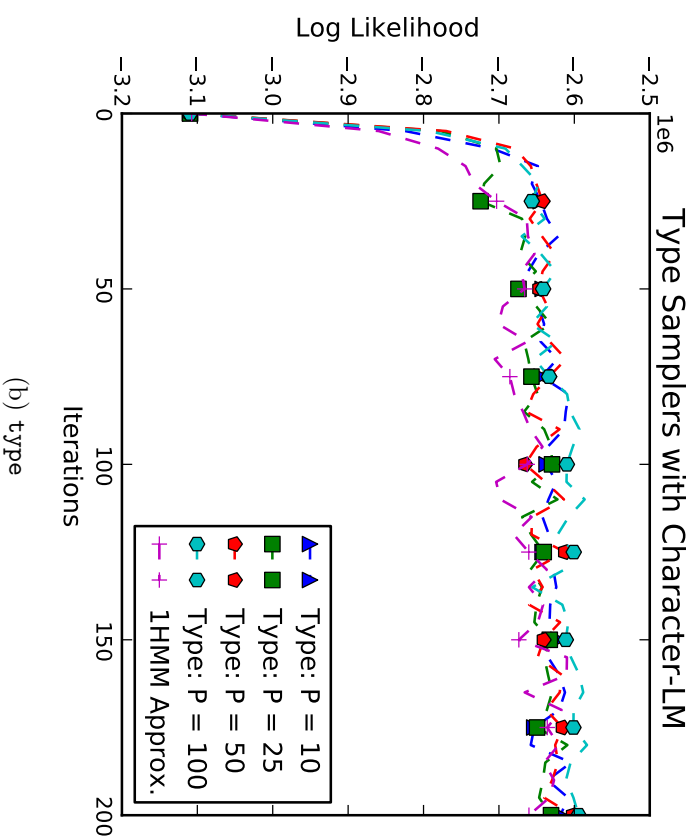
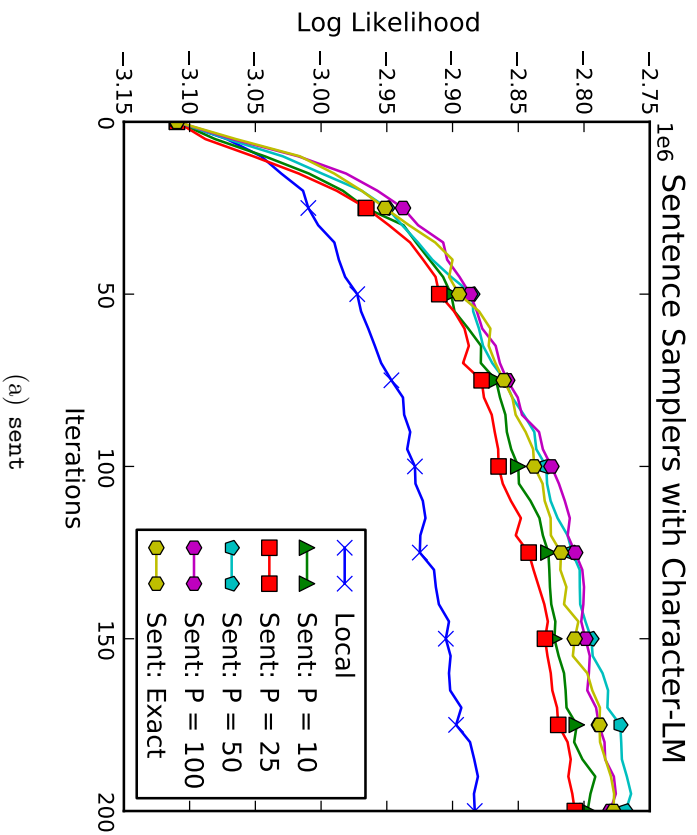


Figure 4.4: Posterior Log-Likelihood of PYP-HMM-LM inference with the sent sampler (4.4a), and the type sampler as well as the IHMM approximation proposed by Blunson and Cohn (2011) (4.4b) with various numbers of particles.

variance, the mixed samplers all show the rapid migration toward the mode shown by the `type` sampler, but each iteration takes less time on average because of the `sent` and local samplers. The results in figure 4.5b suggest that the computational costs of the Particle Gibbs samplers can be mitigated by mixed sampling without eliminating the benefits from particle filtering. Finally, the samplers with the `sent` sampler in the mix performed quite well with ten particles despite its poor performance with the same number of particles on its own. Perhaps mixing the block samplers diminishes the particle cost of poor block choice.

4.4.3 Unsupervised Part-of-Speech Tagging

Table 4.1 compares the M-1 accuracy of the `sent` and `type` particle filter samplers, from sections 4.3.1 and 4.3.2, with 100 particles each. Each site in a corpus is assigned the most commonly visited tag assignment at that site over all iterations. The particle filter based samplers rarely score a higher accuracy than even the local sampler, which completes 500 iterations before the particle filters complete 200.

While figures 4.2a and 4.3a show that all of the samplers surpass the 1HMM sampler in likelihood, the accuracies of the 1HMM and 1HMM-LM approximations remain well above the other approaches. This suggests that there are high-likelihood assignments that produce lower accuracy results, presumably related to the fact that the `type` sampler is not restricted to assignments with exactly one tag for each word-type. If the model assigns equal likelihood to these assignments, inference will not be able to distinguish between them.

The Type-10-LM sampler outperforms all of the other non-restrictive approaches, except in the case of Arabic. This suggests that the PYP-HMM-LM model may be a more accurate representation of PoS. As noted in section 4.4.2,

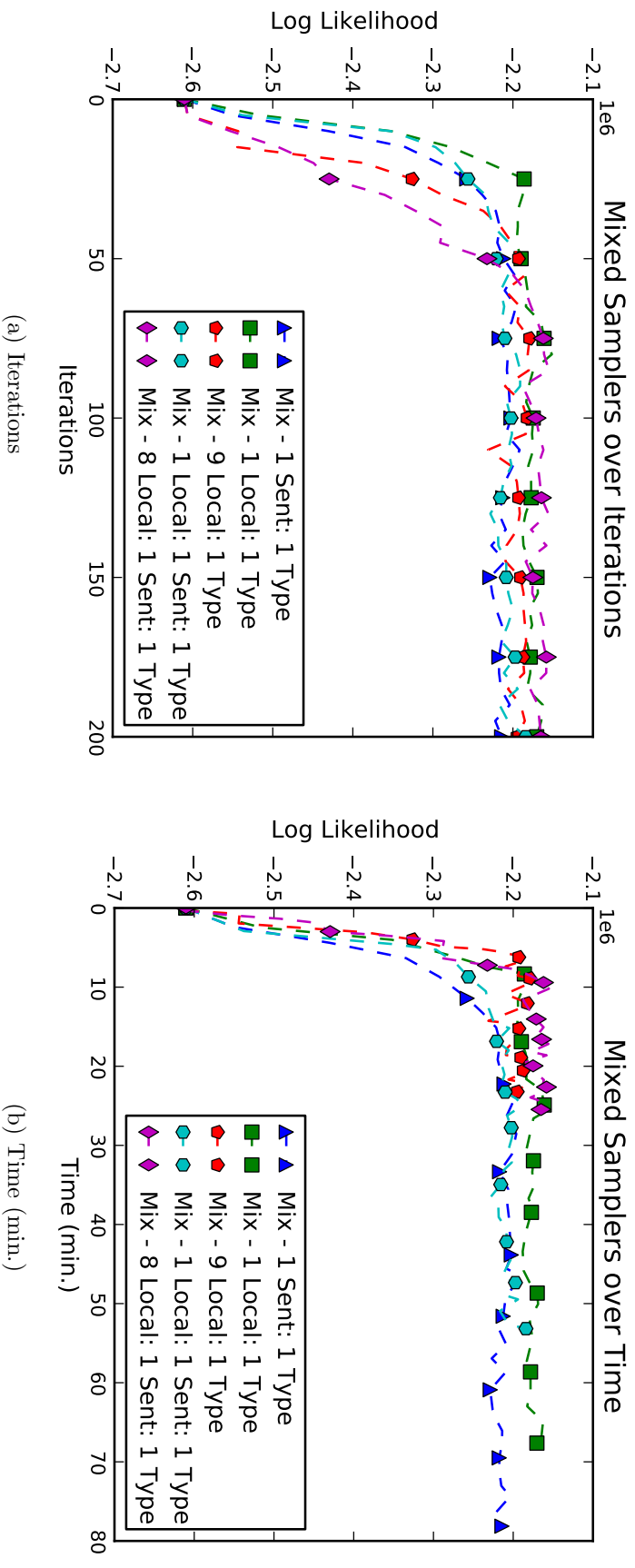


Figure 4.5: Posterior Log-Likelihood of PYP-HMM inference with various mixed samplers over 200 iterations (4.5a) and over time (4.5b). The colon separated lists provide the ratio of each mixed sampler: the sampler run on any given iteration is proportional to the number next to it. In each sampler, the particle filters are run using 10 particles. The samplers were run for 200 iterations each, the end of a line represents the time to finish those 200 iterations.

Language	Sent-100	Type-100	Type-10-LM	Local	1HMM	1HMM-LM	Tokens	Tag types
WSJ	69.8%	70.1%	74.7%	70.2%	75.6%	77.5%	1,173,766	45
Arabic	53.5%	57.6%	51.3%	56.2%	61.9%	62.0%	54,379	20
Bulgarian	64.8%	67.8%	71.6%	67.6%	71.4%	76.2%	190,217	54
Czech	59.8%	61.6%	65.2%	64.5%	65.4%	67.9%	1,249,408	12 ^c
Danish	65.0%	70.3%	74.9%	69.1%	70.6%	74.6%	94,386	25
Dutch	61.6%	71.6%	70.1%	64.1%	73.2%	72.9%	195,069	13 ^c
Hungarian	61.8%	61.8%	68.5%	64.8%	69.6%	73.2%	131,799	43
Portuguese	59.4%	71.1%	74.4%	68.1%	72.0%	77.1%	206,678	22
Spanish	66.3%	69.1%	75.3%	68.5%	74.7%	78.8%	89,334	47
Swedish	62.9%	63.5%	68.3%	67.6%	67.2%	68.6%	191,467	41

Table 4.1: Many-to-1 accuracies on CoNLL and Penn-Treebank Wall Street Journal corpora for sentence- (Sent) and type- (Type) based filtering. The table lists the average M-1 accuracy measured according to the maximum marginal tag assignments over 3 separate runs after 200 iterations for the **sent**, **type**, 1HMM and 1HMM-LM samplers, and 500 iterations for the HMM local sampler.

the PYP-HMM-LM model is biased toward fewer tags per type than the more standard HMM model, resulting in an average number of tags per word-type that is closer to the true value. Even so, figure 4.4b shows that the **type** sampler mixes at least as well as the 1HMM sampler with the character language model, yet the 1HMM sampler still scored a higher many-to-one accuracy on all languages other than Danish. This discrepancy between model likelihood and accuracy suggests that a different model is necessary to further improve unsupervised PoS induction.

The discrepancy between the 1HMM sampler and the **type** sampler without constraints reveals that the unsupervised PYP-HMM model over assigns the number of tags per type. One solutions to reduce the number of tags assigned to a word-type are semi-supervised constraints on the tag assignments. Such constraints can be easily extracted from a relatively small set of labelled data or from freely available dictionaries such as the Wiktionary. Another approach is to extend the PYP-HMM so that it assigns a higher probability to tag assignments with fewer tags per type.

4.5 Discussion

This chapter explores the effects and behaviour of the Particle Gibbs Sampler, described in detail in section 4.3. By only accounting for those variables that have already been sampled, particle filters avoid the intractable calculations inherent to some samples. Global effects are accounted for as sampling continues by updating an importance weight and tracking multiple, independent sequences of samples, called particles. The particle filtering algorithm is incorporated into a Gibbs sampler through the conditional SMC update, which ensures ergodicity by guaranteeing that one particle represents the sample from

the previous iteration. Sections 4.3.1 and 4.3.2 describe the particle filtering algorithm for sentence- and type-based block samplers, respectively.

Then, section 4.4 explored the behaviour and performance of these new algorithms in a variety of experiments. The graphs in section 4.4.2 demonstrated the overall behaviour of the Particle Gibbs approach. Comparing the Particle Gibbs **sent** sampler to the forward-backward sentence sampler showed that increasing the number of particles increases the mixing of the sampler. Increasing the number of particles only improves performance up to a point. Still, with sufficient particles, the Particle Gibbs **sent** sampler can match the performance of the forward-backward sentence sampler in less time.

Additionally, the local and Particle Gibbs **type** samplers achieve a mode with a better likelihood under the model than the 1HMM. The Particle Gibbs **type** sampler and 1HMM sampler both mix very quickly, reaching a plateau within 50 iterations. Unfortunately, the Particle Gibbs **type** sampler requires many particles to mix well, resulting in a considerable time cost. This time cost can be somewhat mitigated by mixing the **type** and local samplers, achieving similar results while only using the **type** particle filter occasionally.

Finally, section 4.4.3 compared the M-1 accuracy of each of the local, **sent**, **type**, and 1HMM samplers on 10 different language corpora. This analysis revealed that there was a disconnect between the likelihood of the samples generated on the PYP-HMM model and the accuracy of the resulting tag assignments. While the Particle Gibbs **type** sampler achieves a higher accuracy than the local sampler and is able to mix with the morphological character language model, it still performs worse than the one-tag-per-type 1HMM approximation

These results ultimately suggest that there is more benefit from the limitations of the one-tag-per-type assumption than tractable inference. The un-

restricted local and `type` samplers are assignment too many tags per type because the PYP-HMM model favours too many tags per type. One solution to this problem is to restrict the tag assignments for each word-type according to the observed tag assignments from a semi-supervised labelled corpus or tag-dictionary. The other solution is to expand the PYP-HMM model to include a soft constraint on the number of tags per type, an approach that is viable with the `type` Particle Gibbs sampler.

