Unsupervised Multilingual Learning

by

Benjamin Snyder

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author ....
Department of Electrical Engineering and Computer Science
September 3, 2010

Certified by ....
Regina Barzilay
Associate Professor
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Unsupervised Multilingual Learning

by

Benjamin Snyder

Submitted to the Department of Electrical Engineering and Computer Science
on September 3, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

For centuries, scholars have explored the deep links among human languages. In this thesis, we present a class of probabilistic models that exploit these links as a form of naturally occurring supervision. These models allow us to substantially improve performance for core text processing tasks, such as morphological segmentation, part-of-speech tagging, and syntactic parsing. Besides these traditional NLP tasks, we also present a multilingual model for lost language decipherment. We test this model on the ancient Ugaritic language. Our results show that we can automatically uncover much of the historical relationship between Ugaritic and Biblical Hebrew, a known related language.

Thesis Supervisor: Regina Barzilay
Title: Associate Professor

Acknowledgments

This thesis would not be possible without my collaborators: Tahira Naseem, Jacob Eisenstein, Kevin Knight, and most importantly, my advisor Regina Barzilay. Regina has been a brilliant and tireless advisor throughout my five years at MIT. She has set consistently high standards and has always provided the support and guidance needed to meet those standards. It is simply hard to imagine an advisor with more intellectual energy.

My thesis committee has been incredibly helpful and patient. Beyond their role in shaping the final form of this thesis, Michael Collins and Tommi Jaakkola have consistently made themselves available for research discussions. Much of the work of this thesis has its seed in these discussions. In addition, the influence of my lab-mates cannot be discounted. We have spent many long nights together, reading each others' paper drafts, and sometimes discussing far-flung topics like the state of world politics and the history of ideas.

I would also like to thank my friends and family for all their love and support. My friend Ali Mohammad deserves special mention for hosting me this past week. His sister, Asma Al-Rawi, also deserves my gratitude for feeding both her brother and me (the homemade grape leaves and tabouli deserve special mention). Finally, I dedicate this thesis to my entire family, but most especially to my parents and siblings who have always loved me, and to my newborn nephew Gavriel Yishay Frogel, whom I love without even having met.

Bibliographic Note

Some of the work presented in this thesis has appeared in previous publications. The work on part-of-speech tagging was first published in Snyder et al. [111] and then expanded upon in Snyder et al. [112] and Naseem et al. [85]. The grammar induction work was originally published in Snyder et al. [110]. Parts of the decipherment work were originally published in Snyder et al. [109]. Finally, the work on morphological segmentation (mentioned briefly in the concluding chapter) was originally published in Snyder and Barzilay [108] and Snyder and Barzilay [107]. The code and data from the experiments discussed in this thesis are available at `http://groups.csail.mit.edu/rbg/code/unsupervised_multilingual`.

Contents

Chapter 1

Introduction

As I write this sentence, millions of human beings are busy communicating with one another through the written word. In fact, reading and writing now constitute a greater part of human communication than ever before. As populations become more literate and world-wide access to technology increases, communication through electronic text has also become more linguistically diverse. Many dozens of languages are used everyday on the web, in emails, and in text messages.

In this age of written communication, the development of human language technology takes on greater importance than ever before. One of the chief goals of this enterprise is to develop models that can automatically analyze large bodies of text and quickly perform the kinds of tasks that would normally require intense human effort. Some examples of these tasks include the automatic translation between languages and the automatic extraction of information from text. The primary difficulty in achieving human performance on these tasks is that natural language is *ambiguous*.

This thesis aims to tackle the problem of natural language ambiguity within a novel framework: *multilingual learning*. Throughout this thesis, we will argue that by carefully modeling cross-lingual connections, we can push the state-of-the-art in language technology to new limits. The key idea is that patterns of ambiguity differ across languages. By jointly modeling the latent structure of multiple languages, the idiosyncratic ambiguities of each language can be more effectively resolved. This

Figure 1-1: Example of an Ugaritic text found at Ras Shamra. We thank Dr. N. Wyatt and Dr. J. B.. Lloyd of the Ras Shamra project at University of Edinburgh for the use of this image.

thesis presents several novel findings:

Multilingual modeling improves accuracy for classical text analysis tasks. These tasks are of fundamental importance and have been studied extensively within the statistical NLP community. Some tasks, such as grammar induction, involve the prediction of complex latent structure. Even so, we show that cross-lingual regularities can be captured while still allowing realistic language variation. We also show that multilingual accuracy *continues* to grow as more languages are added. These results point to a future multilingual NLP paradigm.

Multilingual modeling enables new language analysis tasks. In particular, we present the first model to successfully decipher a *lost language*. It took scholars four years to initially crack the ancient Ugaritic language. Decades of painstaking scholarship has continued to flesh out its relationship to other Semitic languages. We present a statistical model which automatically uncovers much of the historical relationship between Ugaritic and Biblical Hebrew. We design our model to capture the many intuitions that have guided human scholars. By modeling these intuitions, we can automatically decipher a substantial portion of the Ugaritic vocabulary.

1.1 Chapter Overview

The next two sections outline some of the practical and scientific motivations driving the work of this thesis. Section 1.2 starts with the practical side: In it we discuss the recent rise of the (electronic) written word as a means of communication and its increasingly multilingual nature. We argue that in order to develop intelligent text-processing tools for the world's languages, new techniques are needed. Section 1.3 turn to some scientific motivations. We discuss the need to preserve the hundreds of languages in danger of immediate extinction and the thousands of languages under threat of extinction over the coming decades. We also look to the past and discuss the need for technology to help us better understand languages from our ancient history. Section 1.4 introduces the reader to the main ideas and contributions of this thesis. First we describe the diversity of linguistic structure and the nature of human language ambiguity. We then introduce multilingual learning as a conceptual framework. Finally, we show how we applied this framework to several tasks of automatic linguistic analysis. Section 1.5 then outlines some previous research on multilingual language technology and discusses the relationship of our thesis to that work. Finally, section 1.6 provides the reader with an overview and roadmap for the remainder of the thesis.

1.2 Practical Motivation

World literacy rates have skyrocketed from 66% to 84% over the past six decades. Even more recently, we have witnessed a revolution in our ability to communicate with one another through the written word. An astounding volume of human communication now takes place through text: We send an average of 4.1 billion text messages and 47 billion non-spam emails each day [27, 115]. Indeed, text plays a larger role in language communication than ever before in human history; this trend is likely to continue.

Until recently, the technology infrastructure fueling this rapid growth was confined to a handful of countries. As a result, the World Wide Web was initially dominated by English speaking users. Such users constituted a majority of all web users until the turn of the century. Starting in the year 2000, however, the percentage of web users from non-English speaking countries began to rapidly increase. It is estimated that, as of today, native English speakers only constitute a third of all internet users.

As the number of non-English technology consumers has increased, so has the number of non-English electronic texts. In the year 2000, over 70% of webpages were written in English [125]. Seven years later, this number had dropped to 45%.[1] In addition, the web has become an increasingly dynamic and interactive environment. Because of this, much of the world population of internet users now *produces* electronic text in their native languages. For example, the percentage of non-English Wikipedia articles has risen from 10% in 2002 to more than 75% in 2007. Indeed, by 2007 about a third of all Wikipedia articles were not even written in the top 10 languages of the web.[2]

Research within the language technology community has not kept pace with this explosion of language diversity. At the 2008 meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2008), 119 long

[1] http://dtil.unilat.org/LI/2007/ro/resultados_ro.htm

[2] http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

papers were presented. Fewer than one-fifth of these papers examined multiple languages [5].[3] Unsurprisingly, English still dominates the field as an object of study: 63% of single-language papers focused exclusively on English [5].

There are a variety of reasons for the continued dominance of English as an object of study and technology. Some of these are sociological in nature. A disproportionate number of NLP researchers are themselves native English speakers, and it is quite simply easier to develop technology for a familiar language. In addition, there is the phenomenon of data-set inertia. The standardization of data-sets can result in incentives to evaluate one's system on a very narrow range of languages (and genres).

Perhaps more important, though, is the paucity of rich linguistic resources for most languages of the world. As we discuss in the next section, human languages are rife with ambiguity. One way of dealing with this phenomenon is to first have humans annotate texts to resolve the relevant ambiguities, and to then *train* computer models on these annotated texts. Unfortunately, the time and expense involved in creating such resources can be prohibitive. One groundbreaking resource, The Penn English Treebank, took a team of professional computer scientists and linguists years to create [77]. It is therefore unlikely that such richly annotated text corpora will be created for a large portion of the world's languages anytime soon.

Recently, a number of *unsupervised* approaches have been developed in the natural language processing community. These are methods which are trained solely on raw text, without the input of a human annotator. While this is a promising avenue of research, the performance of purely unsupervised systems largely remains too low for practical use. Thus, it is important to look to other ideas and sources of information.

The alternative that we propose in this thesis is *multilingual learning.* The main idea is that we can leverage varying cross-lingual patterns of ambiguity as a form of naturally occurring supervision. As we show throughout the thesis, utilizing this source of information can lead to significant gains in accuracy without the

[3] Or multiple language-pairs in the case of machine translation papers.

involvement of human annotation.

1.3 Scientific Motivation

In the previous section we discussed the practical motivation for multilingual modeling. Namely, that it can enable the rapid development of text analysis tools for the growing number of languages used in electronic communication. In this section we turn to less practical but equally important motivations for our work.

One of these is the threat of language extinction. Of the world's 6,900 spoken languages, hundreds are at risk of immediate extinction and thousands more are likely to disappear over the coming decades. Hale et al. [50] predict that 90% of the world's linguistic diversity will be lost by the year 2100. Without immediate and sustained efforts to document the world's languages, our ability to understand the nature of language may be irreparably harmed.

While language documentation is considered a high priority in the field of linguistics, the computer science community has yet to make substantial contributions to this effort [1]. Immediately developing large annotated corpora for every at-risk language is simply not feasible. Abney and Bird [1] instead propose that we start with a single reference text and then rapidly collect translations of this text into every endangered language possible. Linguistic annotations of this massively parallel text can then be slowly built up through a combination of automatic methods and human supervision.

The methods we have developed dovetail nicely with this goal. Throughout this thesis we endeavor to show that multilingual parallel texts can serve as (an imperfect) replacement for human annotation. We also show that performance *continues* to improve as more languages are added to the mix. Thus, multilingual learning should prove invaluable as a first step in the automatic glossing of such a universal language corpus.

A complete understanding of human languages must also include knowledge of their history and evolution. The scholarly enterprise of *lost language decipherment* endeavors to fill in some of the historical gaps. Many ancient texts have indeed been deciphered over the past two centuries [100], some after decades of scholarly effort. It is hard to overestimate the importance of these discoveries. The history of writing and the earliest forms of human literature have been revealed.

Nonetheless, several crucial languages and scripts have yet to be understood. We believe that computational and statistical methods will be invaluable to such future decipherment efforts. This motivates our development of the first statistical model for lost language decipherment.

1.4 This Thesis

We now introduce the key contributions of this thesis. We first present the framework of multilingual learning at a conceptual level and then show how we realized this framework at the practical level. Because so much of this thesis is based on the idea of systematic differences across languages, we begin with a brief exposition of that topic.

1.4.1 Language Diversity

Anyone who has attempted to learn a foreign language knows that it requires much more than memorizing a bilingual dictionary. Rote learning of a new vocabulary is certainly difficult. Even more challenging, though, is learning to express oneself and communicate in a new tongue. Part of the reason for this difficulty is that languages differ from one another in a variety of ways. In particular, languages vary in the way meaning is mapped to linguistic structure.

Perhaps the most obvious manifestation of this diversity is the systematic difference in word order across languages. Consider the following pair of English and Japanese sentences:

English: IBM bought Lotus.
Japanese: *IBM Lotus bought.*

English: Sources said that IBM bought Lotus yesterday.
Japanese: *Sources yesterday IBM Lotus bought that said.*

As Collins [25] points out, the correspondence between the Japanese and English versions of these sentences can be succinctly captured by a single rule. In English, the standard word order is subject-verb-object. In Japanese, however, the position of the verb and object are reversed: subject-object-verb.

Another striking example of language difference comes from morphology, which studies how words are formed from smaller units of meaning (called *morphemes*). Consider the pair of English and Hebrew sentences:

English: I took a walk.
Hebrew: ṭayalti

The first thing we might notice is the difference in sentence lengths. The meaning which we express in English using four words, is expressed using only *one* word in Hebrew. However, if we examine this Hebrew word more closely, we will see that it is composed of two smaller morphemes, each bearing a distinct meaning:

(1) ṭayal-/*took a walk* (2)-ti/*I*

In general, Hebrew and many other languages can pack a lot of complex meaning into single words by forming them from multiple morphemes. The exact manner in which this is done is language-specific, but given any set of languages, consistent cross-lingual patterns will become apparent. In fact, morphology interacts with word-order in an interesting way. For languages such as English, word order is the primary indicator of the grammatical role of words. In our first example, we knew that IBM was the company that purchased Lotus (rather than the other way around) because of the order of the words. In many other languages, though, the grammatical role of individual words is determined through the addition of a *case marker*. [4] As a result, the word order of these languages tends to be much more free and can vary to provide different emphases.

Finally, we make one final observation about our English-Hebrew example. The English verb phrase *took a walk* corresponds to the intransitive Hebrew verb *ṭayal*. Most of the semantic weight in the English verb phrase is carried by the noun *walk* (*took* in this context is an instance of a “light verb”). In contrast, Hebrew packs the same meaning into a single, semantically weighty verb. Some languages (e.g. Urdu) use light verb constructions much more regularly than others. Thus, we see that languages can also differ in how they inject meaning into different parts-of-speech.

1.4.2 Multilingual Learning: The Main Idea

On the face of it, this vast linguistic diversity would make the development of multilingual language processing tools very difficult. Techniques developed for one language, or one set of languages, may not account for the kinds of linguistic structures encountered in other languages. In this thesis, however, we argue on the contrary, that it is actually possible to harness linguistic diversity and use it to our advantage. To do so, we develop the framework of *multilingual learning*. Our underlying hypothesis is that cross-lingual variations in linguistic forms correspond to systematic variations in *ambiguity*. As a result, the ambiguities encountered in each language differ to some degree; by jointly modeling multiple languages, the overall ambiguity can be drastically reduced. Before we flesh this idea out in a systematic way, we must briefly discuss the notion of ambiguity itself.

[4] A case marker is a suffix that specifies the grammatical role of the word.