Chapter 5

Type-based Semi-supervision

5.1 Introduction

Much of the research into fully unsupervised PoS tagging is motivated by the lack of expensive fully labelled language corpora. However, there is a growing body of less informative, but less expensive, supervised data in the form of smaller labelled corpora (Garrette and Baldridge, 2013), the wiktionary (Li et al., 2012), and parallel corpora (Das and Petrov, 2011). An advantage of unsupervised induction methods like Gibbs sampling is that it is relatively straightforward to incorporate this type of minimal supervision. For example, labelled data contributes to the Chinese Restaurant Franchise table counts from Chapter 3 when calculating probabilities but doesn't need to be resampled.

Furthermore, semi-supervised tag-dictionaries provide explicit constraints on the set of tags assigned to a word type. These constraints limit the common tendency of unsupervised models to assign too many tags per word-type. They provide this effect without restricting word-types that should have multiple tags to only one tag. In fact, even a small supervised tag-dictionary can

improve the assignments of unseen words when a word in the dictionary is in an unseen word’s local context.

An obvious source of tag-dictionaries is a small labelled corpus. If a word is common enough that it occurs multiple times with such a corpus it is reasonable to assume that all of its possible labels have been observed. This approach has the additional advantage of providing some fully labelled data that gives additional supervision. It is interesting to explore the benefits of context-based, type-based, and combined semi-supervision.

A newer approach extracts tag-dictionaries from the Wiktionary (Li et al., 2012). The Wiktionary is an open dictionary that can be freely accessed and modified by the public. As such, it is a steadily growing source of information in a large variety of languages. As a dictionary, each word entry lists the possible parts-of-speech of a word. Thus, the Wiktionary acts as a massive, freely available source of semi-supervised tag-dictionaries. Unfortunately, the Wiktionary’s tagset is usually different from the tagset of the test data. The solution is to map both the Wiktionary and test tag sets to an intermediary, like the Universal Part-of-Speech Tagset proposed by Petrov et al. (2012).

This chapter explores both approaches to semi-supervised part-of-speech induction. Section 5.2 describes the related literature of semi-supervised PoS induction. Then, section 5.3 explores type and context based semi-supervision with a subset of the Penn Wall Street Journal Treebank corpus. Section 5.4 explores type based semi-supervision with tag-dictionaries extracted from the Wiktionary. Finally, section 5.5 summarises the results and conclusions from this examination of semi-supervised PoS induction.

5.2 Background

As explained in Chapter 2, the first attempts at unsupervised PoS tagging assumed the existence of tag-dictionaries (Brill, 1995, Merialdo, 1993, Elworthy, 1994, Kupiec, 1992, Cutting et al., 1992). These tag-dictionaries would list the ambiguity class, the set of all tags that a word-type is labelled with, for every observed word-types. This turns the task of PoS induction into one of PoS *discrimination*: determining which of the available tags to assign to a given token. Tag-dictionaries are less demanding in time and other expenses and can generally be extracted from online dictionaries or native speakers without much training (Garrette and Baldridge, 2013).

This approach has many advantages over the fully unsupervised PoS induction from Chapter 2. First, PoS discrimination results in a tagged corpus that can be directly evaluated against a labelled gold-standard. The tag-dictionary injects some human knowledge of what each part-of-speech means because of the word-types to which it can be assigned. A tag-dictionary can result in a significant improvement in accuracy by reducing confusion for words that do not occur enough to be distinguished according to their distribution. Finally, training with a tag-dictionary is considerably faster than without because only a limited subset of the full tagset is considered for assignment to each token.

Historically, the tag-dictionaries were extracted from the entire corpus, including the test partition (Banko and Moore, 2004). Including the test data ensures that the vocabulary of the model accounts for words that only exist in the test data. However, this construction of tag-dictionaries is problematic because many words only occur a few times and with only one tag, allowing baseline test accuracies of e.g 90.7% (Brill, 1995). Beyond that, many of these tag-dictionaries were automatically pruned to only include tags for a given word-type if that tag-type pair occurred above a specific threshold

(Banko and Moore, 2004). These facts lead to serious questions about the representativeness of the tag-dictionaries generated this way.

Some approaches attempt to address this by artificially limiting the tag-dictionary to frequent word-types or extracting only from a subset of the full corpus (Toutanova and Johnson, 2008, Smith and Eisner, 2005). Another proposal in Haghighi and Klein (2006) uses only a few word-type prototypes for each part-of-speech.

More recent work has focused on extracting tag-dictionaries from independent sources. Garrette et al. (2013) create new type- and token-annotations for low-resource languages by limiting the time spent annotating to only a few hours. Das and Petrov (2011) utilize a supervised English PoS tagger and bilingual parallel data to extract tag-dictionaries across aligned words. Similarly, Li et al. (2012) extract a tag dictionary from the Wiktionary. Both of these methods require that the labels are converted to the Universal Part-of-Speech Tagset proposed by Petrov et al. (Petrov et al., 2012). Still, they both are practical and extendible to a variety of languages with limited labelled data.

5.3 Labelled Semi-Supervised Part-of-Speech Discrimination

While the unlabelled data is generally easy to get, even in resource poor languages, a small amount of labelled data can provide a large benefit. Only a few hundred labelled sentences can be enough to surpass state-of-the-art unsupervised approaches. The standard approach to semi-supervision is to take a small amount of labelled data and use unlabelled data to make up for any missing patterns (Chapelle et al., 2006). This approach is complicated by the need to minimize the effects training on noisily tagged unlabelled data.

An alternative is type-based supervision, where a tag-dictionary is extracted from the labelled data to restrict the possible tag assignments for common words. Both of these approaches can be combined, allowing the tag-dictionary to reduce the noise in the assignments of the unlabelled data.

There are many ways to combine labelled and unlabelled data, the following experiments explore the effects of the variables involved in semi-supervised learning. The experiments are performed using the Penn Wall Street Journal Treebank (WSJ) with sections 2-21 used as training data and section 23 used for testing¹. The labelled and unlabelled sentences are both randomly taken from the training data. The following experiments will explore the effects of varying the amount of labelled data, the amount of unlabelled data, the size of the tag-dictionary extracted from labelled data, as well as other variables more specific to each approach. Section 5.3.1 explores the complexities of the traditional approach to semi-supervised learning. Then, section 5.3.2 examines extracting a tag-dictionary from labelled data, as well as the motivations and benefits of doing so. Finally, section 5.3.3 combines the two approaches to gain the most improvement with the fewest labelled sentences.

5.3.1 Traditional Semi-Supervision

The first approach to semi-supervision is the traditional one. A small corpus is labelled in a supervised manner and additional unlabelled data is provided. The intention is for the supervised data to provide a strong bias when learning on the remaining unlabelled data. In turn, the unlabelled data demonstrate new structure and unseen vocabulary to refine the supervised model. The straightforward extension of the unsupervised Gibbs sampler to a semi-supervised one is to not resample the assignments on labelled data while still

¹See section 2.3.1 for more on the WSJ corpus

# Sentences	Supervised	Semi-supervised
100	75.3%	75.1%
500	81.9%	81.9%
1000	85.5%	83.2%
5000	92.5%	89.4%
10,000	93.9%	90.9%
15,000	94.5%	91.8%
20,000	94.9%	91.9%

Table 5.1: Fully supervised and simple semi-supervised PoS tagging accuracy of the local sampler described in Chapter 2 trained on 100 to 20,000 random sentences from the standard training/test split of the Penn Wall Street Journal Treebank. The semi-supervised tagger adds an equal number of unlabelled sentences to the labelled sentences from the training set.

resampling the seating arrangements. In the case of the non-parametric PYP-HMM model, the latent CRP seating arrangements are not observed and must still be resampled occasionally. The labelled data is also used to randomly initialise the unlabelled data before adding the associated customers to the CRF, reducing noise in the initial sample.

The number of labelled sentences has an obvious effect on the final accuracy of the result. As the amount of labelled data increases, it becomes a better representation of the domain. At a some level of supervision, there is little room for improvement from additional labelled or unlabelled data. This becomes a problem for other evaluations because it becomes difficult to judge the difference in performance between different approaches. Table 5.1 shows the evaluation of fully supervised data on the Penn WSJ Treebank from 100 to 20,000 labelled sentences. Based on the results from table 5.1, 5000 labelled sentences is where the benefits of additional labelled data show diminishing returns. The remaining tests will be restricted to only 5000 labelled sentences unless otherwise stated to simplify the analysis of other effects.

The second column of table 5.1 is the accuracy of the simple semi-supervised approach described above. It demonstrates a common problem with semi-

supervision where the unlabelled data overwhelms the labelled data, lowering the overall accuracy. Most semi-supervised applications of this sort heavily weight the limited supervised data to prevent the larger and noisier unlabelled data from overwhelming the model. Unfortunately, the hierarchical non-parametric PYP-HMM model does not easily allow for assigning weights. Counting customers for labelled data as multiple customers leads to questions of how to propagate a multi-customer down the CRF hierarchy. For example, if one supervised customer counts as 100 customers and they decide to sit at a new table, do they all sit at the same new table or are there 100 new tables. Either of these options violate the posterior probability calculated by the Chinese Restaurant Process and have an unpredictable effect, especially on the base probabilities.

The simplest solution is to treat each supervised sentence as if it occurs multiple times. For each token in the labelled data, w customers are entered into the relevant restaurants in the CRP for a given weight, w . For example, if a sentence begins with the word “The” labelled as “DT”, the restaurant representing the bigram “(\$ \$)” will have w customers enter and sit at tables that represent “DT”. Some of these costumers may choose to sit at a new table, this will represent a new customer entering the “\$” restaurant as explained in Chapter 3.

Table 5.2 shows the results of this weighting method with various weights and number of unlabelled sentences. After a weight of 50, the unlabelled data begins to improve, rather than reduce, the accuracy of the trained model. The higher weights significantly slow the training process because this approach adds a customer for each point of weight. With so many customers, there is less variance in the latent seating arrangement, so the supervised seating arrangement only needs to be resampled occasionally. To save on memory, the

	Unlabelled Sentences		
Weight	0	5000	15000
1	92.2%	88.9%	84.9%
3	91.8%	88.8%	85.4%
5	92.6%	89.6%	86.1%
10	93.6%	93.6%	92.7%
50	93.3%	93.6%	93.7%
100	93.3%	93.6%	93.7%
500	93.5%	93.7%	93.4%

Table 5.2: Semi-supervised accuracy with weighted labelled data and increasing unlabelled data. Weight is applied by repeating labelled corpus.

# Sentences	Local	Type	Local+LM	Type+LM
5000	93.6%	93.6%	94.8%	95.0%
15,000	93.7%	93.7%	94.8%	95.0%
25,000	93.8%	93.6%	94.9%	94.6%
35,000	93.8%	93.4%	94.9%	94.6%

Table 5.3: Traditional semi-supervised learning with 5000 labelled sentences at weight 50 with the Local and Type-based samplers with and without the morphological character language model.

implementation should simply add multiple customers for every supervised data point rather than hold multiple copies of the same labelled data. While the weighting approach does minimize the negative effects of the unlabelled data, the additional data does little to improve the accuracy on the test set.

The remaining question is whether this approach results in greater improvements with the PYP-HMM+LM morphological model and with the **type** Particle Gibbs sampler. Table 5.3 shows the semi-supervised approach with 5000 labelled sentences and an increasing number of unlabelled sentences for each of these models. These results were all trained with a weight of 50, based on the results of table 5.2.

The results in table 5.3 show that the local and **type** samplers perform similarly even as more data is added. This is similar to the fully unsupervised results in Chapter 4. The difference is that while the fully unsupervised lo-

cal+LM sampler is unable to mix, resulting in incredibly poor performance, the semi-supervised local+LM sampler performs better than without the character language model. Since the local sampler is generally the fastest, there is no advantage to using type-based sampling in the semi-supervised case.

Interestingly, the additional unlabelled data does not improve the performance of any of the samplers. While weighting the labelled data as discussed earlier prevents the unlabelled data from lowering the accuracy, it does not allow the sampler to benefit from the unlabelled data. One possibility is that the weighted data biases the model to the point that the unlabelled data is ineffectual. The other possibility is that the weighting prevents the noisy assignments on the unlabelled data from compounding errors and instead the erroneous noise is balanced by the correct noise. Type-based semi-supervision using a tag-dictionary is another method to potentially reduce the noisy assignments on unlabelled data by focussing assignments to more likely tags. Such a method can be combined with the traditional, token-based approach to enable the semi-supervised samplers to learn from the addition of unlabelled data.

5.3.2 Type-Based semi-supervision

While the labelled data provide a context-based bias, they do little to address the tags-per-type problem demonstrated in Chapter 4. This problem is best addressed with type-based semi-supervision, wherein the a tag-dictionary, or lexicon of words and the parts-of-speech which may be assigned to them. While such a tag-dictionary is traditionally extracted from a labelled corpus, it is generally meant to represent more easily accessible data. For example, it requires considerably less investment in time, effort, and expertise to assign an ambiguity classes for the 1462 word-types that are common enough to occur

# Sentences	Local	1HMM	Type	Local+LM	1HMM+LM	Type+LM
0	77.2%	77.7%	78.7%	68.4%	82.3%	84.1%
100	77.2%	77.0%	80.2%	66.7%	80.0%	83.9%
1000	75.8%	76.4%	78.7%	67.1%	81.6%	82.0%
5000	75.2%	77.6%	77.6%	67.7%	82.2%	83.0%

Table 5.4: Type-based supervision accuracy with a lexicon of word-types that occur at least 10 times in a corpus of 5000 labelled sentences. The training was done with varying numbers of unlabelled sentences randomly chosen from the WSJ corpus.

at least 10 times within a random 5000 sentences of the WSJ corpus, rather than labelling the 119,204 tokens in the same sentences.