Language Ambiguity

According to its most basic definition, ambiguity occurs when an observed signal can be interpreted in more than one way. This definition takes on more meaning when we contrast natural human languages with computer programming languages. Programming languages are explicitly designed to *avoid* ambiguity. That is, every syntactically valid program must compile into a single Abstract Syntax Tree. In contrast, sentences in natural languages are fraught with ambiguity. Consider the famous example:

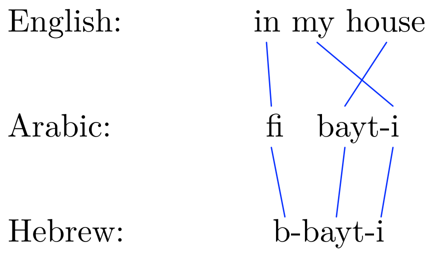
I saw the man on the hill with the telescope.

This sentence can be interpreted in many different way. The most salient ambiguity is the location of the telescope. Is it the man who has the telescope? Was the viewing of the man performed with the telescope? Or is it the hill that has a telescope on it? This example demonstrates that there is a lot more to language than the explicit signal that we observe, whether it be auditory or a textual. Language is rife with *latent structure*. In this example, the different interpretations regarding the placement and use of the telescope each correspond to a different latent *parse tree* of the sentence. Now, in the context of a larger communicative narrative, the intended interpretation of this sentence would probably be completely obvious. In fact, in normal human communication we seem to nearly always resolve ambiguity with ease. To do so, we use our vast store of world knowledge, a deep unconscious understanding of language structures, as well as contextual cues. To a large degree then, ambiguity is in the eyes of the beholder.

Computers, however, are particularly bad at resolving natural language ambiguity. To some degree this can't be helped: We know very little about the way humans process language and represent facts about the world. So, in a certain sense, it is not even clear what it would *mean* for a computer to correctly resolve all the ambiguities of a sentence. However, there are certain ambiguities that we might reasonably expect a computer to settle. For example, we might expect the computer to provide the kind of formal analysis that a linguist would give for the sentence: a parse tree, a sequence of parts-of-speech, and a morphological analysis of each word. Thus, in our context, ambiguity refers to the very complex relationship between the observed signs of the sentence and the latent formal linguistic structure.

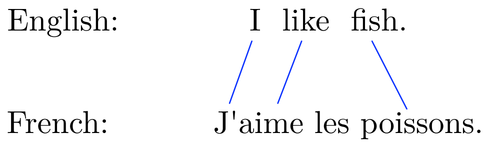
Two Motivating Examples

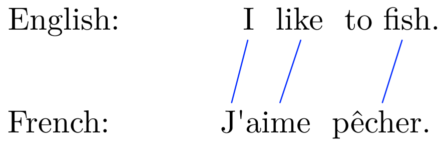
We can now return to our hypothesis: Cross-lingual variations in linguistic structure correspond to variations in ambiguity. We start by illustrating this idea with two examples. First consider the following phrase in English, Arabic, and Hebrew:



For this example, the languages are given in increasing order of morphological complexity. English, a morphologically simple language, employs three distinct words, each consisting of only one morpheme. Hebrew and Arabic, on the other hand, both express the possessive pronoun *my* as a suffix *-i* on the possessed noun. In this example at least, Hebrew displays a bit more morphological complexity than Arabic, expressing the preposition *in* with the prefix *b-* rather than with the separate word *fi*. Now suppose our goal were to uncover the latent morphology of each language. In this case, the separate Arabic preposition would provide a clue that Hebrew is employing a prefix. Furthermore, the three word English phrase would provide powerful evidence regarding the prefixes and suffixes of the two other languages.

Now consider two pairs of sentences in English and French:





The first thing we might notice here is that the English word *fish* displays part-of-speech ambiguity. It can function either as a noun (as in the first sentence), or as a verb (as in the second sentence). In fact, this kind of noun/verb ambiguity is extremely common in English. In contrast, French deploys two very distinct words to express these two meanings: *poissons* for the noun, and *pêcher* for the verb. Thus, we can see here that the part-of-speech ambiguity in the English sentences simply doesn't exist in the French counterparts. Thus, if our goal were to predict the latent part-of-speech categories for English, having French translations could be enormously beneficial.

These examples are instances of a more general phenomenon: what one language leaves implicit, and thus ambiguous for computers (or perhaps even humans), another will express directly through overt linguistic forms. Thus, when jointly modeling multiple languages, we can treat these variations in ambiguity as a form of *naturally occurring supervision* in order to more accurately predict latent structure.

Conceptual Framework

One might conclude from these examples that for any pair of languages, one would consistently provide more explicit information in some linguistic category than the other. For example, languages with more complex morphology may systematically provide more explicit syntactic cues (in the form of case markings) than languages

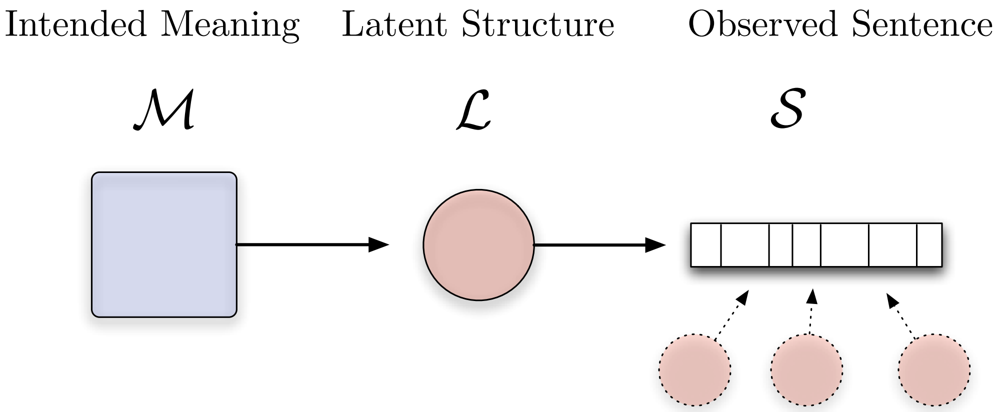


Figure 1-2: **Conceptual overview of ambiguity.** The intended meaning $\mathcal{M}$ first produces a latent linguistic structure $\mathcal{L}$ which in turn produce the observed sentence $\mathcal{S}$. Ambiguity arises since spurious latent structures could have produced the same sentence.

which rely solely on word order. Languages in the latter category, in turn, may systematically yield rich information regarding the morphology of their morphologically complex fellow languages.

However, we can make the argument for multilingual learning more general and symmetric if we approach things from a slightly different perspective. To start, we can view the phenomenon of ambiguity as a result of the language-production process sketched in figure 1-2. First some intended meaning $\mathcal{M}$ arises in the mind of the speaker or writer. That meaning then gets mapped to some set of abstract linguistic structures $\mathcal{L}$, such as parse trees, parts-of-speech, and morphemes. Finally, the linguistic structures produce a physical signal representing the observed sentence $\mathcal{S}$. However, the mapping between abstract linguistic structures and sentences is not one-to-one. As a result, any given sentence may have been generated by any number of *spurious* latent structures.

Figure 1-3 extends this scheme to the production of bilingual parallel sentences. We assume (somewhat unrealistically, see below) that the same meaning and latent linguistic structure underly the sentences in each language. The languages diverge only in the final stage, when each one maps structure to signal in a unique, idiosyncratic way. How does this affect ambiguity? In figure 1-3, we show two entirely

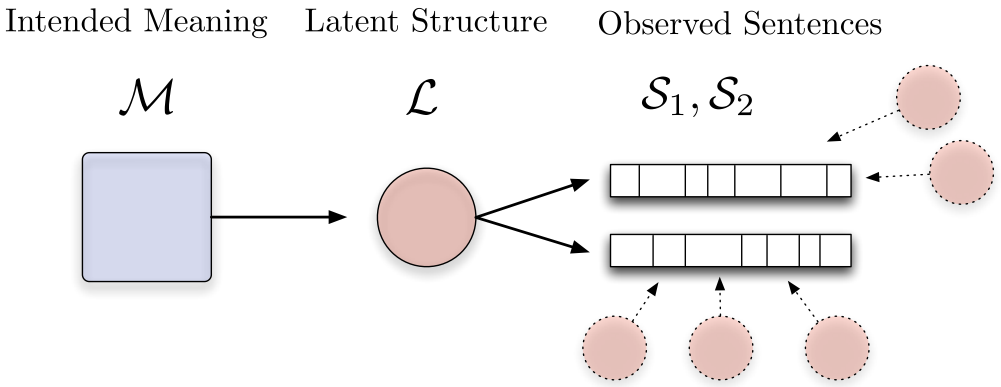


Figure 1-3: **Conceptual overview of multilingual learning 1.** Observing a parallel bilingual sentence pair $\mathcal{S}_1, \mathcal{S}_2$ reduces ambiguity: The sets of spurious latent structures for the two sentences do not overlap; only the true latent structure $\mathcal{L}$ produces both sentences.

disjoint sets of spurious latent structures for the two sentences. According to this picture, then, ambiguity simply ceases to exist in the bilingual scenario: Only the *true* latent structure could have simultaneously produced both sentences. This is obviously unrealistic, but it illustrates the idea well. More sensibly, we could expect some *subset* of the spurious latent structures to apply to both sentences, leading to a *reduction* in ambiguity. Either way, the assumption we make is the following. At least *some* of the spurious structures arise from language-specific features of the structure-to-signal mapping. Consequently, many of these ambiguities will be idiosyncratic to some language.

According to this argument, even languages with very similar *coarse* linguistic properties should provide each other with mutual benefit. For example, consider two languages that are equally morphologically rich — i.e. assume that the average number of morphemes per word is identical. Even so, the languages will surely differ somewhat in their inventory of morphemes and in their exact patterns of morpheme combination. Thus, for a given meaning, the distinct patterns and rules for each language are likely to yield a sentence with correspondingly distinct morphological ambiguities.

In figure 1-4 we remove one of the simplifying assumptions of the previous figure.

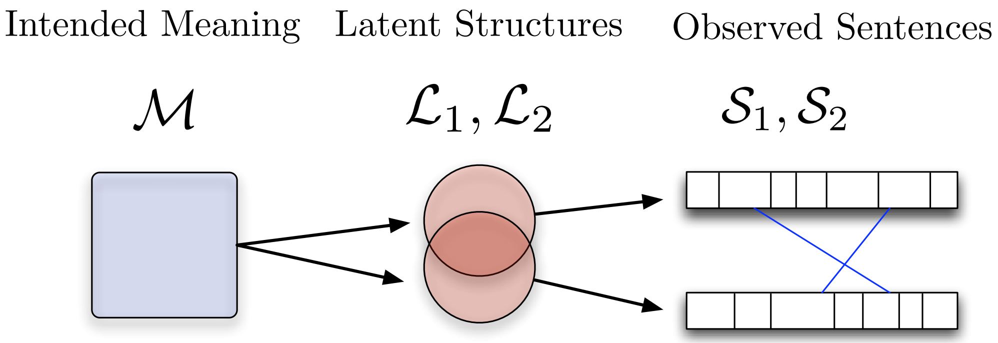


Figure 1-4: **Conceptual overview of multilingual learning 2.** Even when expressing the same meaning, languages often differ in latent structure. Systematic word-level correspondences in the sentences serve as a guide for finding shared structure.

Previously we had assumed that for parallel sentences, the languages would share a single latent structure. In reality, even for parallel sentences, the latent parse trees, morphemes, and parts-of-speech can differ in significant ways. Figure 1-4 reflects this reality by positing two overlapping latent structures, $\mathcal{L}_1$ and $\mathcal{L}_2$. Each such structure is produced by a language-specific mapping from the shared meaning.

This realization brings out one of the key technical challenges of multilingual learning. We need to identify underlying shared structure (i.e. the intersection $\mathcal{L}_1 \cap \mathcal{L}_2$), while still allowing robust language-specific idiosyncrasies (i.e. the symmetric difference of $\mathcal{L}_1$ and $\mathcal{L}_2$). Fortunately, figure 1-4 also displays a source of information that will help us in this task – cross-lingual word alignments.

To clarify, we will say that word w in sentence $\mathcal{S}_1$ is *aligned* to word w' in sentence $\mathcal{S}_2$ when we observe a general pattern of w and w' appearing in parallel sentences. If this is the case, it is likely that w and w' share the same meaning or syntactic function across the two languages. In other words, it is likely that they are *translations* of one another. Throughout this thesis, we will assume that such alignments are observed.[5] We will further assume that these alignments *reflect* the underlying shared structure of the two sentences. In this way, they will repeatedly

[5] In practice, we use the output of the GIZA++ alignment tool [90], which assumes no prior knowledge of either language.

guide our learning algorithms in identifying both shared and idiosyncratic language structure.

1.4.3 Multilingual Learning: In Practice

Now we discuss our methods for realizing the multilingual framework discussed above. In designing probabilistic multilingual models, we employ the hierarchical Bayesian modeling framework (see Gelman [39] and Robert [99] for reference texts). In this framework, we model the observed word-aligned sentences as the final outcome of a cascade of unobserved random variables. By specifying the dependency structure and conditional probabilities of this hierarchy of variables, we provide an inductive bias for our model. For example, if our goal is to discover the latent parts-of-speech of each sentence, then we structure our latent variables as a sequence, mirroring the words themselves. If, on the other hand, our goal is to induce latent parse trees, then we structure our latent variables into trees. In all cases, we predict the latent variable values which have highest posterior probability, given the observed sentences and alignments.

To put it more succinctly: The definition of a model specifies the *structure* of the latent patterns we wish to find. The inference algorithm then searches for those latent patterns which best mirror the *observed* patterns of words and sentences.

This Bayesian framework allows us to neatly capture the main intuition of multilingual learning. Namely, that each sentence pair is the result of a probabilistic process involving both shared and language-specific latent variables. Even so, the *scope* of the shared explanatory mechanism is often unknown: some sets of languages exhibit a much larger degree of shared structure than others. For example, related languages like Hebrew and Arabic will tend to mirror each other in morphological structure much more than unrelated language pairs (such as English and Hebrew). To account for this variability, we employ non-parametric statistical methods which allow for a flexible number of shared variables, as dictated by the languages and data at hand.

In the remainder of the section, we will briefly describe how we applied multilingual learning to three different tasks: part-of-speech tagging, grammar induction, and lost language decipherment. In each case, we designed our models and experiments to touch on some fundamental questions about the viability of the multilingual framework:

Question 1: Will multilingual learning provide more or less benefit when the languages in question are from the same family (e.g. Hebrew and Arabic, Italian and French, German and Dutch)? One might argue either way. One the one hand, related languages are likely to have a greater degree of shared latent structure. On the other hand, if their patterns of ambiguity are almost identical then little benefit would be gained.