By restricting the parts-of-speech of common word-types, these tag-dictionaries counteract the tendency of the PYP-HMM model to favour more tags-per-type than are observed. This would result in the **type** Particle Gibbs sampler matching the performance of the 1HMM approximation. In addition, the tag-dictionary allows a considerable speed increase as the inference has fewer options for the most frequent word-types. There is also the possibility that the tag restrictions can cause the local sampler to overcome the mixing problems it encounters with the PYP-HMM+LM model. Table 5.4 shows the type-based semi-supervision results with the lexicon described earlier for the local and **type** samplers with and without the character language model.

Table 5.4 shows that the type-based semi-supervision eliminates most of the difference between the PG **type** sampler and the 1HMM sampler. This confirms the hypothesis that type-supervision is a straightforward method of mitigating the PYP-HMM’s tendency to assign too many tags-per-type. Furthermore, while the local sampler does perform better with the tag-dictionary, it still gets stuck in local optima with the PYP-HMM+LM, preventing mixing and resulting in lower performance than without the character language model. Note that the local sampler appears to struggle as more unlabelled training

data is added, this is likely a result of an increasing number of unseen words. The supervised lexicon is so small that it doesn't have enough examples of some tags to distinguish between them. The type-based samplers do not show this effect, perhaps because they mix better. This implies that as the size of the unlabelled corpus grows, the local sampler requires more coverage while the type samplers do not.

The **type** sampler actually surpasses the 1HMM sampler with the type-based supervision. This is because the **type** sampler has the flexibility to assign multiple tags to a given word type. Unlike the fully unsupervised case, the tag-dictionary limits the **type** sampler to only assign observed tags and only where multiple tags should be assigned. On the other hand, the 1HMM sampler is still assigning only one tag-per-type even when the tag-dictionary shows that the word-type occurs with more. Interestingly, the **type**+LM sampler achieves up to 84.1% accuracy with the very limited tag-dictionary extracted from only 5000 sentences with words that occurred at least 10 times in those sentences. This approaches the 85% reported by the LDA PoS model² of Toutanova and Johnson (2008) with a tag-dictionary extracted from the full WSJ corpus with a cutoff of 3. This suggests that a considerable benefit results from a relatively small lexicon as well as a relatively small amount of labelled data.

5.3.3 Combined Supervision

The promising results of the type-based supervision suggest that traditional semi-supervision can be improved by including the tag-dictionary extracted from the labelled data itself. This approach combines the benefits of biasing the HMM parameters with labelled data with the noise reducing effects of the tag-dictionary. As before, the type-based semi-supervision also reduces the

²without the full ambiguity class model

	Unlabelled Sentences		
Weight	0	5000	15,000
1	92.8%	92.1%	91.2%
3	93.5%	93.4%	92.7%
5	93.5%	93.6%	93.1%
10	93.7%	93.8%	93.5%
50	93.7%	93.7%	93.9%

Table 5.5: Combined semi-supervision with varying weights. The local sampler is trained on 5000 labelled sentences and 0 to 15,000 unlabelled sentences. The labelled sentences are weighted in the manner described in section 5.3.1.

training time on the unlabelled data. The idea is to improve the traditional semi-supervised approach by incorporating a tag-dictionary extracted from the already available labelled sentences. The noise-reducing effects of the tag-dictionary may compensate for lowering or even eliminating the need to heavily weight the labelled data. Table 5.5 examines combined type- and token- based semi-supervision with different weights. The combined approach still shows the most improvement with a weight of 50, but the difference between a weight of 1 or 5 and a weight of 50 is much less than in the traditional semi-supervised approach.

Finally, section 5.3.2 showed that the **type** sampler outperformed the local sampler with the type-based semi-supervision. This bears examining the various samplers performance with the combined supervision approach. The superior performance of the **type** sampler in the type-supervision task leads to the hypothesis that it will also outperform the local sampler in the combined supervision task. Table 5.6 shows the combined supervision performance with 5000 labelled sentences and the tags extracted from them for both the local and the **type** samplers with and without the morphology model.

Like in the traditional semi-supervision approach, the local and **type** samplers achieve almost the same accuracy with both type- and token-based semi-supervision. Additionally, while the type-supervision in section 5.3.2 did not

# Sentences	Local	Type	Local+LM	Type+LM
0	93.7%	93.5%	94.9%	94.9%
1000	93.7%	93.6%	95.0%	95.0%
5000	93.7%	93.7%	94.9%	95.1%
35,000	94.1%	93.7%	94.8%	94.7%

Table 5.6: Accuracy of semi-supervised local and **type** samplers with 5000 sentences of labelled data and a tag-dictionary extracted from it.

improve the performance of the local+LM sampler, the traditional supervision did. The combined supervision approach retains the benefits of the local+LM sampler, so that it performs as well as the **type**+LM sampler. This implies that when labelled data is available, the local sampler is superior to the **type** approach as it is considerably faster. A universal speed-up results from the use of the tag-dictionary. Even the limited tag-dictionary explored in this analysis restricts the most common words enough that the **type** sampler is three times as fast with the tag-dictionary than without it.

Unlike traditional semi-supervision, table 5.6 shows an improvement in accuracy of .4% with the addition of 35,000 unlabelled sentences. This confirms the hypothesis that the type-supervision reduces the noise from the tags assigned to unlabelled data. These results suggest that further accuracy improvements might be possible if the labelled data is augmented with large amounts of unlabelled data. More labelled data can also be used to increase the coverage of the unlabelled vocabulary, further reducing the noise and increasing the speed. The speed benefits of the tag-dictionary should also be stressed as larger amounts of unlabelled data will result in a considerable time investment.

Ultimately, the combined sampler achieves an accuracy of 95% with only 5000 labelled sentences and a generative model with minimal features. This approach can be improved with more labelled data, more unlabelled data, and a more extensive tag-dictionary. As the amount of labelled data increases, the

tag-dictionary extracted from it becomes more comprehensive. The results here show that a tag-dictionary extracted from 5000 sentences can still cover most of the word-tokens in a corpus of 35,000 sentences. However, for massive amounts of unlabelled data, or when very little labelled data exists, better sources are needed for tag-dictionaries.

5.4 Wikily-Supervised Part-of-Speech Induction

Section 5.3.2 shows how type-based semi-supervision prevents the PYP-HMM model from assigning too many tags per type. It also demonstrates the benefit of larger tag-dictionaries for larger corpora. However, relying on larger labelled corpora to extract these large tag-dictionaries is counter to the purpose of type-based semi-supervision. Creating a tag-dictionary from scratch for low-resource languages requires access to native speakers in each language. An alternative proposed by Li et al. (2012) is to use the free and publicly maintained Wiktionary at wiktionary.org.

The Wiktionary is an open dictionary, allowing the public to extend and amend it. As of writing, the Wiktionary hosts actively updated dictionaries in 148 different languages. These dictionaries are steadily being updated and expanded with more words. Li et al. (2012) analyses the type- and token-coverage of the Wiktionary on several languages and finds that while type-coverage varies considerably, especially frequent words are well covered by the Wiktionary. This means that even in low type-coverage languages like Greek and Portuguese, most of the tokens are accounted for by the Wiktionary. The main issue with the Wiktionary is that each language uses its own tagset, and therefore must be mapped to an intermediary tagset, like the Universal tagset

sentence:	The	new	rate	will	be	payable	Feb.	15	.
original:	DT	JJ	NN	MD	VB	JJ	NNP	CD	.
universal:	DET	ADJ	NOUN	VERB	VERB	ADJ	NOUN	NUM	.

Figure 5.1: An example English sentence with the Penn Treebank and the Universal tagsets. Note that NNP and NN have been merged into NOUN and MD and VB have been merged into VERB.

of Petrov et al. (2012), for evaluation. This is a reduced tagset of only 12 general parts-of-speech, figure 5.1 gives an example from the WSJ corpus with the original Penn Treebank tagset and the new Universal Tagset. Furthermore, the downstream applications of the PoS tags may require a specific tagset, and some errors will be introduced by the conversion.

Li et al. (2012) present the wikily-supervised results trained with a first and second order model hidden Markov Model, referred to as HMM and SHMM, respectively. While these models do not improve especially well, the extended models include feature-based Maximum Entropy models for the emission distribution. The HMM-ME and SHMM-ME models use various morphological features to improve tagging accuracy. However, these maximum entropy models must be summed over the entire vocabulary for each tag, slowing the overall training process. This approach takes advantage of the tag-dictionary by limiting the size of the vocabulary for each tag, minimizing the slowing effects of the ME morphology model.

Table 5.7 evaluates the wikily-supervised approach on the same corpora as Li et al. (2012) for comparison. This evaluation is performed on the PoS tagged corpora of nine languages used for the CoNLL-X and CoNNL-2007 dependency parsing shared task. In order to evaluate the accuracy, the tags in the corpora are converted to the Universal tagset with the mappings found online³. The entire corpus is trained on without labels and the accuracy is

³<http://code.google.com/p/wikily-supervised-pos-tagger/>

evaluated against a test set for each language. Table 5.7 shows the accuracies for each of the samplers from Chapter 4 as well as the results reported by Li et al. (2012).

The results in table 5.7 help clarify and expand the conclusions reached in section 5.3. While the type-supervised local sampler without the morphology model performed worse as the number of labelled sentences increase, the wikily-supervised local sampler matches the **type** sampler. This is because the Wiktionary has considerably more coverage than the limited tag-dictionary in section 5.3.2. These results confirm that the local sampler is more sensitive to the coverage of the tag-dictionary than the **type** sampler.

Like the type-supervised experiments in section 5.3.2, the 1HMM samplers performs worse than the Particle Gibbs **type** sampler. As before, the tag-dictionary makes up for the PYP-HMM’s tendency to assign too many tags per word-type. On the other hand, the strict one-tag-per-type restriction of the 1HMM prevents it from taking full advantage of the tag-dictionary. In fact, the local sampler also outperforms the 1HMM sampler in table 5.7. The more extensive data from the wiktionary benefits the local sampler considerably more than the other samplers, eliminating the difference in accuracy observed in Chapter 4. However, the local sampler still gets stuck in local optima with the PYP-HMM+lm, scoring the lowest average accuracy.

The **type**+LM sampler shows the best wikily-supervised performance of the samplers demonstrated in Chapter 4. The average accuracy of the **type**+LM sampler is still lower than that of the SHMM-ME of Li et al. (2012). The SHMM-ME also has more consistently high accuracies across all of the languages, with the highest accuracy in four of the nine languages. The other highest accuracies are split among the PYP-HMM samplers. Each of the local and **type** samplers have uncharacteristically low accuracies for at least one

		Danish	Dutch	German	Greek	English	Italian	Portuguese	Spanish	Swedish	avg.
Li et al.	HMM	71.8%	80.8%	77.1%	73.1%	85.4%	84.6%	79.1%	83.9%	76.7%	79.2%
	HMM-ME	82.8%	86.1%	81.2%	80.1%	86.1%	85.4%	83.7%	84.6%	85.9%	83.9%
	SHMM	74.5%	81.6%	81.2%	73.1%	85.0%	85.2%	79.9%	84.5%	78.7%	80.4%
	SHMM-ME	83.3%	86.3%	85.8%	79.2%	87.1%	86.5%	84.5%	86.4%	86.1%	85.0%
	Local	82.5%	83.7%	83.2%	75.7%	87.6%	83.4%	84.7%	86.0%	86.4%	83.7%
	Local+LM	71.8%	79.3%	77.8%	67.5%	85.7%	82.3%	75.5%	83.3%	80.6%	78.2%
	Type	82.2%	84.1%	80.4%	75.2%	86.2%	87.9%	84.4%	86.3%	86.3%	83.7%
	Type+LM	80.9%	83.4%	83.2%	81.6%	86.2%	87.8%	83.4%	84.7%	85.4%	84.1%
	1HMM	81.0%	84.7%	82.9%	78.4%	81.1%	87.3%	81.4%	84.6%	85.7%	83.0%
	1HMM+LM	80.4%	83.0%	82.3%	79.3%	81.3%	86.8%	79.4%	83.3%	85.8%	82.4%

Table 5.7: Accuracy of wikily-supervised part-of-speech taggers over 9 languages.

language. For example, the `type+LM` sampler score a lower accuracy in Danish than either the non-LM `type` sampler or the local sampler. This leads to an overall lower performance over all of the languages.

5.5 Discussion

Chapter 4 demonstrated the tendency in unsupervised PoS induction for Hidden Markov Models to favour to many tags per type. Even when the Particle Gibbs sampler enabled tractable inference over the entire problem space, the results were worse than the 1HMM one-tag-per-type approximation. This chapter examined one potential solution to the tags per type problem: type-based semi-supervision. Using observed data, rather than the artificial one-tag-per-type assumption, to constrain inference should allow stronger results. Two sources of tag-dictionaries were examined, labelled data and the open on-line dictionary Wiktionary. The results confirm the hypothesis that constraining inference via tag-dictionaries eliminates the need for the one-tag-per-type assumption.

Section 5.3.1 explored the use of labelled data to perform semi-supervised PoS discrimination. The experiments found that unlabelled data can overwhelm impact of the labelled data in the traditional inference task when using non-parametric models. By counting the labelled data as multiple, independent customers in the model’s Chinese Restaurant Franchise, the negative impact of unlabelled data can be minimised. This method outperforms standard unsupervised induction with only a few hundred labelled sentences.

In section 5.3.2, a small amount of labelled data was used to generate a small, but useful, tag-dictionary. This dictionary allowed type-based semi-supervision, with the goal of allowing inference with the PYP-HMM without

resorting to the one-tag-per-type assumption. In this task, the 1HMM actually performs worse than the Particle Gibbs `type` sampler because it is too restrictive to take full advantage of the tag-dictionary. The `type`+LM sampler performs very well with a relatively limited tag-dictionary. The fact that the PG `type` sampler outperforms the 1HMM sampler with a tag-dictionary confirms that the poor performance compared to the 1HMM in Chapter 4 is a result of the HMM assigning too high a likelihood to solutions with too many tags per type.

The next step in section 5.3.3 is to take advantage of the constraints of the tag-dictionary to improve the traditional semi-supervision results. The first observation from this approach is that a much lower weight is necessary to outweigh the unlabelled data. This results from the tag-dictionary limiting the noisiness of the sampler’s assignments for the unlabelled data. This leads to the semi-supervised samplers achieving an accuracy of 95%, compared to the annotator agreement of 97%.

The type-based results are interesting considering how well they compare to other approaches with tag-dictionaries extracted from larger corpora and with lower thresholds. The example in the beginning of section 5.3.2 demonstrates the two key variables that address this trade off when deciding what word-types are added to the lexicon: the size of the corpus and the frequency of the word-types therein. The more frequently a word occurs, the greater the confidence that all of its possible labels have occurred. Similarly, the larger the corpus, the lower the likelihood that a word occurs in some not-yet-encountered context. Of course, if the tag-dictionary is being labelled by humans or when generating tag-dictionaries from labelled corpora, larger corpora require a greater investment. While there is some research on the effects of lexicon quality on performance (Banko and Moore, 2004, Ravi and Knight,

2009), there is still room for a thorough exploration on estimating the quality of a lexicon and the resultant tags.

The benefit of type-based semi-supervision was further confirmed in section 5.4, by experimenting on nine different languages with tag-dictionaries extracted from the Wiktionary. Here the **type**+LM sampler achieved a higher average accuracy across all of the languages than the other PYP-HMM samplers. The excellent coverage of the Wiktionary allowed the local sampler to match the **type** sampler average. However, the tag-dictionary is still not enough to overcome the local+lm sampler’s failure to mix. Once again, the 1HMM sampler performs relatively poorly because it cannot take as much advantage of the tag-dictionary as the **type** and local samplers.

Ultimately, the PYP-HMM samplers do not perform as well as the SHMM-ME model of Li et al. They are somewhat inconsistent across the nine languages, with poor performance in one or two languages bringing down the overall average. Still, the local and **type** samplers combined have the highest accuracies in five of the nine languages. There is room for improvement in the model by allowing it to learn more from the tag-dictionaries and better model morphology.

Toutanova and Johnson (2008) shows very strong type-based accuracy results by incorporating a semi-supervised model of the ambiguity classes that make up a tag-dictionary, or lexicon. This model is able to learn and predict unseen ambiguity classes, generalizing the tag-dictionary to a lexicon that incorporates all of the word-types, including unlabelled ones. Such a model naturally favours limited tags per type through observed data and the design of the model itself. The type-based inference allowed by the Particle Gibbs sampler enables the creation of a generative lexicon model.

This suggests a second solution to the tags per type problem of Chapter

4. Rather than using observed data like labelled corpora or the Wiktionary to constrain the ambiguity class of a word, the PYP-HMM model itself can be extended to incorporate a lexicon. This unsupervised lexicon can use Bayesian priors to have a bias for smaller ambiguity classes. By leveraging the **type** Particle Gibbs sampler, a word-type’s ambiguity class can simply be sampled with its tokens.

The results of this chapter suggest that the best model in a semi-supervised application depends on the coverage of the available tag-dictionary and the trade-off between accuracy and time. In the combined or traditional supervised case, the local+LM sampler matches the performance of the considerably slower **type**+LM sampler and is therefore the superior choice. When there is sufficient type coverage, as in the Wiktionary experiments, the SHMM-ME sampler of Li et al. (Li et al., 2012) has the best average accuracy and the best performance on the most languages. Unfortunately, the maximum entropy morphology it incorporates requires normalisation over the entire vocabulary, making it much slower than the **type**+LM sampler that has the next best average performance. In general, though, the local sampler has the best performance on more languages than the **type**+LM samplers and a very high average performance while easily being the fastest of the examined approaches⁴.

⁴see section 4.4.2 for graphs comparing the training times of the various samplers

Chapter 6

Modelling the Lexicon in Unsupervised Part-of-Speech Induction

Chapter 6 introduces and examines the Lex-HMM model, an extension of the PYP-HMM of Blunsom and Cohn (2011) described in Chapter 3. This chapter is based on research originally published in Dubbin and Blunsom (2014).

6.1 Introduction

Research on the unsupervised induction of part-of-speech (PoS) tags has the potential to improve both our understanding of the plausibility of theories of first language acquisition, and Natural Language Processing applications such as Speech Recognition and Machine Translation. While there has been much prior work on this task (Brown et al., 1992, Clark, 2003, Christodoulopoulos et al., 2010, Toutanova and Johnson, 2008, Goldwater and Griffiths, 2007,

Blunsom and Cohn, 2011), a common thread in many of these works is that models based on a Hidden Markov Model (HMM) graphical structure suffer from a tendency to assign too many different tags to the tokens of a given word type. Models which restrict word types to only occur with a single tag show a significant increase in performance, even though this restriction is clearly at odds with the gold standard labelling (Brown et al., 1992, Clark, 2003, Blunsom and Cohn, 2011). While the empirically observed expectation for the number of tags per word type is close to one, there are many exceptions, e.g. words that occur as both nouns and verbs (*opening*, *increase*, *related* etc.).

On the other hand, the standard Hidden Markov Model (HMM) for unsupervised PoS tagging does not directly model the observed sparsity of the number of tags assigned to tokens of a given word-type. As a result, unrestricted unsupervised PoS models often assign too many tags per type. One solution to this problem is to use semi-supervised tag-dictionaries to restrict the tag assignments for some words. Unfortunately these approaches do not always extend well to out of vocabulary words or they require restricted tag sets to match available semi-supervised data.

The alternative to semi-supervised tag-dictionaries is to explicitly model tags per type sparsity. Hidden Markov Models do not have a parameter to encode sparsity between word-types and part-of-speech. The model must be extended by assigning an ambiguity class representing the tags which can be assigned to tokens of each word-type. Ambiguity classes are used in supervised part-of-speech inference to simplify inference and improve results. Unsupervised ambiguity classes, unfortunately, result in non-local correlations that make standard inference intractable. However, a straightforward extension to the word-type based Particle Gibbs (**type** sampler performs inference without issue.

In this chapter we extend the Pitman-Yor HMM tagger (Blunsom and Cohn, 2011) to explicitly include a model of the lexicon that encodes from which tags a word-type may be generated. For each word-type we draw an ambiguity class which is the set of tags that it may occur with, capturing the fact that words are often ambiguous between certain tags (e.g. *Noun* and *Verb*), while rarely between others (e.g. *Determiner* and *Verb*). We extend the type based Sequential Monte Carlo (SMC) inference algorithm of (Dubbin and Blunsom, 2012) (`type`) to incorporate our model of the lexicon, removing the need for the heuristic inference technique of (Blunsom and Cohn, 2011).

We start in Section 6.3 with a brief overview of the original PYP-HMM model and our extended model of the lexicon. Chapter 3 has a more in depth examination of the PYP-HMM and non-parametric models in general. Section 6.4 introduces a Particle Gibbs sampler for this model, a basic SMC method that generates samples from the model’s posterior. We evaluate these algorithms in Section 6.5, analysing their behaviour in comparisons to previously proposed state-of-the-art approaches.

6.2 Background

We find that modelling the lexicon in part-of-speech inference can lead to more efficient algorithms that match the state-of-the-art unsupervised performance. We also note that the lexicon model relies heavily on morphological information, and suffers without it on languages with flexible word ordering. These results promise further improvements with more advanced lexicon models.

6.3 The Pitman-Yor Lexicon Hidden Markov Model

This article proposes enhancing the standard Hidden Markov Model by explicitly incorporating a model of the lexicon that consists of word types and their associated tag ambiguity classes. The ambiguity class of a word-type is the set of possible lexical categories to which tokens of that type can be assigned. Ambiguity classes have traditionally been used in tag dictionaries for partially supervised PoS induction. Toutanova and Johnson (2008) show that modelling ambiguity class generation significantly improves partially supervised induction with a small supervised tag dictionary. In this work we aim to learn the ambiguity classes unsupervised, i.e. without than have them specified in a tag dictionary.

The Lexicon HMM (Lex-HMM) extends the Pitman-Yor HMM (PYP-HMM) described in the previous chapters. When the ambiguity class of all of the word types in the lexicon is the complete tagset, the two models are the same.

6.3.1 PYP-HMM

The Lex-HMM is an extension of the PYP-HMM of Blunsom and Cohn (2011). This model is covered in detail in Chapter 3. The following is a brief re-examination of the model with some slight alterations to the terminology to simplify the description of the Lex-HMM.

The base of the model applies a hierarchical Pitman-Yor process (PYP) prior to a trigram hidden Markov model to jointly model the distribution of a sequence of latent word tags, $\mathbf{t}$, and word tokens, $\mathbf{w}$. The joint probability defined by the transition, $P(t_l|t_{n-1}, t_{n-2}, \Theta)$, and emission, $P(w_n|t_n, \Theta)$,

distributions of a trigram HMM is

$$P(\mathbf{t}, \mathbf{w} | \Theta) = \prod_{n=1}^{N+1} P(t_n | t_{n-1}, t_{n-2}, \Theta) P(w_n | t_n, \Theta)$$

where $N = |\mathbf{t}| = |\mathbf{w}|$ and the special tag \$ is added to denote the sentence boundaries. The model defines a generative process in which the tags are selected from a transition distribution, $t_l | t_{l-1}, t_{l-2}, T$, determined by the two previous tags in their history, and the word tokens are selected from the emission distribution, $w_l | t_l, E$, of the latest tag.

$$\begin{aligned} t_n | t_{n-1}, t_{n-2}, T &\sim T_{t_{n-1}, t_{n-2}} \\ w_n | t_n, E &\sim E_{t_n} \end{aligned}$$

The PYP-HMM draws the above multinomial distributions from a hierarchical Pitman-Yor Process prior. The Pitman-Yor prior defines a smooth back off probability from more complex to less complex transition and emission distributions. In the PYP-HMM trigram model, the transition distributions form a hierarchy with trigram transition distributions drawn from a PYP with the bigram transitions as their base distribution, and the bigram transitions similarly backing off to the unigram transitions. The hierarchical prior can be intuitively understood to smooth the trigram transition distributions with bigram and unigram distributions in a similar manner to an ngram language model (Teh, 2006b). This back-off structure greatly reduces sparsity in the trigram distributions and is achieved by chaining together the PYPs through

their base distributions:

$$\begin{aligned}
T_{ij}|a^T, b^T, B_i &\sim \text{PYP}(a^T, b^T, B_i) \\
B_i|a^B, b^B, U &\sim \text{PYP}(a^B, b^B, U) \\
U|a^U, b^U &\sim \text{PYP}(a^U, b^U, \text{Uniform}). \\
E_i|a^E, b^E, C &\sim \text{PYP}(a^E, b^E, C_i),
\end{aligned}$$

where T_{ij} , B_i , and U are trigram, bigram, and unigram transition distributions respectively, and C_i is either a uniform distribution (PYP-HMM) or a bigram character language model distribution to model word morphology (PYP-HMM+LM).

Sampling from the posterior of the hierarchical PYP is calculated with a variant of the Chinese Restaurant Process (CRP) called the Chinese Restaurant Franchise (CRF) (Teh, 2006b, Goldwater et al., 2006). In the CRP analogy, each latent variable (tag) in a sequence is represented by a customer entering a restaurant and sitting at one of an infinite number of tables. A customer chooses to sit at a table in a restaurant according to the probability

$$P(z_n = k | \mathbf{z}_{1:n-1}) = \begin{cases} \frac{c_k^- - a}{n-1+b} & 1 \leq k \leq K^- \\ \frac{K^- - a + b}{n-1+b} & k = K^- + 1 \end{cases} \quad (6.1)$$

where z_n is the index of the table chosen by the n th customer to the restaurant, $\mathbf{z}_{1:n-1}$ is the seating arrangement of the previous $n - 1$ customers to enter, c_k^- is the count of the customers at table k , and K^- is the total number of tables chosen by the previous $n - 1$ customers. All customers at a table share the same dish, representing the value assigned to the latent variables. When customers sit at an empty table, a new dish is assigned to that table according

to the base distribution of the PYP. To expand the CRP analogy to the CRF for hierarchical PYPs, when a customer sits at a new table, a new customer enters the restaurant of the PYP of the base distribution.

Blunsom and Cohn (2011) explored two Gibbs sampling methods for inference with the PYP-HMM model. The first individually samples tag assignments for each token. The second employs a tactic shown to be effective by earlier works by constraining inference to only one tag per word-type (PYP-1HMM). However, marginalizing over all possible table assignments for more than a single tag is intractable. Blunsom and Cohn (2011) approximates the PYP-1HMM tag posteriors for a particular sample according to heuristic fractional table counts. This approximation is shown to be particularly inaccurate for values of a close to one.