Question 2: Can multilingual learning be made to scale-up beyond pairs of languages? It seems that the *a priori* arguments in favor of multilingual learning would only be strengthened as additional languages are modeled. Each language may provide some unique disambiguation cues lacking in the others. As a practical matter, massively multilingual data-sets do exist (e.g. the Bible, which has been translated into over 1,000 languages) and an ideal multilingual learning technique would thus scale gracefully in the number of languages.

Question 3: Can multilingual learning account for complex latent structure where cross-lingual shared elements are minimal and difficult to discern? To do so effectively and efficiently will require an unobtrusive *representation* of whatever shared structure exists.

Question 4: Can multilingual learning be effective without parallel data? Throughout this section our arguments have depended on the existence of parallel sentences as a computational Rosetta stone. However, if the languages in question come from the same family, it may be possible to use language-wide structural correspondences rather than the correspondences delivered by parallel text.

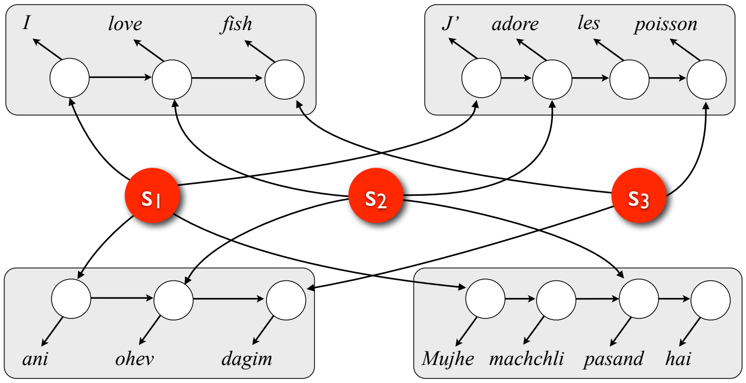


Figure 1-5: Part-of-speech graphical model structure for example sentence. In this instance, we have three *superlingual tags*: one for the cluster of words corresponding to English “I”, one for the cluster of words corresponding to English “love”, and one for the cluster of words corresponding to English “fish.”

Answering all of these questions *conclusively* is beyond the scope of this thesis. Nevertheless, as we discuss in the concluding chapter, our experiments yield some initial answers.

Part-of-Speech Tagging

Perhaps the simplest of the three tasks is unsupervised part-of-speech tagging. As input for the task, we are given (i) a multilingual parallel text corpus and (ii) a seed dictionary which lists parts-of-speech for some subset of words in each language. For example, the word “can” in English would be listed with three part-of-speech tags: an auxiliary verb, a noun, and a regular verb. The goal is to automatically select the contextually appropriate part-of-speech for each word in the corpus. Although we utilize the multilingual parallel corpus for training, we test our performance separately for each language on a monolingual test corpus.

For this task, the latent structure we wish to induce for each sentence is very simple: fixed-length sequences of part-of-speech tags, one for each language. Because of this simplicity, we view this as an ideal task for multilingual experimentation. We designed two models for this task. The first is inherently bilingual and helps address the first of our questions, namely whether pairings within a language family will be more or less beneficial than pairings of unrelated languages.

The second model was designed from the beginning to scale gracefully in the number of languages. As such it can provide some answers to the second of our questions: whether multilingual learning can keep providing additional benefit as languages are added to the mix. Here we give a brief overview of the structure of this second model.

We posit a separate Hidden Markov Model (HMM) for each language, in which the hidden states correspond to part-of-speech tags. In order to model shared cross-lingual structure, we posit an additional layer of latent variables, referred to as *superlingual tags.* We place such a tag over each cluster of aligned words in the sentence.[6] Intuitively, the superlingual tag propagates information across languages by encouraging cross-lingual regularities.

In a standard HMM, we can write the joint probability of a sequence of words $\mathbf{w}$ and part-of-speech tags $\mathbf{y}$ as product of *transition* and *emission* probabilities:

$$P(\mathbf{w}, \mathbf{y}) = \prod_i P(y_i | y_{i-1}) P(w_i | y_i)$$

Under our latent variable model, the probability of bilingual parallel sentences $(\mathbf{w}^1, \mathbf{w}^2)$, bilingual part-of-speech sequences $(\mathbf{y}^1, \mathbf{y}^2)$, and superlingual tags $\mathbf{s}$ is given by:

$$\begin{aligned}
&\prod_i P(s_i) \\
&\prod_j P\left(y^1_j | y^1_{j-1}, s_{f(j,1)}\right) P(w^1_j | y^1_j) \\
&\prod_k P\left(y^2_k | y^2_{k-1}, s_{f(k,2)}\right) P(w^2_k | y^2_k),
\end{aligned}$$

where $f(m, n)$ gives the index of the superlingual tag associated with word m in language n. Notice that the part-of-speech tagging decisions of each language are independent when conditioned on the superlingual tags $\mathbf{s}$. It is this conditional

[6] The word alignments are produced by an standard word alignment tool and are considered fixed.

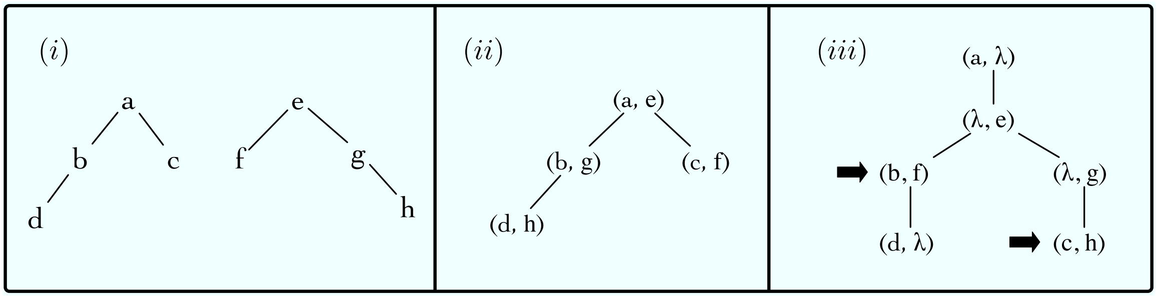


Figure 1-6: A pair of trees (i) and two possible alignment trees. In (ii), no empty spaces are inserted, but the order of one of the original tree's siblings has been reversed. In (iii), only two pairs of nodes have been aligned (indicated by arrows) and many empty spaces inserted.

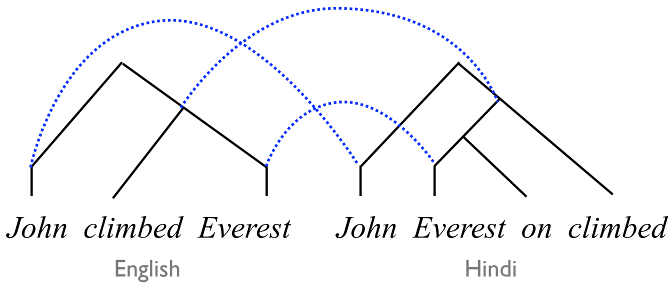
independence which gives our model some of its crucial properties. Superlingual variables promote cross-lingual regularities. Yet word order, part-of-speech selection, and even part-of-speech inventory are permitted to vary arbitrarily across languages. In addition, this architecture allows our model to scale linearly in the number of languages: when a language is added to the mix we simply add new directed edges from the existing set of superlingual tags for each sentence.

Findings: Accuracy for each of the eight languages we studied improves substantially over a state-of-the-art monolingual baseline. In one scenario, the gap between unsupervised and supervised performance was cut by two-thirds without any human annotation. These are the first results to show that performance *continues* to improve as languages are added to the mix.

Grammar Induction

A more complex task is that of unsupervised grammar induction. The goal is now to induce the underlying grammatical structure of each sentence in the form of a tree bracketing. In the monolingual setting, learning accurate parsing models without human-annotated texts has proven quite difficult [20, 64]. Here we consider the unsupervised bilingual scenario, where parsing models are induced simultaneously for pairs of languages using parallel texts. As before, we train our model on the bilingual corpus, but test our performance on separate monolingual data.

In the previous task of part-of-speech tagging, the *structure* of the latent variables was essentially observed, as they were determined by the sentences and their word alignments. In contrast, grammar induction is a task of structure *prediction.* In the monolingual scenario, the latent structure is a single tree. However, even for very literal translations, parse trees across languages can diverge significantly. Consider the following pair of parsed sentences in English and Hindi:



Even in this simplest of sentence pairs, we notice syntactic divergence. While the English sentence uses the simple transitive verb "climbed" to express the fact that John completed his climb of Everest, the verb in the Hindi sentence takes the post-positional argument "Everest on." The syntactic divergence in real-life examples can be much more severe.

This task addresses the third in our list of questions. Can we induce complex latent structure for each language with minimal shared elements? The key challenge here is *representational.* We need to parse both sentences with possibly quite divergent trees, while recognizing shared syntactic elements. In effect, we seek to produce two loosely bound trees.

We achieve this loose binding of trees by adapting the formalism of *unordered tree alignment* [60] to a probabilistic setting. Under this formalism, any two trees can be aligned using an *alignment tree.* The alignment tree embeds the original two trees within it: each node is labeled by a pair (x, y), (λ, y), or (x, λ) where x is a node from the first tree, y is a node from the second tree, and λ is an empty space. The individual structure of each tree must be essentially preserved under the embedding.

The flexibility of this formalism can be demonstrated by two extreme cases: (1) an alignment between two trees may actually align *none* of their individual nodes, instead inserting an empty space λ for each of the original two trees' nodes. (2) if the original trees are isomorphic to one another, the alignment may match their nodes exactly, without inserting any empty spaces. See Figure 1-6 for an example. An additional benefit of this formalism is computational: The marginalized probability over all possible alignments for any two trees can be efficiently computed with a dynamic program in polynomial time in the size of the two trees.

We embed this formalism in a Bayesian probabilistic model. The key objective underlying our model is the following: We want to predict tree pairs (T_1, T_2) with tree alignments $\mathcal{A}$ such that:

1. Tree T_1 best explains the grammatical regularities of language 1.

2. Tree T_2 best explains the grammatical regularities of language 2.

3. The tree alignment $\mathcal{A}$ best explains the bilingual word alignments.

4. Aligned constituents best explain cross-lingual grammatical regularities.

Findings: For each of three different language pairs, our bilingual model outperforms a state-of-the-art baseline, sometimes by quite substantial margins. These are the first results to show that the complex correspondences between bilingual parse trees can be effectively captured in a probabilistic model.

Lost Language Decipherment

The two tasks just discussed all assumed the existence of multilingual parallel text. For traditional text processing tasks this is a reasonable assumption, as parallel texts are readily available for many of the world's languages. In contrast, we now turn to the task of *lost language decipherment*.

When a lost script or language is discovered we rarely have the luxury of parallel data. Typically, our only hope of recovering the language comes from a cross-lingual structural analysis that links the lost writing system to a known language. Such analysis can take humans decades to perform. Dozens of lost languages and scripts have been manually deciphered by scholars over the last two centuries. Perhaps surprisingly, computers have never played a role in the successful decipherment of any language.

This task then, addresses the final of our four questions above. We have no parallel corpus directly linking sentences in the lost language to a known language. Nevertheless, we hope to discover language-wide similarities connecting the lost language to a living relative.

Our definition of the computational decipherment task closely follows the setup typically faced by human decipherers [100]. Our input consists of texts in a lost language and a corpus of non-parallel data in a known related language. The decipherment itself involves two related sub-tasks: (i) finding the mapping between alphabets of the known and lost languages, and (ii) translating words in the lost language into corresponding cognates of the known language.

We formulate a Bayesian probabilistic model which captures many of the intuitions that have guided human decipherers. First among these is that both character and lexical correspondences across related languages should be consistent. In addition, morphological analysis plays a key role in our model, as correspondences between highly frequent prefixes and suffixes can be particularly revealing (and easy to find). Finally, we develop a novel prior that encodes a crucial intuition: that the mapping between alphabets should be *structurally sparse*. Each character in the lost language should map to a very limited number of characters in the related language, and vice versa. We applied our decipherment model to a corpus of Ugaritic, an ancient Semitic language discovered in 1928 and manually deciphered four years later, using knowledge of Hebrew, a related language. As input to our model, we use the corpus of Ugaritic texts along with a Hebrew lexicon extracted from the Hebrew Bible.

Findings: Our model yields an almost perfect decipherment of the Ugaritic alphabetic symbols. In addition, our model successfully deciphers 63% of all Ugaritic words with Hebrew cognates. These are the first results showing the automatic decipherment of a lost language.

1.5 Previous Approaches

In this section we outline several past approaches to multilingual NLP to better highlight the novelty of our work.

Interest in developing language technology in a multilingual setting goes back to the early days of statistical NLP. In its most basic form, this can simply refer to studies which considered the performance of a model on a large range of languages (without any explicit cross-lingual modeling). To cite just two recent examples, Ganchev et al. [38] studied whether better bilingual word alignments in text lead to more accurate translation models for six different language pairs. Nivre and McDonald [88] present a dependency parsing model which they test on a suite of 13 languages from many different families. More generally, a community-wide interest in multilingual experiments has certainly been growing. This interest is reflected in the fact that several of the past few shared tasks at the Conference on Computational Natural Language Learning (CoNLL) have utilized data-sets spanning multiples languages [15, 87, 49]. Establishing a norm of multilingual experimentation helps avoid communal "overfitting" of models to the English language [5].

Certain tasks are also inherently multilingual. Thus, the tasks of machine translation and bilingual dictionary construction, by their nature, occur in multilingual settings. Some researchers have shown that by considering more than two languages at a time, even bilingual dictionaries can be more accurately induced automatically [41, 76]. Likewise, by considering multiple source languages, automatic translations can be improved [89, 118, 24, 21, 8]. In contrast, this thesis focuses on the accurate induction of *monolingual* language structure, albeit by jointly modeling multiple languages.

Another influential line of prior work starts with the observation that rich linguistic resources exist for some languages but not others. The idea then is to *project* linguistic information from one language onto others. Yarowsky and his collaborators first pioneered this idea and applied it to the problems of part-of-speech tagging, noun-phrase bracketing, and morphology induction [128, 127, 126]. In all three cases, the existence of a bilingual parallel text along with highly accurate predictions for one of the languages was assumed. Projection methods have now been applied to a wide variety of NLP tasks, from parsing [58, 124] to semantic role labeling [91]. In addition, some recent work even eschews the use of parallel texts. Instead of projecting information at the *annotation*-level, projection occurs at the *parameter*-level. The learned parameters of a supervised system in one language are directly applied to related languages. This idea has been applied to the tasks of morphology induction and part-of-speech tagging for Slavic languages [53, 35]. In stark contrast to the line of research sketched out above, this thesis does not assume that accurate supervised systems or annotations exist for *any* of the languages in question. Instead, it is the cross-lingual patterns *themselves* which are regarded as a rich source of information.

Perhaps closest to the spirit of this thesis is a line of work begun even earlier. Dagan et al. [29] propose the use of bilingual parallel texts for automatic word sense disambiguation. The main idea was that patterns of word-meaning ambiguity vary in systematic ways across languages. For example, the Hebrew verb *laḥtom* has various meanings, including (a) *to sign* and (b) to *to finish.* One way to automatically distinguish between these two senses would be to consult the parallel English sentence. If the English word *sign* is used, then we assign meaning (a), and if the English word *finish* is used, then we assign meaning (b). This idea has been taken up by quite a number of researchers who have developed word-sense disambiguation systems using bilingual texts [97, 31, 9, 86]. Dagan et al. [29] even suggest that this idea could be extended to multiple languages and to other tasks of linguistic analysis. To a large degree, this thesis can be viewed as fully taking up that challenge.

In particular, we extend the vision of multilingual learning to a broad range of classical NLP problems, including part-of-speech tagging, grammar induction, and morphological analysis. In all cases, we show substantial gains over state-of-the-art unsupervised models. In one case, the gap between unsupervised and supervised performance is cut by over two-thirds, without any human annotation. We show for the first time that performance *continues* to improve as additional languages are thrown into the mix. We also demonstrate for the first time that cross-lingual syntactic structures can be modeled while still allowing significant language variation. Finally, we present the first statistical model to automatically decipher a lost language. This model also demonstrates that multilingual analysis can be effective even in the absence of parallel corpora.

1.6 Thesis Overview

Chapter 2 provides full details regarding our two multilingual part-of-speech models. The first model was designed exclusively for bilingual data, whereas the second model can easily scale up to large numbers of languages. We compare the performance of both models, and conclude that when multiple languages are available the latter model is preferable. However, in certain bilingual circumstances the first, simpler model may yield better results. We present several experiments designed to answer some fundamental questions regarding multilingual learning. First, what is the impact of language relatedness on performance? And second, how does the number of languages impact average performance? Much of the work in this chapter was originally described by Snyder et al. [111], Snyder et al. [112], and Naseem et al. [85].

Chapter 3 considers the application of multilingual learning to grammar induction. The main challenge for this problem is *representational.* How can we simultaneously represent two distinct parse trees which may be related in complex, unpredictable ways? We adapt a flexible, yet computationally tractable tree alignment formalism to a Bayesian probabilistic setting. We tested our bilingual grammar induction model on three language pairs, and show a 19% reduction in error relative to a state-of-the-art baseline and a theoretical upper bound. Much of the work in this chapter was originally described by Snyder and Barzilay [110].

Chapter 4 considers the difficult problem of lost language decipherment. In this scenario, we are given some texts in a dead language with no direct knowledge of the writing system or any other features of the language. Our goal is to use knowledge from known related languages to recover information about the alphabet and vocabulary of the lost language. The key departure from previous chapters is that in this scenario we do not have access to multilingual parallel data. Instead, our model must ferret out language-wide structural similarities between the the lost and known languages. We conduct numerous experiments, all focused on the ancient language of Ugaritic. We show that our model can automatically decipher a large portion of this dead language. Much of the work in this chapter was originally described by Snyder and Barzilay [109].

Chapter 5 concludes the thesis with some final thoughts and directions for future work. Finally, we note that the exposition throughout this thesis will assume basic familiarity with probabilistic models, though not with the particular tasks we study.

Chapter 2

Unsupervised Multilingual Part-of-speech Tagging

We were all taught, long ago in some elementary school classroom, that verbs are words for "actions" and that nouns are words for "things." Eventually, we learned to distinguish among *many* different parts-of-speech (pronouns, articles, adjectives, adverbs, to name a few). We also realized that the distinctions can be more subtle than we at first thought. In some sentences the nouns convey much more "action" than the verbs ("seeing is believing"). Nevertheless, as far as latent linguistic structure goes, part-of-speech categories are fairly straightforward. As such, their automatic prediction serves as a first test of the multilingual learning framework.

More formally, this chapter deals with the classical NLP task of *part-of-speech tagging* in an unsupervised setting. For this task, we are given written texts in some language without any human annotation. We are also given a dictionary which lists the possible parts-of-speech for some (but perhaps not all) of the words. Our goal is to automatically assign the most likely part-of-speech to each word in the written text, depending on its context. As an example, consider the following English sentence as input:

That factory can definitely can a good can.

Our goal would be to label the sentence with a sequence of part-of-speech tags: [1]

DT	NN	AUX	ADV	VB	DT	ADJ	NN
That	factory	can	definitely	can	a	good	can

Note that the word "can" serves as three distinct parts-of-speech in this sentence, an auxiliary verb, a regular verb, and a noun. The goal of a part-of-speech tagger is to resolve this ambiguity by examining the surrounding words.

2.1 Chapter Overview

Section 2.2 gives a broad introduction to the chapter. We argue that a multilingual approach will lead to more accurate part-of-speech predictions. We sketch two multilingual models, the first of which is designed for language *pairs*, and the second of which scales to larger language groupings, and we summarize our main experimental findings. Section 2.3 compares our approach with previous work on multilingual learning and unsupervised part-of-speech tagging. Section 2.4 presents an overview of our two approaches to modeling multilingual tag sequences. Section 2.5 presents our bilingual model, and section 2.6 details our corresponding inference procedure. Section 2.7 presents our multilingual model, and section 2.8 details our corresponding inference procedure. Section 2.9 provides implementation details for both models. Section 2.10 describes corpora used in the experiments, preprocessing steps and various evaluation scenarios. The results of the experiments and their analysis are given in Sections 2.11, and 2.12. We summarize our contributions and consider directions for future work in Section 2.13.

2.2 Introduction

In this chapter, we explore the application of multilingual learning to unsupervised part-of-speech tagging. The underlying idea throughout this chapter is that the

[1] DT = determiner, NN = noun, AUX = auxiliary verb, VB = verb, ADV = adverb, ADJ = adjective.

patterns of ambiguity in part-of-speech tag assignments differ across languages. At the lexical level, a word with part-of-speech tag ambiguity in one language may correspond to an unambiguous word in the other language. As we saw above, the word "can" in English may function as an auxiliary verb, a noun, or a regular verb. However, many other languages express these different senses with three distinct lexemes. Languages also differ in their patterns of structural ambiguity. For example, the presence of an article in English (e.g. "the") greatly reduces the ambiguity of the succeeding tag. In languages without articles, however, this constraint is obviously absent. The key idea of multilingual learning is that by combining natural cues from multiple languages, the structure of each becomes more apparent.

Even in expressing the same meaning, languages take different syntactic routes, leading to cross-lingual variation in part-of-speech patterns. Therefore, an effective multilingual model must accurately represent common linguistic structure, yet remain flexible to the idiosyncrasies of each language. This tension only becomes stronger as additional languages are added to the mix. Thus, a key challenge of multilingual learning is to capture cross-lingual correlations while preserving individual language tagsets, tag selections, and tag orderings.

In this chapter, we explore two different approaches for modeling cross-lingual correlations. The first approach directly merges pairs of tag sequences into a single bilingual sequence, employing joint distributions over aligned tag-pairs; for unaligned tags, language-specific distributions are still used. The second approach models multilingual context using latent variables instead of explicit node merging. For a group of aligned words, the multilingual context is encapsulated in the value of a corresponding latent variable. Conditioned on the latent variable, the tagging decisions for each language remain independent. In contrast to the first model, the architecture of the hidden variable model allows it to scale gracefully as the number of languages increases.

Both approaches are formulated as hierarchical Bayesian models with an underlying trigram HMM substructure for each language. The first model operates as a simple directed graphical model with only one additional *coupling* parameter beyond the transition and emission parameters used in monolingual HMMs. The latent variable model, on the other hand, is formulated as a non-parametric model; it can be viewed as performing multilingual clustering on aligned sets of tag variables. Each latent variable value indexes a separate distribution on tags for each language, appropriate to the given context. For both models, we perform inference using Markov Chain Monte Carlo sampling techniques.

We evaluate our models on a parallel corpus of eight languages: Bulgarian, Czech, English, Estonian, Hungarian, Romanian, Serbian, and Slovene. We consider a range of scenarios that vary from combinations of bilingual models to a single model that is jointly trained on all eight languages. Our results show consistent and robust improvements over a monolingual baseline for almost all combinations of languages. When a complete tag lexicon is available and the latent variable model is trained using eight languages, average performance increases from 91.1% accuracy to 95%, more than halving the gap between unsupervised and supervised performance. In more realistic cases, where the lexicon is restricted to only frequently occurring words, we see even larger gaps between monolingual and multilingual performance. In one such scenario, average multilingual performance increases to 82.8% from a monolingual baseline of 74.8%. For some language pairs, the improvement is especially noteworthy; for instance, in complete lexicon scenario, Serbian improves from 84.5% to 94.5% when paired with English.

We find that in most scenarios the latent variable model achieves higher performance than the merged structure model, even when it too is restricted to pairs of languages. Moreover the hidden variable model can effectively accommodate large numbers of languages which makes it a more desirable framework for multilingual learning. However, we observe that the latent variable model is somewhat sensitive to lexicon coverage. The performance of the merged structure model, on the other hand, is more robust in this respect. In the case of the drastically reduced lexicon (with 100 words only), its performance is clearly better than the hidden variable model. This indicates that the merged structure model might be a better choice for the languages that lack lexicon resources.

A surprising discovery of our experiments is the marked variation in the level of improvement across language pairs. If the best pairing for each language is chosen by an oracle, average bilingual performance reaches 95.4%, compared to average performance of 93.1% across all pairs. Our experiments demonstrate that this variability is influenced by cross-lingual links between languages as well as by the model under consideration. We identify several factors that contribute to the success of language pairings, but none of them can uniquely predict which supplementary language is most helpful. These results suggest that when multi-parallel corpora are available, a model that simultaneously exploits all the languages – such as the latent variable model proposed here – is preferable to a strategy that selects one of the bilingual models. We found that performance tends to improves steadily as the number of available languages increases.

In realistic scenarios, tagging resources for some number of languages may already be available. Our models can easily exploit any amount of tagged data in any subset of available languages. As our experiments show, as annotation is added, performance increases even for those languages lacking resources.

2.3 Related Work

We identify two broad areas of related work: multilingual learning and inducing part-of-speech tags without labeled data. Our discussion of multilingual learning focuses on unsupervised approaches that incorporate two or more languages. We then describe related work on unsupervised and semi-supervised models for part-of-speech tagging.

2.3.1 Multilingual Learning

The potential of multilingual data as a rich source of linguistic knowledge has been recognized since the early days of empirical natural language processing. Because patterns of ambiguity vary greatly across languages, unannotated multilingual data can serve as a learning signal in an unsupervised setting. We are especially interested in methods to leverage more than two languages jointly, and compare our approach with relevant prior work.

Multilingual learning may also be applied in a semi-supervised setting, typically by projecting annotations across a parallel corpus to another language where such resources do not exist [127, 31, 92, 124]. As our primary focus is on the unsupervised induction of cross-linguistic structures, we do not address this area.

Bilingual Learning

Word sense disambiguation (WSD) was among the first successful applications of automated multilingual learning [29, 14]. Lexical ambiguity differs across languages – each sense of a polysemous word in one language may translate to a distinct counterpart in another language. This makes it possible to use aligned foreign-language words as a source of noisy supervision. Bilingual data has been leveraged in this way in a variety of WSD models [14, 97, 86, 31, 73, 9], and the quality of supervision provided by multilingual data closely approximates that of manual annotation [86]. Polysemy is one source of ambiguity for part-of-speech tagging; thus our model implicitly leverages multilingual WSD in the context of a higher-level syntactic analysis.

Multilingual learning has previously been applied to syntactic analysis; a pioneering effort was the inversion transduction grammar of Wu [121]. This method is trained on an unannotated parallel corpus using a probabilistic bilingual lexicon and deterministic constraints on bilingual tree structures. The inside-outside algorithm [3] is used to learn parameters for manually specified bilingual grammar. These ideas were extended by subsequent work on synchronous grammar induction and hierarchical phrase-based translation [123, 22].

One characteristic of this family of methods is that they were designed for inherently multilingual tasks such as machine translation and lexicon induction. While we share the goal of *learning* from multilingual data, we seek to induce monolingual syntactic structures that can be applied even when multilingual data is unavailable.

In this respect, our approach is closer to the unsupervised multilingual grammar induction work of Kuhn [70]. Starting from the hypothesis that trees induced over parallel sentences should exhibit cross-lingual structural similarities, Kuhn uses word-level alignments to constrain the set of plausible syntactic constituents. These constraints are implemented through hand-crafted deterministic rules, and are incorporated in expectation-maximization grammar induction to assign zero likelihood to illegal bracketings. The probabilities of the productions are then estimated separately for each language, and can be applied to monolingual data directly. Kuhn shows that this form of multilingual training yields better monolingual parsing performance.

Our methods incorporate cross-lingual information in a fundamentally different manner. Rather than using hand-crafted deterministic rules – which may require modification for each language pair – we estimate probabilistic multilingual patterns directly from data. Moreover, the estimation of multilingual patterns is incorporated directly into the tagging model itself.

Beyond Bilingual Learning

While most work on multilingual learning focuses on bilingual analysis, some models operate on more than one pair of languages. For instance, Genzel [41] describes a method for inducing a multilingual lexicon from a group of related languages. This work first induces bilingual models for each pair of languages and then combines them. We take a different approach by simultaneously learning from all languages, rather than combining bilingual results.

A related thread of research is multi-source machine translation [89, 118, 24, 21, 8] where the goal is to translate from multiple source languages to a single target language. By using multi-source corpora, these systems alleviate sparseness and increase translation coverage, thereby improving overall translation accuracy. Typically, multi-source translation systems build separate bilingual models and then select a final translation from their output. For instance, a method developed by Och and Ney [89] generates several alternative translations from source sentences expressed in different languages and selects the most likely candidate. Cohn and Lapata [24] consider a different generative model: rather than combining alternative sentence translations in a post-processing step, their model estimates the target phrase translation distribution by marginalizing over multiple translations from various source languages. While their model combines multilingual information at the phrase level, at its core are estimates for phrase tables that are obtained using bilingual models.

In contrast, we present an approach for unsupervised multilingual learning that builds a single joint model across all languages. This makes maximal use of unlabeled data and sidesteps the difficult problem of combining the output of multiple bilingual systems without supervision.