6.3.2 The Lexicon HMM

We define the lexicon to be the set of all word types (W) and a function ($\mathcal{L}$) which maps each word-type ($W_i \in W$) to an element in the power set of possible tags T ,

$$\mathcal{L} : W \rightarrow 2^T.$$

The Lexicon HMM (Lex-HMM) generates the lexicon with all of the word types and their ambiguity classes before generating the standard HMM parameters. The set of tags associated with each word-type is referred to as its ambiguity class $s_i \subseteq T$. The ambiguity classes are generated from a multinomial distribution with a sparse, Pitman-Yor Process prior,

$$s_i | S \sim S \tag{6.2}$$

$$S | a^S, b^S \sim PYP(a^S, b^S, G) \tag{6.3}$$

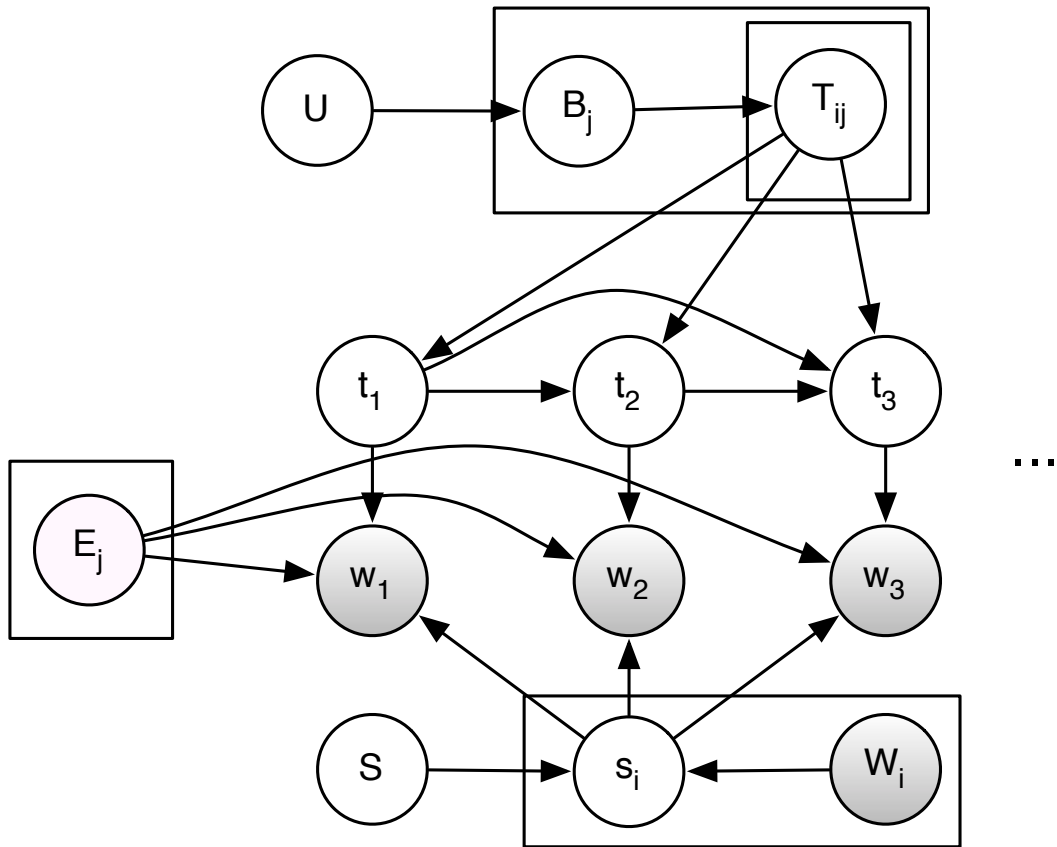


Figure 6.1: **Lex-HMM Structure:** The graphical structure of the Lex-HMM model. In addition to the trigram transition (T_{ij}) and emission (E_j), the model includes an ambiguity class (s_i) for each word-type (W_i) drawn from a distribution S with a PYP prior.

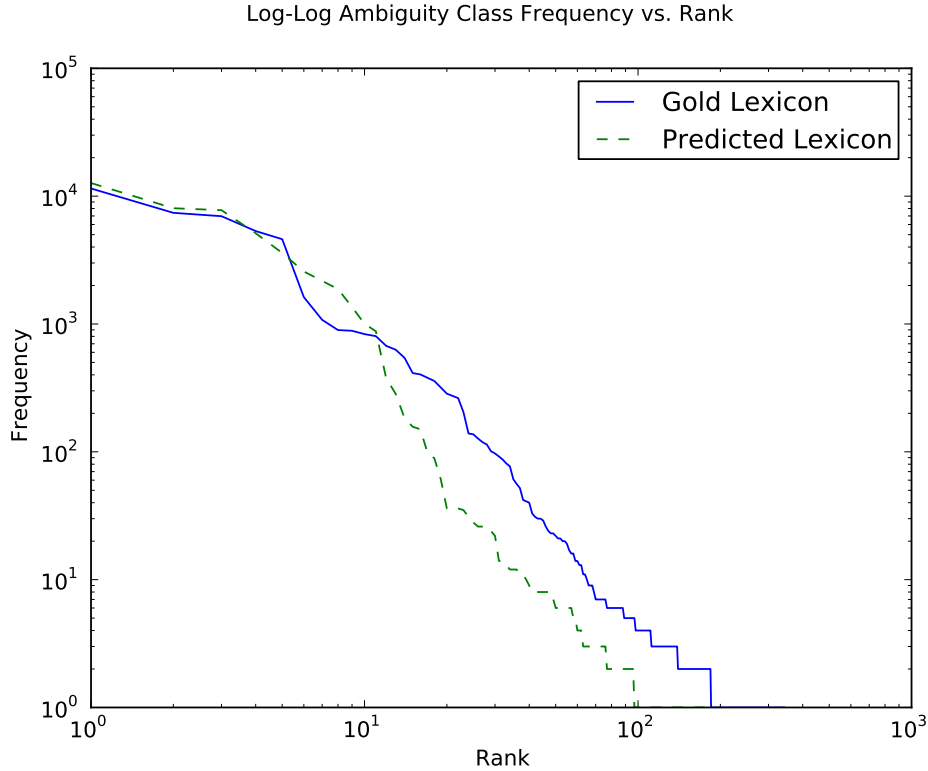


Figure 6.2: **Ambiguity Class Distribution:** Log-log plot of ambiguity class frequency over rank for the Penn-Treebank WSJ Gold Standard lexicon highlighting a Zipfian distribution and the ambiguity of classes extracted from the predicted tags.

where S is the multinomial distribution over all possible ambiguity classes. The base distribution of the PYP, G , chooses the size of the ambiguity class according to a geometric distribution (normalized so that the size of the class is at most the number of tags $|T|$). G assigns uniform probability to all classes of the same size. A plate diagram for this model is shown in Figure 6.1.

This model represents the observation that there are relatively few distinct ambiguity classes over all of the word types in a corpus. For example, the full Penn-Treebank Wall Street Journal (WSJ) corpus with 45 possible tags and 49,206 word types has only 343 ambiguity classes. Figure 6.2 shows that ambiguity classes in the WSJ have a power-law distribution. Furthermore, these classes are generally small; the average ambiguity class in the WSJ corpus

has 2.94 tags. The PYP prior favours power-law distributions and the modified geometric base distribution flavours smaller class sizes.

Once the lexicon is generated, the standard HMM parameters can be generated as described in section 6.3.1. The base emission probabilities C are constrained to fit the generated lexicon. The standard Lex-HMM model emission probabilities for tag t_i are uniform over all word types with t_i in their ambiguity class. The character language model presents a challenge because it is non-trivial to renormalise over words with t_i in their ambiguity class. In this case word types without t_i in their ambiguity class are assigned an emission probability of 0 and the model is deficient.

Neither of the samplers proposed by Blunsom and Cohn (2011) and briefly described in section 6.3.1 are well suited to inference with the lexicon. Local Gibbs sampling of individual token-tag assignments would be very unlikely to explore a range of confusion classes, while the type based approximate sample relies on a one-tag-per-type restriction. Thus in the next section we extend the Particle Filtering solution presented by Dubbin and Blunsom (2012) to the problem of simultaneous resampling the ambiguity class as well as the tags for all tokens of a given type. This sampler provides both a more attractive inference algorithm for the original PYP-HMM and one adaptable to our Lex-HMM.

6.4 Inference

To perform inference with both the lexicon and the tag assignments, we block sample the ambiguity class assignment as well as all tag assignments for tokens of the same word-type. It would be intractable to exactly calculate the probabilities to sample these blocks. Particle filters are an example of a Sequential

Monte Carlo technique which generates unbiased samples from a distribution without summing over the intractable number of possibilities.

The particle filter samples multiple independent sequences of ambiguity classes and tag assignments. Each sequence of samples, called a particle, is generated incrementally. For each particle, the particle filter first samples an ambiguity class, and then samples each tag assignment in sequence based only on the previous samples in the particle. The value of the next variable in a sequence is sampled from a proposal distribution based only on the earlier values in the sequence. Each particle is assigned an importance weight such that a particle sampled proportional to its weight represents an unbiased sample of the true distribution. Algorithm 3 shows the pseudocode for the Lex-HMM Particle Gibbs sampler, it is very similar to algorithm 2 with extensions to account for the sampling of the ambiguity class s^W for word W .

Each particle represents a specific sampling of an ambiguity class, tag sequence, $\mathbf{t}_{1:n}^{W,p}$, and the count deltas, $\mathbf{z}_{1:n}^{W,p}$. The term $\mathbf{t}_{1:n}^{W,p}$ denotes the sequence of n tags generated for word-type W and stored as part of particle $p \in [1, P]$. The count deltas store the differences in the seating arrangement necessary to calculate the posterior probabilities according to the Chinese restaurant franchise described in section 6.3.1. The table counts from each particle are the only data necessary to calculate the probabilities described in equation (6.1).

The particle filter first samples a new proposal ambiguity class for each particle. Then, the particle filter uses the appropriate ambiguity class to sample sequences of seating arrangements in the Chinese Restaurant representation and tag assignments jointly, $\mathbf{x}_{1:n}^W = (\mathbf{t}_{1:n}^W, \mathbf{z}_{1:n}^W)$, for the word-type, W . The final table counts are resampled once a tag assignment has been chosen from the particles. The exchangeability of seating decisions means that only counts of customers are necessary to represent the history.

```

At iteration  $i = 0$  randomly initialise PoS assignments;
for  $i \leftarrow 1$  to Iterations do
  foreach word-type  $W$  observed in the corpus do
    Remove table counts from  $\mathbf{z}_{1:N}^W$ ;
     $s^{W,0} \leftarrow s^W$  ; // from iteration  $i - 1$ 
    for  $p \leftarrow 1$  to  $P$  do
      Sample ambiguity class  $s^{W,p} \sim S$ ; // equation (6.2)
      Set  $\omega_0^{W,p}$  ; // equation (6.4)
    end
    for  $n \leftarrow 1$  to  $N$  do
       $t_n^{W,0} \leftarrow t_n^W$  ; // from iteration  $i - 1$ 
      for  $p \leftarrow 1$  to  $P$  do
        Sample  $t_n^{W,p} \sim q_n^{W,p}(\circ | \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W})$  ; // equation (6.5)
        Update weight  $\omega_n^{W,p} \leftarrow \omega_{n-1}^{W,p} \cdot \alpha_n(t_{1:n}^{W,p})$  ; // equation (4.1)
      end
    end
    Sample  $p \propto \omega_N^{W,p}$ ;
    Set  $s^W \leftarrow s^{W,p}$ ;
    Set  $\mathbf{t}_{1:N}^W \leftarrow \mathbf{t}_{1:N}^{W,p}$ ;
    Sample new table counts  $\mathbf{z}_{1:N}^W$  based on  $\mathbf{t}_{1:N}^W$ ;
  end
end

```

Algorithm 3: Lex-HMM Particle Gibbs Sampler

The ambiguity class for a particle is proposed by sampling from the Pitman-Yor process that defines the lexicon distribution. The particle weights are then set to

$$\omega_0^{W,p} = \frac{P(s_{W,p}|S^{-W})}{\prod_{t \in s_{W,p}} (e_t + 1)^{\#(E_t)} \prod_{t \in T - s_{W,p}} (e_t)^{\#(E_t)}} \quad (6.4)$$

where $P(s_{W,p}|S^{-W})$ is the probability of the ambiguity class proposed for particle p for word-type W given the ambiguity classes for the rest of the vocabulary, e_t is the number of word types with t in their ambiguity class, and $\#(E_t)$ is the number of tables in the CRP for the emission distribution of tag t . The last two terms of the equation correct for the difference in the base probabilities of the words that have already been sampled with a different lexicon.

At each token occurrence n , the next tag assignment, $t_n^{W,p}$ for each particle $p \in [1, P]$ is determined by the seating decisions $z_n^{W,p}$, which are made according to the proposal distribution:

$$\begin{aligned} q_n^{W,p}(z_n^{W,p} | \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W}) &\propto \\ &P(z_n^{W,p} | c_n^{-2}, c_n^{-1}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W}) \\ &\times P(c_n^{+1} | c_n^{-1}, z_n^{W,p}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W}) \\ &\times P(c_n^{+2} | z_n^{W,p}, c_n^{+1}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W}) \\ &\times P(w_n^W | z_n^{W,p}, \mathbf{z}_{1:n-1}^{W,p}, \mathbf{z}^{-W}). \end{aligned} \quad (6.5)$$

In this case, $c_n^{\pm k}$ represents a tag in the context of site t_n^W offset by k , while $\mathbf{z}_{1:n-1}^{W,p}$ and $\mathbf{z}^{-W}$ represent the table counts from the seating decisions previously chosen by particle p and the values at all of the sites where a word token of type W does not appear, respectively. This proposal distribution ignores changes to the seating arrangement between the three transitions involving the site n . The specific tag assignment, t_n^W, p_n , is completely determined by the seating decisions sampled according to this proposal distribution. Once all

of the particles have been sampled, one of them is sampled with probability proportional to its weight. This final sample is a sample from the target distribution.

6.5 Experiments and Results

I provide an empirical evaluation of our proposed Lex-HMM in terms of the accuracy of the taggings learned according to the most popular metric, and the distributions over ambiguity classes. This experimental evaluation considers the impact of the improved Particle Gibbs inference algorithm both for the original PYP-HMM and when used for inference in our extended model.

The goal is to learn whether the lexicon model can match or exceed the performance of the other models despite focusing on only a subset of the possible tags each iteration. I hypothesize that an accurate lexicon model and the sparsity it induces over the number of tags per word-type will improve the performance over the standard PYP-HMM model while also decreasing training time. Furthermore, the lexicon model is novel, and its accuracy in representing ambiguity classes is an important aspect of its performance. The model focuses inference on the most likely tag choices, represented by ambiguity classes.