2.3.2 Unsupervised Part-of-Speech Tagging

Unsupervised part-of-speech tagging involves predicting the tags for words, without annotations of the correct tags for any word tokens. Generally speaking, the unsupervised setting does permit the use of declarative knowledge about the relationship between tags and word *types*, in the form of a dictionary of the permissible tags for the most common words. This setup is referred to as “semi-supervised” by Toutanova and Johnson [116], but is considered “unsupervised” in most other papers on the topic [43]. Our evaluation considers tag dictionaries of varying levels of coverage.

Since the work of Merialdo [79], the hidden Markov model (HMM) has been the most common representation[2] for unsupervised tagging [4]. Part-of-speech tags are encoded as a linear chain of hidden variables, and words are treated as emitted observations. Recent advances include the use of a fully Bayesian HMM [61, 43], which places prior distributions on tag transition and word-emission probabilities. Such Bayesian priors permit integration over parameter settings, yielding models that perform well across a range of settings. This is particularly important in the case of

[2] In addition to the basic HMM architecture, other part-of-speech tagging approaches have been explored [13, 81]

small datasets, where many of the counts used for maximum-likelihood parameter estimation will be sparse. The Bayesian setting also facilitates the integration of other data sources, and thus serves as the departure point for our work.

Several recent papers have explored the development of alternative training procedures and model structures in an effort to incorporate more expressive features than permitted by the generative HMM. Smith and Eisner [106] maintain the HMM structure, but incorporate a large number of overlapping features in a conditional log-linear formulation. *Contrastive estimation* is used to provide a training criterion which maximizes the probability of the observed sentences compared to a set of similar sentences created by perturbing word order. The use of a large set of features and a discriminative training procedure led to strong performance gains.

Toutanova and Johnson [116] propose an LDA-style model for unsupervised part-of-speech tagging, grouping words through a latent layer of ambiguity classes. Each ambiguity class corresponds to a set of permissible tags; in many languages this set is tightly constrained by morphological features, thus allowing an incomplete tagging lexicon to be expanded. Haghigi and Klein [47] also use a variety of morphological features, learning in an undirected Markov Random Field that permits overlapping features. They propagate information from a small number of labeled “prototype” examples using the distributional similarity between prototype and non-prototype words.

Our focus is to effectively incorporate multilingual evidence, and we require a simple model that can easily be applied to multiple languages with widely varying structural properties. We view this direction as orthogonal to refining monolingual tagging models for any particular language.

2.4 Model Overviews

The motivating hypothesis of this work is that patterns of ambiguity at the part-of-speech level differ across languages in systematic ways. By considering multiple languages simultaneously, the total inherent ambiguity can be reduced in each lan-

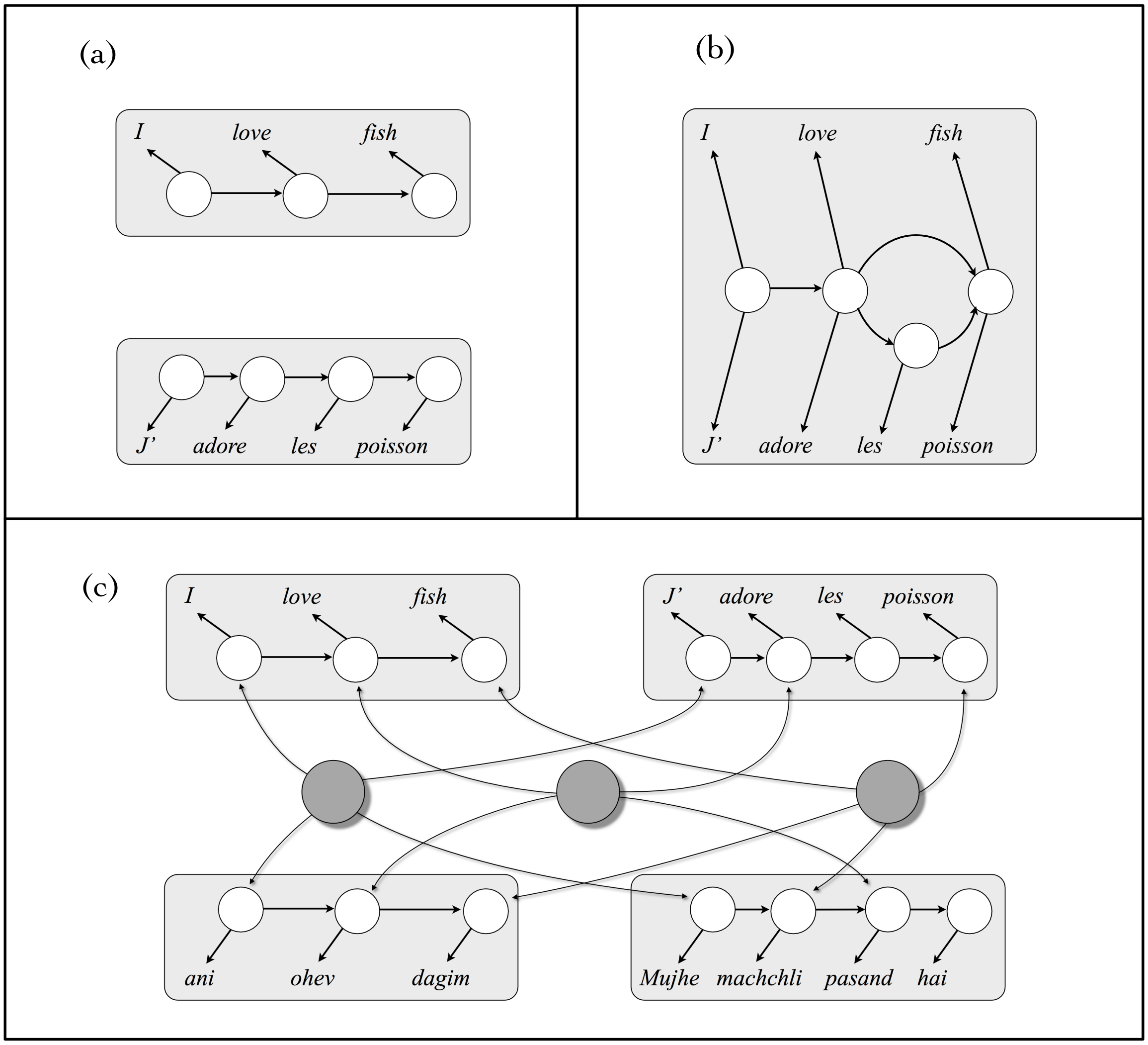


Figure 2-1: Example graphical structures of (a) two standard monolingual HMMs, (b) our merged node model, and (c) our latent variable model with three superlingual variables.

guage. But with the potential advantages of leveraging multilingual information comes the challenge of respecting language-specific characteristics such as tag inventory, selection and order. To this end, we develop models that jointly tag parallel streams of text in multiple languages, while maintaining language-specific tag sets and parameters over transitions and emissions.

Part-of-speech tags reflect the syntactic and semantic function of the tagged words. Across languages, pairs of word tokens that are known to share semantic or syntactic function should have tags that are related in systematic ways. The *word alignment* task in machine translation is to identify just such pairs of words in parallel sentences. Aligned word pairs serve as the cross-lingual anchors of our model, allowing information to be shared via joint tagging decisions. Research in machine translation has produced robust tools for identifying word alignments; we use such a tool as a black box and treat its output as a fixed, observed property of the parallel data.

Given a set of parallel sentences, we posit a hidden Markov model (HMM) for each language, where the hidden states represent the tags and the emissions are the words. In the unsupervised monolingual setting, inference on the part-of-speech tags is performed jointly with estimation of parameters governing the relationship between tags and words (the *emission* probabilities) and between consecutive tags (the *transition* probabilities). Our multilingual models are built upon this same structural foundation, so that the emission and transition parameters retain an identical interpretation as in the monolingual setting. Thus, these parameters can be learned on parallel text and later applied to monolingual data.

We consider two alternative approaches for incorporating cross-lingual information. In the first model, the tags for aligned words are merged into single bi-tag nodes; in the second, latent variable model, an additional layer of hidden *superlingual tags* instead exerts influence on the tags of clusters of aligned words. The first model is primarily designed for bilingual data, while the second model operates over any number of languages. Figure 2-1 provides a graphical model representation of the monolingual, merged node, and latent variable models instantiated over a single parallel sentence.

Both the merged node and latent variable approaches are formalized as hierarchical Bayesian models. This provides a principled probabilistic framework for integrating multiple sources of information, and offers well-studied inference techniques. table 2.1 summarizes the mathematical notation used throughout this section. We now describe each model in depth.

2.5 Bilingual Unsupervised Tagging: A Merged Node Model

In the bilingual merged node model, cross-lingual context is incorporated by creating joint bi-tag nodes for aligned words. It would be too strong to insist that aligned words have an identical tag; indeed, it may not even be appropriate to assume that two languages share identical tag sets. However, when two words are aligned, we do want to choose their tags jointly. To enable this, we allow the values of the bi-tag nodes to range over all possible tag pairs $\langle t, t' \rangle \in T \times T'$, where T and T' represent the tagsets for each language.

The tags t and t' need not be identical, but we do believe that they are systematically related. This is modeled using a coupling distribution ω, which is multinomial over all tag pairs. The parameter ω is combined with the standard transition distribution ϕ in a product-of-experts model. Thus, the aligned tag pair $\langle y_i, y'_j \rangle$ is conditioned on the predecessors y_{i-1} and y'_{j-1}, as well as the coupling parameter $\omega(y_i, y'_j)$.[3] The coupled bi-tag nodes serve as bilingual "anchors" – due to the Markov dependency structure, even unaligned words may benefit from cross-lingual information that propagates from these nodes.

We now present a generative account of how the words in each sentence and the parameters of the model are produced. This generative story forms the basis of our

[3] While describing the merged node model, we consider only the two languages ℓ and ℓ', and use a simplified notation in which we write $\langle y, y' \rangle$ to mean $\langle y^{\ell}, y^{\ell'} \rangle$. Similar abbreviations are used for the language-indexed parameters.

Notation used in both models

$\mathbf{x}^{\ell}$	–	The sequence of words in language ℓ.
$\mathbf{y}^{\ell}$	–	The corresponding part-of-speech tag sequence in language ℓ.
x_i^{ℓ}	–	The i^{th} word token in language ℓ.
y_i^{ℓ}	–	The i^{th} part-of-speech tag in language ℓ.
$\mathbf{a}^{\ell,\ell'}$	–	The word alignments for the language pair $\langle \ell, \ell' \rangle$.
ϕ_t^{ℓ}	–	The transition distribution (over tags), conditioned on tag t in language ℓ. We describe a bigram transition model, though our implementation uses trigrams (without bigram interpolations); the extension is trivial.
θ_t^{ℓ}	–	The emission distribution (over words), conditioned on tag t in language ℓ.
ϕ_0	–	The parameter of the symmetric Dirichlet prior on the transition distributions.
θ_0	–	The parameter of the symmetric Dirichlet prior on the emission distributions.

Notation used in the merged node model

ω	–	A coupling parameter that assigns probability mass to each pair of aligned tags.
ω_0	–	A Dirichlet prior on the coupling parameter.
A_b	–	Distribution over bilingual alignments.

Notation used in the latent variable model

π	–	A multinomial over the superlingual tags $\mathbf{z}$.
α	–	The *concentration parameter* for π, controlling how much probability mass is allocated to the first few values.
z_j	–	The setting of the j^{th} superlingual tag, ranging over the set of integers, and indexing a distribution set in Ψ.
$\Psi_z = \langle \psi_z^1, \psi_z^2, \ldots, \psi_z^n \rangle$	–	The z^{th} set of distributions over tags in all languages ℓ_1 through ℓ_n.
G_0	–	A base distribution from which the Ψ_z are drawn, whose form is a set of n symmetric Dirichlet distributions each with a parameter ψ_0.
A_m	–	Distribution over multilingual alignments.

Table 2.1: Summary of notation used in the description of both models. As each sentence is treated in isolation (conditioned on the parameters), the sentence indexing is left implicit.

sampling-based inference procedure.

2.5.1 Generative Story

Our generative story assumes the existence of two tagsets, T and T', and two vocabularies W and W' – one of each for each language. For ease of exposition, we formulate our model with bigram tag dependencies. However, in our experiments we used a trigram model (without any bigram interpolation), which is a trivial extension of the described model.

1. **Transition and Emission Parameters**. For each tag $t \in T$, draw a *transition* distribution ϕ_t over tags T, and an *emission* distribution θ_t over words W. Both the transition and emission distributions are multinomial, so they are drawn from their conjugate prior, the Dirichlet [39]. We use symmetric Dirichlet priors, which encode only an expectation about the uniformity of the induced multinomials, but not do encode preferences for specific words or tags.

   For each tag $t \in T'$, draw a *transition* distribution ϕ'_t over tags T', and an *emission* distribution θ'_t over words W', both from symmetric Dirichlet priors.

2. **Coupling Parameter**. Draw a bilingual *coupling* distribution ω over tag pairs $T \times T'$. This distribution is multinomial with dimension $|T| \cdot |T'|$, and is drawn from a symmetric Dirichlet prior ω_0 over all tag pairs.

3. **Data**. For each bilingual parallel sentence:

   (a) Draw an alignment $\mathbf{a}$ from a bilingual alignment distribution A_b. The alignments and their distribution are defined formally below.

   (b) Draw a bilingual sequence of part-of-speech tags $(y_1, ..., y_m)$, $(y'_1, ..., y'_n)$ according to: $P((y_1, ..., y_m), (y'_1, ..., y'_n)|\mathbf{a}, \phi, \phi', \omega)$.[4] This joint distribution

[4] We use a special end state, rather than explicitly modeling sentence length. Thus the values of m and n are determined stochastically.

thus conditions on the alignment structure, the transition probabilities for both languages, and the coupling distribution; a formal definition is given in Formula 2.1.

(c) For each part-of-speech tag y_i in the first language, emit a word from the vocabulary W: $x_i \sim \theta_{y_i}$,

(d) For each part-of-speech tag y'_j in the second language, emit a word from the vocabulary W': $x'_j \sim \theta'_{y'_j}$.

This completes the outline of the generative story. We now provide more detail on how alignments are handled, and on the distribution over coupled part-of-speech tag sequences.

Alignments

An alignment $\mathbf{a}$ defines a bipartite graph between the words $\mathbf{x}$ and $\mathbf{x}'$ in two parallel sentences . In particular, we represent $\mathbf{a}$ as a set of integer pairs, indicating the word indices. Crossing edges are not permitted, as these would lead to cycles in the resulting graphical model; thus, the existence of an edge (i, j) precludes any additional edges $(i + a, j - b)$ or $(i - a, j + b)$, for $a, b \geq 0$. From a linguistic perspective, we assume that the edge (i, j) indicates that the words x_i and x'_j share some syntactic and/or semantic role in the bilingual parallel sentences.

From the perspective of the generative story, alignments are treated as draws from a distribution A_b. Since the alignments are always observed, we can remain agnostic about the distribution A_b, except to require that it assign zero probability to alignments which either: (i) align a single index in one language to multiple indices in the other language or (ii) contain crossing edges. The resulting alignments are thus one-to-one, contain no crossing edges, and may be sparse or even possibly empty. Our technique for obtaining alignments that display these properties is described in Section 2.10.2.

Generating Tag Sequences

In a standard hidden Markov model for part-of-speech tagging, the tags are drawn as a Markov process from the transition distribution. This permits the probability of a tag sequence to factor across the time steps. Our model employs a similar factorization: the tags for unaligned words are drawn from their predecessor's transition distribution, while joined tag nodes are drawn from a product involving the coupling parameter and the transition distributions for both languages.

More formally, given an alignment **a** and sets of transition parameters ϕ and ϕ', we factor the conditional probability of a bilingual tag sequence $(y_1, ..., y_m), (y'_1, ..., y'_n)$ into transition probabilities for unaligned tags, and joint probabilities over aligned tag pairs:

$$\begin{aligned} P((y_1, ..., y_m),\ (y'_1, ..., y'_n)|\mathbf{a}, \phi, \phi', \omega) = & \prod_{\text{unaligned } i} \phi_{y_{i-1}}(y_i) \prod_{\text{unaligned } j} \phi'_{y'_{j-1}}(y'_j) \\ & \prod_{(i,j)\in\mathbf{a}} P(y_i, y'_j | y_{i-1}, y'_{j-1}, \phi, \phi', \omega). \end{aligned} \tag{2.1}$$

Because the alignment contains no crossing edges, we can still model the tags as generated sequentially by a stochastic process. We define the distribution over aligned tag pairs to be a product of each language's transition probability and the coupling probability:

$$P(y_i, y'_j | y_{i-1}, y'_{j-1}, \phi, \phi', \omega) = \frac{\phi_{y_{i-1}}(y_i)\ \phi'_{y'_{j-1}}(y'_j)\omega(y_i, y'_j)}{Z} \tag{2.2}$$

The normalization constant here is defined as:

$$Z = \sum_{y,y'} \phi_{y_{i-1}}(y)\ \phi'_{y'_{j-1}}(y')\ \omega(y, y')$$

This factorization allows the language-specific transition probabilities to be shared across aligned and unaligned tags.

Another way to view this probability distribution is as a product of three experts: the two transition parameters and the coupling parameter. Product-of-expert

Algorithm 1: Gibbs sampler for merged-node part-of-speech tagging model.

```
Input: A bilingual corpus consisting of word sequences $(\mathbf{x}, \mathbf{x}')$ (spanning
       multiple sentences). Corresponding word-level alignments $\mathbf{a}$
Output: 200 samples of corresponding part-of-speech tag sequences $(\mathbf{y}, \mathbf{y}')$
Initialize part-of-speech tags;
for $r \leftarrow 1$ to 200 do
    foreach unaligned word $x_i \in \mathbf{x}$ (i.e. $\neg\exists j : (i,j) \in \mathbf{a}$) do
        Sample tag $y_i$                                        // Section 2.6.1
    foreach unaligned word $x'_j \in \mathbf{x}'$ (i.e. $\neg\exists i : (i,j) \in \mathbf{a}$) do
        Sample tag $y'_j$                                       // Section 2.6.1
    foreach aligned word-pair $x_i, x'_j$ (i.e. $(i,j) \in \mathbf{a}$) do
        Sample tag-pair $y_i, y'_j$                             // Section 2.6.2
```

models [56] allow each information source to exercise very strong negative influence on the probability of tags that they consider to be inappropriate, as compared with additive models. This is ideal for our setting, as it prevents the coupling distribution from causing the model to generate a tag that is unacceptable from the perspective of the monolingual transition distribution. In preliminary experiments we found that a multiplicative approach was strongly preferable to additive models.

2.6 Merged Node Model: Inference

The goal of our inference procedure is to obtain transition and emission parameters θ and ϕ that can be applied to monolingual test data. Ideally we would choose the parameters that have the highest marginal probability, conditioned on the observed words $\mathbf{x}$ and alignments $\mathbf{a}$:

$$\hat{\theta}, \hat{\phi} = \underset{\theta,\phi}{\operatorname{argmax}} \int P(\theta, \phi, \mathbf{y}, \omega | \mathbf{x}, \mathbf{a}, \theta_0, \phi_0, \omega_0) d\mathbf{y} d\omega$$

While the structure of our model permits us to decompose the joint probability, it is not possible to analytically marginalize all of the hidden variables. We resort to standard Monte Carlo approximation, in which marginalization is performed through sampling. By repeatedly sampling individual hidden variables according to the appropriate distributions, we obtain a Markov chain that is guaranteed to converge to a stationary distribution centered on the desired posterior. Thus, after an initial burn-in phase, we can use the samples to approximate a marginal distribution over any desired parameter [42].

The core element of our inference procedure is Gibbs sampling [40]. Gibbs sampling begins by randomly initializing all unobserved random variables; at each iteration, each random variable u_i is then sampled from the conditional distribution $P(u_i|\mathbf{u}_{-i})$, where $\mathbf{u}_{-i}$ refers to all variables other than u_i. Eventually, the distribution over samples drawn from this process will converge to the unconditional joint distribution $P(\mathbf{u})$ of the unobserved variables. When possible, we avoid explicitly sampling variables which are not of direct interest, but rather integrate over them. This technique is known as *collapsed sampling*; it is guaranteed never to increase sampling variance, and will often reduce it [74].

In the merged node model, we need sample only the part-of-speech tags and the priors. We are able to exactly marginalize the emission parameters θ and approximately marginalize the transition and coupling parameters ϕ and ω (the approximations are required due to the re-normalized product of experts — see below for more details). We draw repeated samples of the part-of-speech tags, and construct a sample-based estimate of the underlying tag sequence. After sampling, we construct maximum *a posteriori* estimates of the parameters of interest for each language, θ and ϕ. See algorithm 1 for an overview of the Gibbs sampler. In the remainder of the section we describe the individual sampling equations.

2.6.1 Sampling Unaligned Tags

For unaligned part-of-speech tags, the conditional sampling equations are similar to the monolingual Bayesian hidden Markov model. The posterior probability of each tag decomposes into two factors, one for transitions and one for emissions. To arrive at this decomposition we apply Bayes' rule:

$$\begin{aligned}
&P(y_i \mid \mathbf{y}_{-i}, \mathbf{y}', \mathbf{x}, \mathbf{x}', \theta_0, \phi_0, \phi_0', \omega_0) \\
&\quad = \frac{P(x_i \mid \mathbf{y}, \mathbf{y}', \mathbf{x}_{-i}, \mathbf{x}', \theta_0, \phi_0, \phi_0', \omega_0) \cdot P(y_i \mid \mathbf{y}_{-i}, \mathbf{y}', \mathbf{x}_{-i}, \mathbf{x}', \theta_0, \phi_0, \phi_0', \omega_0)}{P(x_i \mid \mathbf{y}_{-i}, \mathbf{y}', \mathbf{x}_{-i}, \mathbf{x}', \theta_0, \phi_0, \phi_0', \omega_0)} \\
&\quad \propto P(x_i \mid \mathbf{y}, \mathbf{x}_{-i}, \theta_0) \cdot P(y_i \mid \mathbf{y}_{-i}, \mathbf{y}', \phi_0, \phi_0', \omega_0)
\end{aligned}$$

The notation $\mathbf{y}_{-i}$ denotes all the sampled tags other than y_i and $\mathbf{x}_{-i}$ denotes all the observed words besides x_i. In the last equality we exploited several conditional independencies of our model. In particular, the transition factor is conditionally independent of the words in either language, and the emission factor is conditionally independent of the words and tags of the *other* language. We now derive the form of each of these two factors, marginalizing out the emission parameters θ, the transition parameters ϕ, and the coupling parameter ω.

For the emission factor, we can *exactly* marginalize out the emission distribution θ, whose prior is Dirichlet with hyperparameter θ_0. The resulting distribution is a ratio of counts, where the prior acts as a pseudo-count:

$$P(x_i|\mathbf{y}, \mathbf{x}_{-i}, \theta_0, \phi_0', \omega_0) = \int_{\theta_{y_i}} \theta_{y_i}(x_i) P(\theta_{y_i}|\mathbf{y}, \mathbf{x}_{-i}, \theta_0) d\theta_{y_i} = \frac{n(y_i, x_i) + \theta_0}{n(y_i) + |W_{y_i}|\theta_0} \qquad (2.3)$$

Here, $n(y_i)$ is the number of occurrences of the tag y_i in $\mathbf{y}_{-i}$, $n(y_i, x_i)$ is the number of occurrences of the tag-word pair (y_i, x_i) in $(\mathbf{y}_{-i}, \mathbf{x}_{-i})$, and W_{y_i} is the set of word types in the vocabulary W that can take tag y_i. The integral is tractable due to Dirichlet-multinomial conjugacy, and an identical marginalization was applied in the monolingual Bayesian HMM of [43].

The transition factor is more complicated. We start by again applying Bayes' rule:

$$\begin{aligned}
&P(y_i \mid \mathbf{y}_{-i}, \mathbf{y}', \phi_0) \\
&\quad \propto P(y_{i+1} \mid \mathbf{y}_{-(i+1)}, \mathbf{y}', \phi_0, \phi_0', \omega_0) \cdot P(y_i \mid \mathbf{y}_{-(i,i+1)}, \mathbf{y}', \phi_0, \phi_0', \omega_0)
\end{aligned}$$

Here, $\mathbf{y}_{-(i,i+1)}$ denotes all tags $\mathbf{y}$ besides y_i and y_{i+1}. The first factor corresponds to the transition from y_i to y_{i+1}, and the second factor corresponds to the transition from y_{i-1} to y_i. For both factors, we seek to marginalize out the transition distribution ϕ. This is difficult to do exactly, as the tags of the other language, $\mathbf{y}'$, exert a subtle influence on the probabilities of the tags $\mathbf{y}$, through the renormalized product-of-experts (equation 2.2). Nevertheless, we approximate the marginal using monolingual transition counts:

$$\begin{aligned} &P(y_i \mid \mathbf{y}_{-(i,i+1)}, \mathbf{y}', \phi_0, \phi_0', \omega_0) = \\ &\qquad \int_{\phi_{y_{i-1}}} \phi_{y_{i-1}}(y_i)\; P(\phi_{y_{i-1}} \mid \mathbf{y}_{-(i,i+1)}, \mathbf{y}', \phi_0, \phi_0', \omega_0)\; d\phi_{y_{i-1}} \approx \frac{n(y_{i-1}, y_i) + \phi_0}{n(y_{i-1}) + |T|\phi_0} \end{aligned} \tag{2.4}$$

The factors here are similar to the emission probability: $n(y_{i-1})$ is the number of occurrences of the tag y_i in $\mathbf{y}_{-(i,i+1)}$, $n(y_{i-1}, y_i)$ is the number of occurrences of the tag sequence (y_{i-1}, y_i), and T is the tagset. We can understand this approximation in the following way. Each aligned tag-pair (y, y') in the corpus was generated by a renormalized product of three factors: ϕ, ϕ', and ω. However, for the purposes of integrating over the transition parameter ϕ, we treat all tags $\mathbf{y}$ as having been generated *solely* by ϕ. This allows us to treat these tags as observed draws from a multinomial, which in turn allows the use of the standard closed-forms given by Dirichlet-multinomial conjugacy. We will use similar approximations when marginalizing over ϕ' and ω.

The probability for the transition from i to $i+1$ is exactly analogous when y_{i+1} is also unaligned. Here we consider the more complex case where y_{i+1} is aligned to some tag y'_{j+1} in the other language. The sampling formulas must now account for the effect of y_i on the joint probability of the succeeding tags, which is no longer a simple multinomial transition probability. In this case, we approximate the transition probability as:

$$P(y_{i+1} \mid \mathbf{y}_{-(i+1)}, \mathbf{y}', \phi_0, \phi'_0, \omega_0) \;\propto\; P(y_{i+1}, y'_{j+1} \mid \mathbf{y}_{-i}, \mathbf{y}'_{-(j+1)}, \phi_0, \phi'_0, \omega_0) \;=$$
$$\int_{\omega,\phi,\phi'} \left[\frac{\omega(y_{i+1}, y'_{j+1})\phi_{y_i}(y_{i+1})\phi'_{y'_j}(y'_{j+1})}{Z}\right] P(\omega, \phi_{y_i}, \phi'_{y'_j} | \mathbf{y}_{-(i+1)}, \mathbf{y}'_{-(j+1)}, \phi_0, \phi'_0, \omega_0) d\omega d\phi d\phi'$$
$$\approx \left(\frac{n(y_i, y_{i+1}) + \phi_0}{n(y_i) + |T|\phi_0}\right) \cdot \left(\frac{n(y'_j, y'_{j+1}) + \phi_0}{n(y'_j) + |T|\phi_0}\right) \cdot \left(\frac{n(y_{i+1}, y'_{j+1}) + \omega_0}{N(\mathbf{a}) + |T \times T'|\omega_0}\right) \cdot \left(\frac{1}{Z'}\right)$$
$$\propto \left(\frac{n(y_i, y_{i+1}) + \phi_0}{n(y_i) + |T|\phi_0}\right) \cdot \left(\frac{n(y_{i+1}, y'_{j+1}) + \omega_0}{N(\mathbf{a}) + |T \times T'|\omega_0}\right) \cdot \left(\frac{1}{Z'}\right) \tag{2.5}$$

As before (equation 2.4), the transition factor is approximated using the language-specific transition counts. Similarly, the coupling factor is approximated using coupling counts (as if the coupling parameter had produced all aligned tag-pairs on its own): $n(y_{i+1}, y'_{j+1})$ is the number of times tags y_{i+1} and y'_{j+1} were aligned, excluding (y_{i+1}, y'_{j+1}) itself, and $N(\mathbf{a})$ is the total number of alignments. As above, the hyperparameter ω_0 appears as a smoothing factor; in the denominator it is multiplied by the dimensionality of ω, which is the size of the cross-product of the two tagsets. The normalization term Z' is given by:

$$Z' = \sum_{t,t'} \left[\left(\frac{n(y_i, t) + \phi_0}{n(y_i) + |T|\phi_0}\right) \cdot \left(\frac{n(y'_j, t') + \phi_0}{n(y'_j) + |T|\phi_0}\right) \cdot \left(\frac{n(t, t') + \omega_0}{N(\mathbf{a}) + |T \times T'|\omega_0}\right)\right]$$

Intuitively, if the coupling counts are concentrated on a single assignment $y_{i+1} = t, y'_{j+1} = t'$, then the transition from i to $i + 1$ becomes almost irrelevant, since the product-of-experts will be dominated by the coupling term. Conversely, if the coupling counts are indifferent and assigns equal probability to all pairs $\langle t, t' \rangle$, then the sampling formula becomes proportional to the transition factor, which is the same as if y_{i+1} and y_{j+1} were not aligned. In general, as the entropy of the coupling increases, the transition to the succeeding nodes exerts a greater influence on our selection y_i.