6.5.1 Unsupervised Part-of-Speech Tagging

This section evaluates the Lex-HMM according to the many-to-one accuracy metric, as described in previous chapters. For experiments on English, the evaluation reports results on the entire Penn. Treebank (Marcus et al., 1993b). For other languages we use the corpora made available for the CoNLL-X Shared Task (Buchholz and Marsi, 2006). All Lex-HMM results are reported with 10 particles as no significant improvement was found with 50 particles.

Sampler	M-1 Accuracy	Time (h)
Meta-Model (CGS10)	76.1	—
MEMM (BBDK10)	75.5	~40*
Lex-HMM	71.1	7.9
Type PYP-HMM	70.1	401.2
Local PYP-HMM	70.2	8.6
PYP-1HMM	75.6	20.6
Lex-HMM+LM	77.5	16.9
Type PYP-HMM+LM	73.5	446.0
PYP-1HMM+LM	77.5	34.9

Table 6.1: **M-1 Accuracy on the WSJ Corpus:** Comparison of the accuracy of each of the samplers with and without the language model emission prior on the English WSJ Corpus. The second column reports run time in hours where available*. Note the Lex-HMM+LM model matches the PYP-1HMM+LM approximation despite finishing in half the time. The abbreviations in parentheses indicate that the results were reported in CGS10 (Christodoulopoulos et al., 2010) and BBDK10 (Berg-Kirkpatrick et al., 2010) *CGS10 reports that the MEMM model takes approximately 40 hours on 16 cores.

Table 6.1 compares the M-1 accuracies of both the PYP-HMM and the Lex-HMM models on the Penn. Treebank Wall Street Journal corpus. Blunsom and Cohn (2011) found that the Local PYP-HMM+LM sampler is unable to mix, achieving accuracy below 50%, therefore it has been left out of this analysis. The Lex-HMM+LM model achieves the same accuracy as the state-of-the-art PYP-1HMM+LM approximation. The Lex-HMM+LM’s focus on only the most likely tags for each word-type allows it to finish training in half the time as the PYP-1HMM+LM approximation without any artificial restrictions on the number of tags per type. This contrasts with other approaches that eliminate the constraint at a much greater cost, e.g. the **type** PYP-HMM, the MEMM, and the Meta-Model ¹

The left side of table 6.2 compares the M-1 accuracies of the Lex-HMM

¹While I was unable to get an estimate on the runtime of the Meta-Model, it uses a system similar to the feature-based system of the MEMM with an additional feature derived from the proposed class from the brown model. Therefore, it is likely that this model has a similar runtime.

Language	Lex-HMM	PYP-HMM	Local	IHMM	Lex-HMM+LM	PYP-HMM+LM	IHMM+LM
WSJ	71.1	70.1	70.2	75.6	77.5	73.5	77.5
Arabic	57.2	57.6	56.2	61.9	62.1	62.7	62.0
Bulgarian	67.2	67.8	67.6	71.4	72.7	72.1	76.2
Czech	61.3	61.6	64.5	65.4	68.2	67.4	67.9
Danish	68.6	70.3	69.1	70.6	74.7	73.1	74.6
Dutch	70.3	71.6	64.1	73.2	71.7	71.8	72.9
Hungarian	57.9	61.8	64.8	69.6	64.4	69.9	73.2
Portuguese	69.5	71.1	68.1	72.0	76.3	73.9	77.1
Spanish	73.2	69.1	68.5	74.7	80.0	75.2	78.8
Swedish	66.3	63.5	67.6	67.2	70.4	67.6	68.6
Average	66.3 ^(67.2)	66.5 ^(67.0)	66.1 ^(66.2)	70.2 ^(70.3)	71.8 ^(72.6)	70.7 ^(70.8)	72.9 ^(72.9)

Table 6.2: **M-1 Accuracy of Lex-HMM and PYP-HMM models:** Comparison of M-1 accuracy for the lexicon based model (Lex-HMM) and the PYP-HMM model on several languages. The Lex-HMM and PYP-HMM columns indicate the results of word-type based particle filtering with 10 and 100 particles, respectively, while the Local and IHMM columns use the token based sampler and the IHMM approximation described by Blunsom and Cohn (2011). The token based sampler was run for 500 iterations and the other samplers for 200. The percentages in brackets represent the average accuracy over all languages except for Hungarian.

model to the PYP-HMM model. These models both ignore word morphology and rely on word order. The 1HMM approximation achieves the highest average accuracy. The Lex-HMM model matches or surpasses the **type** PYP-HMM approach in six languages while running much faster due to the particle filter considering a smaller set of parts-of-speech for each particle. However, in the absence of morphological information, the Lex-HMM model has a similar average accuracy to the local and **type** PYP-HMM samplers. The especially low performance on Hungarian, a language with free word ordering and strong morphology, suggests that the Lex-HMM model struggles to find ambiguity classes without morphology. The Lex-HMM model has a higher average accuracy than the **type** or local PYP-HMM samplers when Hungarian is ignored.

The right side of table 6.2 compares the M-1 accuracies of the Lex-HMM+LM model to the PYP-HMM+LM. The language model leads to consistently improved performance for each of the samplers excepting the token sampler, which is unable to mix properly with the additional complexity. The accuracies achieved by the 1HMM+LM sampler represent the previous state-of-the-art. These results show that the Lex-HMM+LM model achieves state-of-the-art M-1 accuracies on several datasets, including the English WSJ. The Lex-HMM+LM model performs nearly as well as, and often better than, the 1HMM+LM sampler without any restrictions on tag assignments.

The drastic improvement in the performance of the Lex-HMM model reinforces our hypothesis that morphology is critical to the inference of ambiguity classes. Without the language model representing word morphology, the distinction between ambiguity classes is too ambiguous. This leads the sampler to infer an excess of poor ambiguity classes. For example, the tag assignments from the Lex-PYP model on the WSJ dataset consist of 660 distinct ambiguity classes, while the Lex-PYP+LM tag assignments only have 182 distinct

ambiguity classes.

Note that while the Lex-HMM and Lex-HMM+LM samplers do not have any restrictions on inference, they do not sacrifice time. The additional samples generated by the particle filter are mitigated by limiting the number of tags each particle must consider. In practice, this results in the Lex-HMM samplers with 10 particles running in half time as the 1HMM samplers. The Lex-HMM+LM sampler with 10 particles took 16.9 hours, while the 1HMM+LM sampler required 34.9 hours. Furthermore, the run time evaluation does not take advantage of the inherent distributed nature of particle filters. Each of the particles can be sampled completely independently from the others, making it trivial to run each on a separate core.

6.5.2 Lexicon Analysis

While section 6.5.1 demonstrates that the Lex-HMM+LM sampler performs similarly to the more restricted 1HMM+LM, this section also seeks to evaluate the accuracy of the lexicon model itself. This analysis compares the ambiguity classes extracted from the gold standard and predicted tag assignments of the WSJ corpus. The analysis also explores the relationship between the actual and sampled ambiguity classes.

The solid curve in figure 6.2 shows the distribution of the number of word types assigned to each ambiguity set extracted from the gold standard tag assignments from the Penn Treebank Wall Street Journal corpus. The straight line strongly indicates that ambiguity classes follow a Zipfian distribution. Figure 6.2 also graphs the distribution of the ambiguity classes extracted from the best tag-assignment prediction from the model. The predicted graph has a similar shape to the gold standard but represents half as many distinct ambiguity classes - 182 versus 343.

Rank	Gold Rank	Tags	Top Word Types
1	1	NNP	Mr. , Corp. (1), Inc. (.99), Co. (1), Exchange (.99)
2	2	NN	% (1), company , stock (.99), -RRB- (0), years (0)
3	3	JJ	new , other , first (.9), most (0), major (1)
4	5	NNS	companies , prices (1), quarter (0), week (0), investors
5	4	CD	\$ (0), million (1), billion , 31 , # (0)
15	303	NN, CD	yen (.47, 0), dollar (1, 0), 150 (0, 1), 29 (0, 1), 33 (0, 1)
16	17	VB, NN	plan (.03, .9), offer (.2, .74), issues (0, 0), increase (.34, .66), end (.18, .81)
17	115	DT, NNP	As (0, 0), One (0, .01), First (0, .82), Big (0, .91), On (0, .01)
18	11	NN, JJ	market (.99, 0), U.S. (0, 0), bank (1, 0), cash (.98, 0), high (.06, .9)
20	22	VBN, JJ	estimated (.58, .15), lost (.43, .03), failed (.35, .04), related (.74, .23), reduced (.57, .12)

Table 6.3: **Selection of Predicted Ambiguity Classes:** Common ambiguity classes from the predicted part-of-speech assignments from the WSJ data set, and the five most common word types associated with each ambiguity class. The sets are ranked according to the number of word types associated to them. Words in bold are matched to exactly the same ambiguity set in the gold standard. The lower five ambiguity classes are the most common with more than one part-of-speech. Numbers in parentheses represent the proportion of tokens of that type assigned to each tag in the gold standard for that ambiguity class.

For a qualitative analysis of the generated lexicon, table 6.3 lists frequent ambiguity classes and the most common words assigned to them. The 14 most frequent ambiguity classes contain only one tag each, the top half of table 6.3 shows the five most frequent. One third of the word-types in the first five rows of the table are exactly matched with the ambiguity classes from the gold standard. Most of the remaining words in those rows are assigned to a class representing almost all of the words' occurrences in the gold standard, e.g., 'Corp.' is an NNP in 1514 out of 1521 occurrences. Some words are assigned to classes with similar parts of speech, e.g. {NNS} rather than {NN} for week.

The lower half of table 6.3 shows the most frequent ambiguity classes with more than a single tag. The words assigned to the {NN,CD}, {DT,NNP}, and {NN,JJ} classes are not themselves ambiguous. Rather words that are unambiguously one of the two tags are often assigned to an ambiguity class with both. The most common types in the {NN, CD} set are unambiguously either NN or CD. In many cases the words are merged into broader ambiguity classes because the Lex-HMM+LM uses the language model to model the morphology of words over individual parts-of-speech, rather than entire ambiguity classes. Therefore, a word-type is likely to be assigned a given ambiguity class as long as at least one part-of-speech in that ambiguity class is associated with morphologically similar words. These results suggest modifying the Lex-HMM+LM to model word morphology over ambiguity classes rather than parts-of-speech.

The {VB,NN} and {VBN,JJ} are representative of true ambiguity classes. Occurrences of words in these classes are likely to be either of the possible parts-of-speech. These results show that the Lex-HMM is modelling ambiguity classes as intended.

6.6 Discussion

This chapter introduces and explores the Lex-HMM, an expansion to the PYP-HMM that incorporates a lexicon into the generative model in order to provide a soft constraint on the number of tags per type. Section 6.3 explains the design of the Lex-HMM, which generates ambiguity classes for each word-type and restricts the emission distributions of the PYP-HMM accordingly. This model is tightly coupled on the type level, rather than the token level, making the Particle Gibbs `type` sampler from Chapter 4 instrumental to inference under the model.

Section 6.4 demonstrates the specific Particle Gibbs algorithm implementation for the Lex-HMM. The algorithm relies on type-based sampling to allow the sampler to resample a word-type’s ambiguity class before it assigns any tags to tokens of that type. This greatly simplifies the posterior probability calculation of the ambiguity class sample. A secondary benefit of this approach is that each particle will only be sampling tags from the limited ambiguity class for its type rather than the full tagset for a considerable speed-up.

The first result in section 6.5.1 shows that the unsupervised Lex-HMM+LM matches the performance of the one-tag-per-type 1HMM+LM in half the time on the English WSJ corpus. This result confirms the hypothesis that the soft constraint of the Lex-HMM makes the model favour a more empirically reasonable number of tags per type. Furthermore, efficient inference can be performed on this model without necessitating unjustified constraints and approximations. These speed improvements fail to capture the additional benefit of the inherent ability of the Particle Gibbs sampler to be parallelised.

The next result shows that on average, the Lex-HMM matches the 1HMM approximation across all languages other than Hungarian. Hungarian is a particularly morphologically rich language, and the poor performance of the Lex-

HMM+LM on Hungarian suggests that the model has room for improvement in its model of morphology. The Lex-HMM+LM model surpasses or matches the state-of-the-art results of the 1HMM+LM in 5 out of the 10 languages.

Finally, section 6.5.2 explores the quantitative performance of the Lex-HMM by examining some of the ambiguity classes the model generates. This intention is achieved in several of the examples demonstrated, like plan {VB,NN} and related {VBN,JJ}. On the other hand, the {NN,CD} ambiguity class incorrectly demonstrates a class of similarly distributed word-types that are completely unambiguous in their correct tagging, e.g. dollar is always an NN and 150 is always a CD. This quantitative analysis reveals that while the qualitative results are quite compelling, the lexicon model of the Lex-HMM has considerable room for improvement. The lexicon model could potentially be improved by incorporating morphology or refining its ambiguity class generation to reuse common tags more often.

Chapter 7

Conclusion

This thesis explored the implementation and effects of the application of particle filtering techniques to the problem of unsupervised part-of-speech induction. I show that the Particle Gibbs algorithm enables type-based inference for unsupervised part-of-speech induction without the simplifying one-tag-per-type assumption. The approach is expanded with the inclusion of a semi-supervised lexicon. The thesis culminated with the creation of the Lex-HMM, a new model of part-of-speech that explicitly models the lexicon without supervision. The Lex-HMM is able to match previous state-of-the-art results in all but a few languages tested while being faster than previous work without relying on the one-tag-per-type simplification. These results demonstrate that not only does Particle Gibbs enable tractable inference on existing models that are too intricate for other approaches, it also enables the creation of even more advanced models that would otherwise prove infeasible.