2.6.2 Jointly Sampling Aligned Tags

The situation for tags of aligned words is similar. We sample these tags jointly, considering all $|T \times T'|$ possibilities. We begin by decomposing the probability into three factors, using Bayes' rule:

$$\begin{aligned}
&P(y_i, y'_j \mid \mathbf{y}_{-i}, \mathbf{y}'_{-j}, \mathbf{x}, \mathbf{x}', \mathbf{a}, \theta_0, \theta'_0, \phi_0, \phi'_0, \omega_0) \propto \\
&\quad P(x_i \mid \mathbf{y}, \mathbf{x}_{-i}, \theta_0)\; P(x'_j \mid \mathbf{y}', \mathbf{x}'_{-j}, \theta'_0)\; P(y_i, y'_j \mid \mathbf{y}_{-i}, \mathbf{y}'_{-j}, \mathbf{a}, \phi_0, \phi'_0, \omega_0)
\end{aligned}$$

The first two factors are emissions, and are handled identically to the unaligned case (equation 2.3). The expansion of the final, joint factor depends on the alignment of the succeeding tags. If neither of the successors are aligned, we have a product of the bilingual coupling probability and four transition probabilities:

$$\begin{aligned}
&P(y_i, y'_j | \mathbf{y}_{-i}, \mathbf{y}'_{-j}, \phi_0, \phi'_0, \omega_0) \propto \\
&\approx \left(\frac{n(y_{i-1}, y_i) + \phi_0}{n(y_{i-1}) + |T|\phi_0}\right) \cdot \left(\frac{n(y'_{j-1}, y'_j) + \phi_0}{n(y'_{j-1}) + |T|\phi_0}\right) \cdot \left(\frac{n(y_i, y'_j) + \omega_0}{N(\mathbf{a}) + |T \times T'|\omega_0}\right) \\
&\quad \cdot \left(\frac{n(y_i, y_{i+1}) + \phi_0}{n(y_i) + |T|\phi_0}\right) \cdot \left(\frac{n(y'_j, y'_{j+1}) + \phi_0}{n(y'_j) + |T|\phi_0}\right)
\end{aligned}$$

The derivation is similar to equation 2.5, except now the normalization term Z' need not be computed (since it is unaffected by either y_i or y_j). Whenever one or more of the succeeding words is aligned, the sampling formulas must account for the effect of the sampled tag on the joint probability of the succeeding tags, which is no longer a simple multinomial transition probability. In these cases, the final two transition factors are supplemented by the coupling and normalization factors given in equation 2.5.

The alternative to approximately marginalizing all these parameters would be to sample them using a Metropolis-Hastings scheme as in the work by [111]. The use of approximate marginalizations represents a bias-variance tradeoff, where the decreased sampling variance justifies the bias introduced by the approximations, for practical numbers of samples.

2.7 Multilingual Unsupervised Tagging: A Latent Variable Model

The model described in the previous section is designed for bilingual aligned data; as we will see in Section 2.11, it exploits such data very effectively. However, many resources contain more than two languages: for example, Europarl contains eleven, and the Multext-East corpus contains eight. This raises the question of how best to exploit all available resources when multi-aligned data is available.

One possibility would be to train separate bilingual models and then combine their output at test time, either by voting or some other heuristic. However, we believe that cross-lingual information reduces ambiguity at training time, so it would be preferable to learn from multiple languages jointly during training. Indeed, the results in Section 2.11 demonstrate that joint training outperforms such a voting scheme.

Another alternative would be to try to extend the bilingual model developed in the previous section. While such an extension is possible in principle, the merged node model does not scale well in the case of multi-aligned data across more than two languages. Recall that we use merged nodes to represent both tags for aligned words; the state space of such nodes grows as $|T|^L$, exponential in the number of languages L. Similarly, the coupling parameter ω has the same dimension, so that the counts required for estimation become too sparse as the number of languages increases. Moreover, the bi-tag model required removing crossing edges in the word-alignment, so as to avoid cycles. This is unproblematic for pairs of aligned sentences, usually requiring the removal of less than 5% of all edges (see table B.2 in appendix B). However, as the number of languages grows, an increasing number of alignments will have to be discarded.

Instead, we propose a new architecture specifically designed for the multilingual setting. As before, we maintain HMM substructures for each language, so that the learned parameters can easily be applied to monolingual data. However, rather than merging tag nodes for aligned words, we introduce a layer of *superlingual* *tags*. The role of these latent nodes is to capture cross-lingual patterns. Essentially they perform a non-parametric clustering over sets of aligned tags, encouraging multilingual patterns that occur elsewhere in the corpus.

More concretely, for every set of aligned words, we add a superlingual tag with outgoing edges to the relevant part-of-speech nodes. An example configuration is shown in Figure 2-1c. The superlingual tags are each generated independently, and they influence the selection of the part-of-speech tags to which they are connected. As before, we use a product-of-experts model to combine these cross-lingual cues with the standard HMM transition model.

This setup scales well. Crossing and many-to-many alignments may be used without creating cycles, as all cross-lingual information emanates from the hidden superlingual tags. Furthermore, the size of the model and its parameter space scale linearly with the number of languages. We now describe the role of the superlingual tags in more detail.

2.7.1 Propagating Cross-lingual Patterns with Superlingual Tags

Each superlingual tag specifies a set of distributions — one for each language's part-of-speech tagset. In order to learn repeated cross-lingual patterns, we need to constrain the number of values that the superlingual tags can take and thus the number of distributions they provide. For example, we might allow the superlingual tags to take on integer values from 1 to K, with each integer value indexing a separate set of tag distributions. Each set of distributions should correspond to a discovered cross-lingual pattern in the data. For example, one set of distributions might favor nouns in each language and another might favor verbs, though heterogenous distributions (e.g., favoring determiners in one language and prepositions in others) are also possible.

Rather than fixing the number of superlingual tag values to an arbitrary size K, we leave it unbounded, using a non-parametric Bayesian model. To encourage the desired multilingual clustering behavior, we use a Dirichlet process prior [36]. Under this prior, high posterior probability is obtained only when a small number of values are used repeatedly. The actual number of sampled values will thus be dictated by the data.

We draw an infinite sequence of distribution sets $\Psi_1, \Psi_2, \ldots$ from some base distribution G_0. Each Ψ_i is a set of distributions over tags, with one distribution per language, written $\psi_i^{(\ell)}$. To weight these sets of distributions, we draw an infinite sequence of mixture weights $\pi_1, \pi_2, \ldots$ from a stick-breaking process, which defines a distribution over the integers with most probability mass placed on some initial set of values. The pair of sequences $\pi_1, \pi_2, \ldots$ and $\Psi_1, \Psi_2, \ldots$ now define the distribution over superlingual tags and their associated distributions on parts-of-speech. Each superlingual tag $z \in \mathbb{N}$ is drawn with probability π_z, and is associated with the set of multinomials $\langle \psi_z^\ell, \psi_z^{\ell'}, \ldots \rangle$.

As in the merged node model, the distribution over aligned part-of-speech tags is governed by a product of experts. In this case, the incoming edges are from the superlingual tags (if any) and the predecessor tag. We combine these distributions via their normalized product. Assuming tag position i of language ℓ is connected to M superlingual tags, the part-of-speech tag y_i is drawn according to,

$$y_i \sim \frac{\phi_{y_{i-1}}(y_i) \prod_{m=1}^{M} \psi_{z_m}^{\ell}(y_i)}{Z}, \tag{2.6}$$

where $\phi_{y_{i-1}}$ indicates the transition distribution, z_m is the value of the m^{th} connected superlingual tag, and $\psi_{z_m}^{\ell}(y_i)$ indicates the tag distribution for language ℓ given by Ψ_{z_m}. The normalization Z is obtained by summing this product over all possible values of y_i.

This parameterization allows for a relatively simple parameter space. It also leads to a desirable property: for a tag to have high probability, *each* of the incoming distributions must allow it. That is, any expert can "veto" a potential tag by assigning it low probability, generally leading to consensus decisions.

We now formalize this description by giving the stochastic generative process for the observed data (raw parallel text and alignments), according to the multilingual model.

2.7.2 Generative Story

For n languages, we assume the existence of n tagsets $T^1, \ldots, T^n$ and vocabularies, $W^1, \ldots, W^n$, one for each language. Table 2.1 summarizes all relevant parameters. For clarity the generative process is described using only bigram transition dependencies, but our experiments use a trigram model, without any bigram interpolations.

1. **Transition and Emission Parameters**. For each language $\ell = 1, ..., n$ and for each tag $t \in T^\ell$, draw a *transition* distribution ϕ_t^ℓ over tags T_ℓ and an *emission* distribution θ_t^ℓ over words W^ℓ, all from symmetric Dirichlet priors of appropriate dimension.

2. **Superlingual Tag Parameters**. Draw an infinite sequence of sets of distributions over tags $\Psi_1, \Psi_2, \ldots$, where each Ψ_i is a set of n multinomials $\langle \psi_i^1, \psi_i^2, \ldots \psi_i^n \rangle$, one for each of n languages. Each multinomial ψ_i^ℓ is a distribution over the tagset T^ℓ, and is drawn from a symmetric Dirichlet prior; these priors together comprise the base distribution G_0, from which each Ψ_i is drawn.

   At the same time, draw an infinite sequence of mixture weights $\pi \sim GEM(\alpha)$, where $GEM(\alpha)$ indicates the stick-breaking distribution [103] with concentration parameter $\alpha = 1$. These parameters define a distribution over superlingual tags, or equivalently over the part-of-speech distributions that they index:

$$z \sim \textstyle\sum_k^\infty \pi_k \delta_{k=z} \tag{2.7}$$

$$\Psi \sim \textstyle\sum_k^\infty \pi_k \delta_{\Psi=\Psi_k} \tag{2.8}$$

   where $\delta_{\Psi=\Psi_k}$ is defined as one when $\Psi = \Psi_k$ and zero otherwise. From For-

Algorithm 2: Gibbs sampler for latent variable part-of-speech tagging model.

```
Input: An $n$-language corpus consisting of aligned sentence-tuples
       $(\mathbf{x}^1, \dots, \mathbf{x}^n)$, and corresponding word-level alignments $\mathbf{a}$
Output: 1000 samples of part-of-speech tags $(\mathbf{y}^1, \dots, \mathbf{y}^n)$ for each aligned
        sentence
Initialize part-of-speech tags;
for $r \leftarrow 1$ to 1000 do
    foreach sentence-tuple $(\mathbf{x}^1, \dots, \mathbf{x}^n)$ and word alignment $\mathbf{a}$ do
        foreach word $x_i^\ell \in (\mathbf{x}^1, \dots, \mathbf{x}^n)$ do
            Sample part-of-speech tag $y_i^\ell$                // Section 2.8.1
        foreach word alignment $a_i \in \mathbf{a}$ do
            Sample superlingual tag $z_i$                      // Section 2.8.2
```

mula 2.8, we can say that the set of multinomials Ψ is drawn from a Dirichlet process, conventionally written $DP(\alpha, G_0)$.

3. **Data**. For each multilingual parallel sentence:

   (a) Draw an alignment $\mathbf{a}$ from multilingual alignment distribution A_m. The alignment $\mathbf{a}$ specifies sets of aligned indices across languages; each such set may consist of indices in any subset of the languages.

   (b) For each set of indices in $\mathbf{a}$, draw a superlingual tag value z according to Formula 2.7.

   (c) For each language ℓ, for $i = 1, \dots$ (until end-tag reached):

      i. Draw a part-of-speech tag $y_i \in T^\ell$ according to Formula 2.6.

      ii. Draw a word $w_i \in W^\ell$ according to the emission distribution θ_{y_i}.

One important difference from the merged node model generative story is that the distribution over multilingual alignments A_m is unconstrained: we can generate crossing and many-to-one alignments as needed. To perform Bayesian inference under this model we again use Gibbs sampling, marginalizing parameters whenever possible.

2.8 Latent Variable Model: Inference

As in section 2.6, we employ a sampling-based inference procedure. Again, standard closed forms are used to analytically marginalize the emission parameters $\boldsymbol{\theta}$, and approximate marginalizations are applied to transition parameters $\boldsymbol{\phi}$, and superlingual tag distributions ψ_i^ℓ; similar techniques are used to marginalize the superlingual tag mixture weights $\boldsymbol{\pi}$. As before, these approximations would be exact if each of the parameters in the numerator of Formula 2.6 were solely responsible for other sampled tags.

We still must sample the part-of-speech tags $\mathbf{y}$ and superlingual tags $\mathbf{z}$. See algorithm 2 for an overview of the sampler. The remainder of the section describes the individual sampling equations.

2.8.1 Sampling Part-of-speech Tags

To sample the part-of-speech tag for language ℓ at position i we draw from:

$$\begin{aligned}&P(y_i^\ell|\mathbf{y}_{-(\ell,i)}, \mathbf{x}, \mathbf{a}, \mathbf{z}) \;\propto \\ &\quad P(x_i^\ell|\mathbf{x}_{-i}^\ell, \mathbf{y}^\ell)\; P(y_{i+1}^\ell|y_i^\ell, \mathbf{y}_{-(\ell,i)}, \mathbf{a}, \mathbf{z})\; P(y_i^\ell|\mathbf{y}_{-(\ell,i)}, \mathbf{a}, \mathbf{z}),\end{aligned} \tag{2.9}$$

where $\mathbf{y}_{-(\ell,i)}$ refers to all tags except y_i^ℓ. The first factor handles the emissions, and the latter two factors are the generative probabilities of (i) the current tag given the previous tag and superlingual tags, and (ii) the next tag given the current tag and superlingual tags. These two quantities are similar to equation 2.6, except here we integrate over the transition parameter $\phi_{y_{i-1}}$ and the superlingual tag parameters ω_z^ℓ. We end up with a product of integrals, each of which we compute in closed form.