In its exploration of the history of part-of-speech and part-of-speech tagging, Chapter 2 defines unsupervised part-of-speech induction and the problems associated with it. It found that much of the difficulty in defining the parts of speech results from the flawed assumption that word-types can be

partitioned into simple classes that exactly represent the syntactic context in which they occur. Instead, words-classes and syntactic categories should inform each other without being equated. Linguists represent this separation with a lexicon that describes the possible syntactic behaviours of words. Similarly, part-of-speech taggers assign codes describing the grammatical function of a specific word-token in its particular context. The chapter also discussed the similarity between the flawed traditional approach to part-of-speech and the one-tag-per-type assumption often relied on to simplify unsupervised part-of-speech induction.

Chapter 3 describes the motivation and formulation of hierarchical non-parametric Bayesian models. Bayesian inference allows the encoding of ambiguity regarding model parameters that are related to each other. Non-parametric models use priors like the Dirichlet and Pitman-Yor processes to describe some of these relationships without knowledge of the number or order of the parameters themselves. Hierarchical non-parametric models further extend this concept by sharing non-parametric base distributions of priors, allowing intricate, interconnected models that back-off to simpler ones when less data is available. Chapter 3 concludes with a description of the PYP-HMM, a hierarchical non-parametric hidden Markov Model of part-of-speech. This model achieves state-of-the-art results, but requires the one-tag-per-type assumption for tractable inference. Performing tractable inference on this model without the one-tag-per-type assumption with Particle Gibbs demonstrates the viability of hierarchical non-parametric models, and other similarly complex models, in other linguistic applications.

Chapter 4 aims to perform type-sampling on the PYP-HMM with Particle Gibbs sampling allowing tractable inference without the flawed one-tag-per-type assumption. It describes two implementations of the Particle Gibbs

algorithm, one that block samples sentences and another that samples based on word-type. The different samplers are compared to each other according to the likelihood of the assignments they find under the same models. As expected, the sentence PG sampler has almost the same performance as the forward-backward sentence sampler with sufficient particles, while being slightly faster. Similarly, the type PG sampler achieves higher likelihoods than the 1HMM sampler. These results confirm that the Particle Gibbs algorithm enables tractable type-based inference without resorting to the one-tag-per-type assumption.

However, the Particle Gibbs algorithm has a considerable computational cost, so while the type-based PG sampler is not prohibitively slow, it still significantly slower than the 1HMM sampler. To address this issue, Chapter 4 also shows promising results when mixing the local and type-based samplers. These samplers only occasionally rely on the slower particle filter, matches the performance of the pure Particle Gibbs sampler. In this implementation, the type-based sampler takes large steps through the sample space and the local sampler takes smaller steps to perform inference around the new space.

An interesting result from Chapter 4 comes from the discrepancy between the likelihoods and the accuracy of the tag assignments found by each sampler. While the PG type-sampler found more likely tag assignments, the 1HMM outperforms all of the other samplers when their assignments are compared against a gold-standard, including the type PG sampler. This implies that the one-tag-per-type restriction of the 1HMM prevents it from exploring higher likelihood assignments that have too many tags-per-type and lower accuracy. Therefore, the PYP-HMM model assigns too high of a probability to tag assignments with more tags assigned to words of the same type than empirical data supports. Chapters 5 and 6 examine two proposed solutions to this prob-

lem: semi-supervised tag-dictionaries, and an unsupervised lexicon model.

Chapter 5 explores the related task of semi-supervised part-of-speech discrimination with a tag-dictionary. The chapter first explores adapting the unsupervised samplers to the traditional semi-supervised task with a small amount of labelled data and a larger amount of unlabelled data. These experiments find that the unlabelled data do not improve the results, instead they overwhelm the labelled data without modifying the algorithm to heavily weight the labelled data. Then, Chapter 5 uses some labelled data to extract a tag-dictionary for type-based supervision. The local sampler benefits considerably from this approach, matching the performance of the 1HMM and type PG samplers. Finally, the two approaches are combined to achieve relatively high accuracy with only a small number of labelled sentences.

The exploration of type-based supervision continues with a more extensive tag-dictionary extracted from the freely available online Wiktionary. These results largely replicate the smaller scale results from before. With an extensive semi-supervised lexicon, the 1HMM approximation is hurt by its one-tag-per-type assumption because the tags are already limited to the correct number. While the type and local samplers do not perform as well as the SHMM-ME of Li et al. (2012), they come close with considerably less computational overhead. The local sampler is especially impressive in this regard as it is so much faster than the other top approaches while achieving the best accuracy in multiple languages.

Finally, Chapter 6 addresses the problem of too many tags-per-type in the fully unsupervised induction task with the incorporation of a lexicon into the PYP-HMM. The lexicon acts as a soft constraint on the number of tags assigned to each type. The lexicon uses a fairly simple model to generate an ambiguity class for each word-type, restricting the tags which can be assigned

to tokens of that type. The ambiguity class model prefers a small number of tags, thus counteracting the tendency of the base model to over-assign tags per type. This model has even stronger long distance correlations between tokens of the same type, requiring the type-based Particle Gibbs sampler for tractable inference. Chapter 6 presents the Lex-HMM model and the modified type PG sampler algorithm that can perform inference on it.

The first result from experiments with the Lex-HMM show that it can match the state-of-the-art performance of the 1HMM approximation while being twice as fast. The fact that a word’s ambiguity class is a random variable in the model restricts inference to only those tags that it predicts might be assigned to a given word, resulting in a considerable speed-up. The accuracy of the Lex-HMM matches or surpasses the 1HMM approximation in several different languages. However, the Lex-HMM scores a very low accuracy on the Hungarian test set, suggesting that it needs an improved morphology model for such a morphologically rich language. Further quantitative analysis of the predicted ambiguity classes show that many represent words that might have either tags, but some are simply a combination of otherwise unambiguous words. Overall, the Lex-HMM replicates the performance of the 1HMM without requiring any artificial restrictions.

7.1 Future Work

The work presented here merits many avenues of further research. Many of the results hint at future extensions in the task of part-of-speech induction and disambiguation. Such extensions to this work could result in even further improvements, and should be investigated. Furthermore the successful application of Particle Gibbs and particle techniques in general suggest applications

to a variety of other NLP problems.

While the Lex-HMM model naturally speeds up inference by eliminating unlikely possibilities, there is still room for improving the speed. An interesting option is to replicate the mixed samplers explored in Chapter 4 with the Lex-HMM from Chapter 6. In this formulation, the local sampler would find likely tag assignments within the constraints of the lexicon. The type-sampler steps would sample new ambiguity classes for the lexicon. Chapter 4 suggested that the mixed sampler could match the performance of the pure PG sampler, thus potentially resulting in further speed improvements without dropping performance. Furthermore, the constrained local sampler would be even faster than in the standard PYP-HMM model, because it is still only sampling from the tags in each word’s ambiguity class.

In addition, the lexicon model in the Lex-HMM is relatively simple compared to those proposed by linguists. Expanding the lexicon to improve its model of ambiguity classes or incorporate additional information may further improve speed and accuracy. Furthermore, the qualitative results from Chapter 6 show that there is still room for improvement of the lexicon model. The strength of these results with a fairly simple lexicon model promise to only get stronger as research continues.

For example, a common factor in the history of part-of-speech and the early part-of-speech taggers is the reliance on morphology. The first automatic tagger of Klein and Simmons (1963) used morphology to determine the ambiguity class of most words, reducing the size of the lexicon. Similarly, the semi-supervised model of ambiguity class of Toutanova and Johnson (2008) also relied on morphological features to predict unknown ambiguity classes. The same approach should work with the Lex-HMM, moving the morphological model from the emission distribution to the lexicon model. This would syn-

ergize well with the mixed sampler approach, keeping the morphology model separate from the local sampler.

Another aspect of the parts of speech is their general separation into *closed* and *open* classes (Haspelmath, 2001). Most parts of speech are closed, meaning that only a small, finite set of words can be assigned to each. On the other hand, a few parts of speech are assigned to many different words, with new words constantly added over time. The PYP prior on ambiguity class generation only models the tendency for whole ambiguity classes to experience a power-law distribution over word-types, not the individual tags themselves. Rather than the simple geometric distribution, the base distribution of the ambiguity class model could represent this tendency for tags to be divided between a few large clusters and several smaller ones. For example, if the each tag were added independently according to a Bernoulli distribution with a beta prior, then the ambiguity classes would tend to have common (open) tags and would rarely have uncommon (closed) tags.

Such improvements to the lexicon model would translate to better predictions with a semi-supervised lexicon. Toutanova and Johnson (2008) found that modelling ambiguity classes improved semi-supervised part-of-speech disambiguation. Furthermore, Hasan and Ng (2009) used graphical methods to predict ambiguity classes from a small supervised lexicon, bootstrapping part-of-speech tagging for low-resource languages. Combining Particle Gibbs with hierarchical non-parametric priors to create a strong model of lexicon generation should produce similar results.

Beyond the unsupervised part-of-speech induction, the results of this thesis suggest that particle methods can be applied to a variety of problems in NLP. Others have implemented particle filters in online contexts, leveraging their sequential nature (Canini et al., 2009, Borschinger and Johnson, 2011). Ad-

ditionally, the non-local correlations found in the PYP-HMM and Lex-HMM occur in many hierarchical non-parametric models. This thesis has shown that particle filters are especially well-suited to inference with such models.

To extend these results to other applications, first note that the particle filtering approach itself is applicable outside of the Particle Gibbs algorithm. For example, the weighted average of some function over all particles is an unbiased approximation of the expected value of that function. This can be used in many applications, supervised and unsupervised, that require the calculation of expected value. Additionally, many algorithms, like Gibbs samplers, are often dependent on their initialisation, with results varying depending on the quality of the initial values. Particle filters do not require initialisation, and are therefore good choices for generating the initial iteration for other algorithms.

On the other hand, many applications can take advantage of the continuing advances to the basic Particle Gibbs algorithm presented in this thesis. While the Particle Gibbs sampler is very powerful, it wastes a large amount of computational power generating many particles but ignoring all but one. Since the development of the Particle Gibbs algorithm, much research has focused on reducing the number of particles that need to be generated each iteration by leveraging previous iterations' particles or maximizing the information from the current iteration's particles. One such approach particle filters backward over the generated particles, increasing the diversity of the final selection (Lindsten and Schon, 2012). Another, called Ancestor Sampling, resamples previously generated particles in the new iteration, so that the same particles do not have to be sampled each iteration (Lindsten et al., 2014).

The Particle Gibbs algorithm can also be applied to any other problem with long-range correlations. Dependency parsing often depends on decisions

made by the parser much earlier in the sentence. Word alignment models in bilingual parallel corpora generally assume that a word is commonly aligned to the same word it has been aligned to in other sentences, similar to the tendency of words to be assigned the same tag in PoS induction. Another potential application is reference resolution, where a pronoun’s antecedent might be in a different sentence. Any application where variables are dependent on non-local information might benefit from Particle Gibbs sampling.

Furthermore, many NLP applications are strongly connected to each other, but approached independently for practical reasons. For example, part-of-speech tagging is closely related to morphological analysis of words and grammatical analysis of sentences. This suggests that they might benefit from joint modelling, allowing knowledge from one aspect of the model to inform another, like knowing if the word ”to” is linked to a verb or a noun to determine if it marks an infinitive or a preposition. Christodoulopoulos (2013) implements an iterative framework where morphological analysis and part of speech tagging are each used to improve the other. This idea is similar to the approach of this thesis to jointly model the lexicon and the assignment of parts of speech. Bohnet et al. (2013) shows that jointly performing morphological analysis, part-of-speech tagging, and dependency parsing leads to improvements in morphologically rich languages. Particle filtering over such a joint model would allow the propagation of uncertainty until it can be resolved.

Bibliography

- J. M. Anderson. *The Grammar of Case: Towards a Localistic Theory*. Cambridge University Press, Cambridge, 1971.
- J. M. Anderson. *A Notional Theory of Syntactic Categories*. Cambridge University Press, Cambridge, 1997.
- Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal Of The Royal Statistical Society Series B*, 72(3):269–342, 2010. URL <http://ideas.repec.org/a/bla/jorssb/v72y2010i3p269-342.html>.
- P. Baker. Consistency and accuracy in correcting automatically tagged data. In R. Garside, G. Leech, and A. McEnery, editors, *Corpus Annotation: Linguistic Information from Computer Text Corpora*, pages 243–250. Longman, London, 1997.
- M. Banko and R Moore. A study of unsupervised part-of-speech tagging. *Proceedings of COLING '04*, 2004.
- L. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3: 1–8, 1972.
- Matthew Beal. *Variational Algorithms for Approximate Bayesian Inference*.

PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

Andrew David Beale. Grammatical analysis by computer of the lancaster-oslo/bergen (lob) corpus of british english texts. In *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics*, ACL '85, pages 293–298, Stroudsburg, PA, USA, 1985. Association for Computational Linguistics. doi: 10.3115/981210.981246. URL <http://dx.doi.org/10.3115/981210.981246>.

Mark A. Beaumont. Estimation of Population Growth or Decline in Genetically Monitored Populations. *Genetics*, 164(3):1139–1160, July 2003. URL <http://www.genetics.org/content/164/3/1139.abstract>.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June 2010. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N10-1083>.

L. Bloomfield. *Language*. Holt, New York, 1933.

Phil Blunsom and Trevor Cohn. A hierarchical Pitman-Yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1087>.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter,

and Jan Hajic. Joint morphological and syntactic analysis for richly inflected languages. *TACL*, 1:415–428, 2013.

Benjamin Borschinger and Mark Johnson. A particle filter algorithm for bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia, December 2011. URL <http://www.aclweb.org/anthology/U/U11/U11-1004>.

Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.

Eric Brill. Some advances in transformation-based part of speech tagging. *arXiv preprint cmp-lg/9406010*, 1994.

Eric Brill. Unsupervised learning of disambiguation rules for part of speech tagging. In *In Natural Language Processing Using Very Large Corpora*, pages 1–13. Kluwer Academic Press, 1995.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18:467–479, December 1992. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=176313.176316>.

Sabine Buchholz and Erwin Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Morristown, NJ, USA, 2006. Association for Computational Linguistics. URL <http://portal.acm.org/citation.cfm?id=1596276.1596305>.

- Kevin R. Canini, Lei Shi, and Thomas L. Griffiths. Online inference of topics with latent Dirichlet allocation. In David van Dyk and Max Welling, editors, *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS*09)*, pages 65–72, 2009.
- O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Adaptive computation and machine learning. MIT Press, Cambridge, MA, USA, September 2006. URL <http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11015>.
- N. Chomsky. *The Amherst Lectures*. Documents linguistiques. Département de Recherches Linguistiques, Univ. de Paris, 1975a. URL <http://books.google.com/books?id=J0yJHAAACAAJ>.
- Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.
- Noam Chomsky. *Aspects of the Theory Syntax*. The MIT Press, Cambridge, MA, 1965.
- Noam Chomsky. *The Logical Structure of Linguistic Theory*. Plenum Press, New York, 1975b.
- Christos Christodoulopoulos. *An Iterated Learning Framework for Unsupervised Part-of-Speech Induction*. PhD thesis, The University of Edinburgh, 2013.
- Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584, Cambridge, MA, October 2010. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D10-1056>.

- Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143. Association for Computational Linguistics, 1988.
- Alexander Clark. Inducing syntactic categories by context distribution clustering. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 91–94. Association for Computational Linguistics, 2000.
- Alexander Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth Annual Meeting of the European Association for Computational Linguistics (EACL)*, pages 59–66, 2003. URL [papers/eacl2003.pdf](http://papers.eacl2003.pdf).
- Trevor Cohn, Sharon Goldwater, and Phil Blunsom. Inducing compact but accurate tree-substitution grammars. In *HLT-NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 548–556, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1.
- R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.
- Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 133–140, Stroudsburg, PA,

USA, 1992. Association for Computational Linguistics. doi: 10.3115/974499.974523. URL <http://dx.doi.org/10.3115/974499.974523>.

Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL <http://dl.acm.org/citation.cfm?id=2002472.2002549>.

T. Davidson. The grammar of dionysius thrax: Translated from the greek. *The Journal Speculative Philosophy*, 8:326–39, 1874.

F. C. Diez. *Grammatik der romanischen Sprachen*. Eduard Weber, Bonn, 1836-38.

R. M. W. Dixon. Where have all the adjectives gone? *Studies in Language*, 1: 19–80, 1977.

Arnaud Doucet and Adam M. Johansen. *A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later*. Oxford University Press, 2009. URL <http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/johansen/publications/DJ09.pdf>.

Gregory Dubbin and Phil Blunsom. Unsupervised bayesian part of speech inference with particle gibbs. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *ECML/PKDD (1)*, volume 7523 of *Lecture Notes in Computer Science*, pages 760–773. Springer, 2012. ISBN 978-3-642-33459-7. URL <http://dblp.uni-trier.de/db/conf/pkdd/pkdd2012-1.html#DubbinB12>.

Gregory Dubbin and Phil Blunsom. Modelling the lexicon in unsupervised part of speech induction. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, EACL. Association for Computational Linguistics, 2014.

David Elworthy. Does baum-welch re-estimation help taggers? In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 53–58, Stuttgart, Germany, October 1994. Association for Computational Linguistics. doi: 10.3115/974358.974371. URL <http://www.aclweb.org/anthology/A94-1009>.

Paul Fearnhead, David Wyncoll, and Jonathan Tawn. A sequential smoothing algorithm with linear computational cost. Technical report, Department of Mathematics and Statistics, Lancaster University, 2008.

Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

Jesus Fernandez-Villaverde and Juan F. Rubio-Ramirez. Estimating macroeconomic models: A likelihood approach. *Review of Economic Studies*, 74(4): 1059–1087, October 2007. URL <http://ideas.repec.org/a/bla/restud/v74y2007i4p1059-1087.html>.

W.N. Francis. *A standard sample of present-day English for use with digital computers*. Cooperative research project. Brown University, 1964. URL <http://books.google.com/books?id=KupWAAAAMAAJ>.

W.N. Francis and H. Kučera. *Manual of Information to Accompany A Standard Corpus of Present-day Edited American English, for Use with Digital Computers*. Brown University, Department of Linguistics, 1979. URL <http://books.google.com/books?id=RWJPHQAACAAJ>.

- C. C. Fries. *The Structure of English: An Introduction to the Construction of English Sentences*. Harcourt, Brace and World, New York, 1952.
- Kuzman Ganchev, Ben Taskar, Fernando Pereira, and João Gama. Posterior vs parameter sparsity in latent variable models. In *Advances in Neural Information Processing Systems*, pages 664–672, 2009.
- Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 99:2001–2049, August 2010. ISSN 1532-4435. URL <http://portal.acm.org/citation.cfm?id=1859890.1859918>.
- Jianfeng Gao and Mark Johnson. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 344–352, Morristown, NJ, USA, 2008. Association for Computational Linguistics. URL <http://portal.acm.org/citation.cfm?id=1613715.1613761>.
- Dan Garrette and Jason Baldridge. Learning a part-of-speech tagger from two hours of annotation. pages 138–147, June 2013. URL <http://www.cs.utexas.edu/users/ai-lab/?garrette:naacl13>.
- Dan Garrette, Jason Mielens, and Jason Baldridge. Real-world semi-supervised learning of pos-taggers for low-resource languages. pages 583–592, August 2013. URL <http://www.cs.utexas.edu/users/ai-lab/?garrette:acl13>.
- R. Garside. The robust tagging of unrestricted text: the bnc experience. pages 167–180, 1996.

- R. Garside, G. Sampson, and G.N. Leech. *The Computational analysis of English: a corpus-based approach*. Longman, 1987. ISBN 9780582291492. URL <http://books.google.com/books?id=peZZAAAAMAAJ>.
- Sharon Goldwater and Tom Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proc. of the 45th Annual Meeting of the ACL (ACL-2007)*, pages 744–751, Prague, Czech Republic, June 2007. URL <http://www.aclweb.org/anthology/P/P07/P07-1094>.
- Sharon Goldwater, Tom Griffiths, and Mark Johnson. Interpolating between types and tokens by estimating power-law generators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 459–466. MIT Press, Cambridge, MA, 2006.
- Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. Producing power-law distributions and damping word frequencies with two-stage language models. *The Journal of Machine Learning Research*, 12:2335–2382, 2011.
- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, April 1993. ISSN 0956-375X. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=210672.
- B.B. Greene and G.M. Rubin. *Automatic Grammatical Tagging of English*. Department of Linguistics, Brown University, 1971. URL <http://books.google.com/books?id=VznTygAACAAJ>.
- Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology*

Conference of the North American Chapter of the Association of Computational Linguistics, pages 320–327, Morristown, NJ, USA, 2006. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1220835.1220876>. URL <http://dx.doi.org/10.3115/1220835.1220876>.

Z. S. Harris. From morpheme to utterance. *Language*, 22:161–83, 1946.

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

Kazi Saidul Hasan and Vincent Ng. Weakly supervised part-of-speech tagging for morphologically-rich, resource-scarce languages. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 363–371. Association for Computational Linguistics, 2009.

Martin Haspelmath. Word classes/parts of speech. In Paul Baltes and Neil J. Smelser, editors, *International Encyclopedia of the Social and Behavioral Sciences*, pages 16538–16545. Pergamon, Amsterdam, 2001. URL <http://email.eva.mpg.de/~haspelmt/2001wcl.pdf>.

R. S. Jackendoff. *X'-Syntax: A Study of Phrase Structure*. The MIT Press, Cambridge, MA, 1977.

A. Jasra, A. Doucet, D. Stephens, and C. Holmes. Interacting sequential Monte Carlo samplers for trans-dimensional simulation. *Computational Statistics & Data Analysis*, 52(4):1765–1791, January 2008. ISSN 01679473. doi: 10.1016/j.csda.2007.09.009. URL <http://dx.doi.org/10.1016/j.csda.2007.09.009>.

Mark Johnson. Why doesnt EM find good HMM POS-taggers? In *Proc. of*

the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-2007), pages 296–305, Prague, Czech Republic, 2007.

G Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal Of Computational And Graphical Statistics*, 5(1):1–25, 1996. URL <http://www.jstor.org/stable/1390750?origin=crossref>.

Sheldon Klein and Robert F. Simmons. A computational approach to grammatical coding of english words. *J. ACM*, 10(3):334–347, July 1963. ISSN 0004-5411. doi: 10.1145/321172.321180. URL <http://doi.acm.org/10.1145/321172.321180>.

Julian Kupiec. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6:225–242, 1992.

G. Lakoff. *Irregularity in Syntax*. Holt, Rinehart and Winston, New York, 1965.

Claude Lancelot and Antoine Arnauld. *The Port-Royal grammar: general and rational grammar*. Mouton, The Hague, Netherlands, 1975.

R. W. Langacker. *Foundations of Cognitive Grammar. Volume 1: Theoretical Prerequisites*. Stanford University Press, Stanford, CA, 1987.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. Simple type-level unsupervised pos tagging. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 853–861, Morristown, NJ, USA, 2010. Association for Computational Linguistics. URL <http://portal.acm.org/citation.cfm?id=1870658.1870741>.

- Geoffrey Leech. Corpus annotation schemes. *Literary and linguistic computing*, 8(4):275–281, 1993.
- Shen Li, João V. Graça, and Ben Taskar. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1389–1398, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=2390948.2391106>.
- P. Liang, M. I. Jordan, and D. Klein. Type-based MCMC. In *North American Association for Computational Linguistics (NAACL)*, 2010.
- Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proc. of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-2007)*, pages 688–697, Prague, Czech Republic, 2007.
- F. Lindsten and T.B. Schon. On the use of backward simulation in the particle gibbs sampler. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3845–3848, March 2012. doi: 10.1109/ICASSP.2012.6288756.
- Fredrik Lindsten, Michael I. Jordan, and Thomas B. Schön. Particle Gibbs with Ancestor Sampling, January 2014. URL <http://arxiv.org/abs/1401.0604>.
- Christopher D. Manning. Part-of-speech tagging from 97linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'11, pages 171–

189, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-19399-6.
URL <http://dl.acm.org/citation.cfm?id=1964799.1964816>.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993a. ISSN 0891-2017. URL <http://dl.acm.org/citation.cfm?id=972470.972475>.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993b. ISSN 0891-2017.

Ian Marshall. Choice of grammatical word-class without global syntactic analysis: Tagging words in the LOB corpus. *Computers and the Humanities*, 17: 139–150, 1983.

E. Mätzner. *Englische Grammatik*. Weidmannsche Buchhandlung, Berlin, 1880-85.

Bernard Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 20:155–171, 1993.

C. B. Mervis. Category structure and the development of categorization. In R. J. Spiro, B. C. Bruce, and W. F. Brewer, editors, *Theoretical Issues in Reading Comprehension: Perspectives from Cognitive Psychology, Linguistics, Artificial Intelligence and Education*, pages 279–307. Lawrence Erlbaum, Hillsdale, NJ, 1980.

Lindley Murray. *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*. Collins and Perkins, 1809.

- Radford Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.
- F. J. Newmeyer. *Linguistic Theory in America: The First Quarter-Century of Transformational Generative Grammar*. Academic Press, New York, 1980.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.
- Jim Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102:145–158, 1995.
- Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900, 1997.
- C. Pollard and I. A. Sag. *Information-Based Syntax and Semantics. Volume 1: Fundamentals*. CSLI Publications, Stanford, CA, 1987.
- C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, 1994.
- Gisa Rauh. Word classes as prototypical categories. In S. Scholz, M. Klages, E. Hanston, and U. Römer, editors, *Language: Context and Cognition: Papers in Honour of Wolf-Dietrich Balds 60th Birthday*, pages 259–70. Munchen : Langenscheidt-Longman, 2002.
- Gisa Rauh. *Syntactic categories: Their identification and description in linguistic theories*. Oxford University Press, 2010.

- Sujith Ravi and Kevin Knight. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, pages 504–512, 2009.
- R. H. Robins. The development of the word class system of the european grammatical tradition. *Foundations of Language*, Vol. 2(No. 1):pp. 3–19, Feb. 1966.
- E. Rosch. Human categorization. In N Warren, editor, *Studies in Cross-Cultural Psychology*, pages 1–49. Academic Press, London, 1977.
- I. A. Sag, T. Wasow, and E. M. Bender. *Syntactic Theory*. CSLI Publications, Stanford, CA, 2003.
- Hinrich Schütze. Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 251–258. Association for Computational Linguistics, 1993.
- Hinrich Schütze and Yoram Singer. Part-of-speech tagging using a variable memory markov model. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 181–187. Association for Computational Linguistics, 1994.
- L. Shen, G. Satta, and A. K. Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 760–767. Association for Computational Linguistics, 2007.

- Noah A. Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 354–362, Ann Arbor, Michigan, June 2005.
- Y. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992, 2006a.
- Yee Whye Teh. A hierarchical bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 985–992, Morristown, NJ, USA, 2006b. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1220175.1220299>. URL <http://dx.doi.org/10.3115/1220175.1220299>.
- Kristina Toutanova and Mark Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528. MIT Press, Cambridge, MA, 2008.
- Robert D. Van Valin, Jr. *An Introduction to Syntax*. Cambridge University Press, 2001.
- Ralph Weischedel, Richard Schwartz, Jeff Palmucci, Marie Meteer, and Lance Ramshaw. Coping with ambiguity and unknown words through probabilistic models. *Computational linguistics*, 19(2):361–382, 1993.
- G. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, Cambridge, MA, 1932.