Terms involving the transition distributions $\boldsymbol{\phi}$ and the emission distributions $\boldsymbol{\theta}$ are identical to the bilingual case, as described in Section 2.6. The closed form for integrating over the parameter of a superlingual tag with value z is given by:

$$\int \psi_z^\ell(y_i)\ P(\psi_z^\ell \mid \psi_0^\ell)\ d\psi_z^\ell = \frac{n(z, y_i, \ell) + \psi_0^\ell}{n(z, \ell) + T^\ell\ \psi_0^\ell}$$

where $n(z, y_i, \ell)$ is the number of times that tag y_i is observed together with superlingual tag z in language ℓ, $n(z, \ell)$ is the total number of times that superlingual tag z appears with an edge into language ℓ, and ψ_0^ℓ is a symmetric Dirichlet prior over tags for language ℓ.

2.8.2 Sampling Superlingual Tags

For each set of aligned words in the observed alignment $\mathbf{a}$ we need to sample a superlingual tag z. Recall that z is an index into an infinite sequence

$$\langle \psi_1^{\ell_1}, \ldots, \psi_1^{\ell_n} \rangle, \langle \psi_2^{\ell_1}, \ldots, \psi_2^{\ell_n} \rangle, \ldots,$$

where each ψ_z^ℓ is a distribution over the tagset T^ℓ. The generative distribution over z is given by Formula 2.7. In our sampling scheme, however, we integrate over all possible settings of the mixture weights $\boldsymbol{\pi}$ using the standard Chinese Restaurant Process closed form [34]:

$$P\big(z_i \big| \mathbf{z}_{-i}, \mathbf{y}\big) \ \propto\ \prod_\ell P\big(y_i^\ell \big| z_i, \mathbf{z}_{-i}, \mathbf{y}_{-(\ell,i)}\big) \cdot \begin{cases} \frac{1}{k+\alpha} n(z_i) & \text{if } z_i \in \mathbf{z}_{-i} \\ \frac{\alpha}{k+\alpha} & \text{otherwise} \end{cases} \tag{2.10}$$

The first group of factors is the product of closed form probabilities for all tags connected to the superlingual tag, conditioned on z_i. Each of these factors is calculated in the same manner as equation 2.9 above. The final factor is the standard Chinese Restaurant Process closed form for posterior sampling from a Dirichlet process prior. In this factor, k is the total number of sampled superlingual tags, $n(z_i)$ is the total number of times the value z_i occurs in the sampled superlingual tags, and α is the Dirichlet process concentration parameter (see Step 2 in Section 2.7.2).

2.9 Implementation

This section describes implementation details that are necessary to reproduce our experiments. We present details for the merged node and latent variable models, as well as our monolingual baseline.

2.9.1 Initialization

An initialization phase is required to generate initial settings for the word tags and hyperparameters, and for the superlingual tags in the latent variable model. The initialization is as follows:

- **Monolingual Model**
  - **Tags:** Random, with uniform probability among tag dictionary entries for the emitted word.
  - **Hyperparameters θ_0, ϕ_0:** Initialized to 1.0
- **Merged Node Model**
  - **Tags**: Random, with uniform probability among tag dictionary entries for the emitted word. For joined tag nodes, each slot is selected from the tag dictionary of the emitted word in the appropriate language.
  - **Hyperparameters θ_0, ϕ_0, ω_0:** Initialized to 1.0
- **Latent Variable Model**
  - **Tags:** Set to the final estimate from the monolingual model.
  - **Superlingual Tags:** Initially a set of 14 superlingual tag values is assumed — each value corresponds to one part-of-speech tag. Each alignment is assigned one of these 14 values based on the most common initial part-of-speech tag of the words in the alignment.
  - **Hyperparameters θ_0^ℓ, ϕ_0^ℓ:** Initialized to 1.0

- **Base Distribution G_0^ℓ:** Set to a symmetric Dirichlet distribution with parameter value fixed to 1.0
- **Concentration Parameter α:** Set to 1.0 and remains fixed throughout.

2.9.2 Hyperparameter Estimation

Both models have symmetric Dirichlet priors θ_0 and ϕ_0, for the emission and transition distributions respectively. The merged node model also has symmetric Dirichlet prior ω_0 on the coupling parameter. We re-estimate these priors during inference, based on non-informative hyperpriors.

Hyperparameter re-estimation applies the Metropolis-Hastings algorithm after each full epoch of sampling the tags. In addition, we run an initial 200 iterations to speed convergence. Metropolis-Hastings is a sampling technique that draws a new value u from a proposal distribution, and makes a stochastic decision about whether to accept the new sample [39]. This decision is based on the proposal distribution and on the joint probability of u with the observed and sampled variables $\mathbf{x}^\ell$ and $\mathbf{y}^\ell$.

We assume an improper prior $P(u)$ that assigns uniform probability mass over the positive reals, and use a Gaussian proposal distribution with the mean set to the previous value of the parameter and variance set to one-tenth of the mean.[5] For non-pathological proposal distributions, the Metropolis-Hastings algorithm is guaranteed to converge in the limit to a stationary Markov chain centered on the desired joint distribution. We observe an acceptance rate of approximately 1/6, which is in line with standard recommendations for rapid convergence [39].

2.9.3 Final Parameter Estimates

The ultimate goal of training is to learn models that can be applied to unaligned monolingual data. Thus, we need to construct estimates for the transition and

[5] This proposal is identical to the parameter re-estimation applied for emission and transition priors by [43].

emission parameters ϕ and θ. Our sampling procedure focuses on the tags $\mathbf{y}$. We construct maximum *a posteriori* estimates $\hat{y}$, indicating the most likely tag sequences for the aligned training corpus. The predicted tags $\hat{y}$ are then combined with priors ϕ_0 and θ_0 to construct maximum *a posteriori* estimates of the transition and emission parameters. These learned parameters are then applied to the monolingual test data to find the highest probability tag sequences using the Viterbi algorithm.

For the monolingual and merged node models, we perform 200 iterations of sampling, and select the modal tag settings in each slot. Further sampling was not found to produce different results. For the latent variable model, we perform 1000 iterations of sampling, and select the modal tag values from the last 100 samples.

2.10 Experimental Setup

We perform a series of empirical evaluations to quantify the contribution of bilingual and multilingual information for unsupervised part-of-speech tagging. Our first evaluation follows the standard procedures established for unsupervised part-of-speech tagging: given a tag dictionary (i.e., a set of possible tags for each word type), the model selects the appropriate tag for each token occurring in a text. We also evaluate tagger performance when the available dictionaries are incomplete [106, 43]. In all scenarios, the model is trained using only untagged text.

In this section, we first describe the parallel data and part-of-speech annotations used for system evaluation. Next we describe a monolingual baseline and the inference procedure used for testing.

2.10.1 Data

As a source of parallel data, we use Orwell's novel "Nineteen Eighty Four" in the original English as well as its translation to seven languages — Bulgarian, Czech, Estonian, Hungarian, Slovene, Serbian and Romanian.[6] Each translation was produced by a different translator and published in print separately by different publishers.

This dataset has representatives from four language families — Slavic, Romance, Ugric and Germanic. This data is distributed as part of the publicly available Multext-East corpus, Version 3 [33]. The corpus provides detailed morphological annotation at the token level, including part-of-speech tags. In addition, a lexicon for each language is provided.

The corpus consists of 118,426 English words in 6,736 sentences (see table 2.3). Of these sentences, the first 75% are used for training, taking advantage of the multilingual alignments. The remaining 25% are used for evaluation. In the evaluation, only monolingual information is made available to the model, to simulate performance on non-parallel data.

2.10.2 Alignments

In our experiments we use sentence-level alignments provided in the Multext-East corpus. Word-level alignments are computed for each language pair using GIZA++ [90]. The procedures for handling these alignments are different for the merged node and latent variable models.

Merged Node Model

We obtain 28 parallel bilingual corpora by considering all pairings of the eight languages. To generate one-to-one alignments at the word level, we intersect the one-to-many alignments going in each direction. This process results in alignment of about half the tokens in each bilingual parallel corpus. We further automatically remove crossing alignment edges, as these would induce cycles in the graphical model. We employ a simple heuristic: crossing alignment edges are removed based

[6] In our initial publication [111], we used a subset of this data, only including sentences that have one-to-one alignments between all four languages considered in that paper. The current set-up makes use of all the sentences available in the corpus.

			Percentage Aligned							
	Sentences	Words	BG	CS	EN	ET	HU	RO	SL	SR
BG	6681	101175	-	41.7	50.5	33.5	31.3	41.5	45.4	45.9
CS	6750	102834	41.0	-	41.9	39.1	30.7	31.7	56.2	48.4
EN	6736	118426	43.2	36.4	-	34.4	32.9	42.5	44.6	40.9
ET	6477	94900	35.7	42.4	42.9	-	33.8	29.2	44.8	39.7
HU	6767	98428	32.2	32.0	39.6	32.6	-	26.9	34.6	30.3
RO	6519	118330	35.5	27.5	42.5	23.4	22.4	-	30.8	32.1
SL	6688	116908	39.3	49.4	45.2	36.4	29.1	31.2	-	51.2
SR	6676	112131	41.4	44.4	43.2	33.6	26.6	33.9	53.4	-

Table 2.2: Percentage of the words in the row language that have alignments when paired with the column language. See table 2.3 for language name abbreviations.

on the order in which they appear from left to right; this step eliminates on average 3.62% of the edges. Table 2.2 shows the number of aligned words for each language pair after removing crossing edges. More detailed statistics about the total number of alignments are provided in appendix B.

Latent Variable Model

As in the previous setting, we run GIZA++ on all 28 pairings of the 8 languages, taking the intersection of alignments in each direction. Since we want each latent superlingual variable to span as many languages as possible, we aggregate pairwise lexical alignments into larger sets of densely aligned words and assign a single latent superlingual variable to each such set. Specifically, for each word token, we consider the set of the word itself and all word tokens to which it is aligned. If pairwise alignments occur between two-thirds of all token pairs in this set, then it is considered densely connected and is admitted as an alignment set. Otherwise, increasingly smaller subsets are considered until one that is densely connected is found. This procedure is repeated for all word tokens in the corpus that have at least one alignment. Finally, the alignment sets are pruned by removing those which are subsets of larger alignment sets. Each of the remaining sets is considered the site of a latent superlingual variable.

This process can be illustrated by an example. The sentence "I know you, the

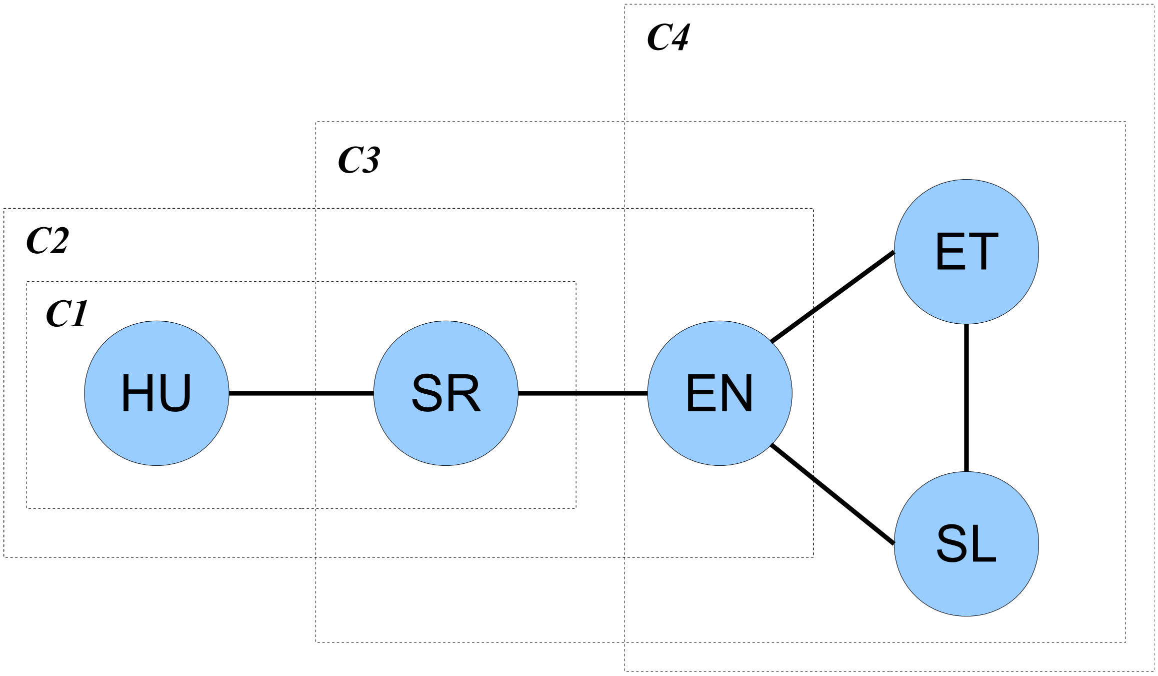


Figure 2-2: An example of a multilingual alignment configuration. Nodes correspond to words tokens, and are labeled by their language. Edges indicate pairwise alignments produced by GIZA++. Boxes indicate alignment sets, though the set ***C1*** is subsumed by ***C2*** and eventually discarded, as described in the text.

eyes seemed to say, I see through you," appears in the original English version of the corpus. The English word token *seemed* is aligned to word tokens in Serbian (*činilo*), Estonian (*näis*), and Slovenian (*zdelo*). The Estonian and Slovenian tokens are aligned to each other. Finally, the Serbian token is aligned to a Hungarian word token (*mintha*), which is itself not aligned to any other tokens. This configuration is shown in Figure 2-2, with the nodes labeled by the two-letter language abbreviations.

We now construct alignment sets for these words.

- For the Hungarian word, there is only one other aligned word, in Serbian, so the alignment set consists only of this pair (***C1*** in the figure).

- The Serbian word has aligned partners in both Hungarian and English; overall this set has two pairwise alignments out of a possible three, as the English and Hungarian words are not aligned. Still, since 2/3 of the possible alignments are present, an alignment set (***C2***) is formed. ***C1*** is subsumed by ***C2***, so it is eliminated.

- The English word is aligned to tokens in Serbian, Estonian, and Slovenian; four of six possible links are present, so an alignment set (***C3***) is formed. Note that if the Estonian and Slovenian words were not aligned to each other then we would have only three of six links, so the set would not be densely connected by our definition; we would then remove a member of the alignment set.
- The Estonian token is aligned to words in Slovenian and English; all three pairwise alignments are present, so an alignment set (***C4***) is formed. An identical alignment set is formed by starting with the Slovenian word, but only one superlingual tag is created.

Thus, for these five word tokens, a total of three overlapping alignment sets are created. Over the entire corpus, this process results in 284,581 alignment sets, covering 76% of all word tokens. Of these tokens, 61% occur in exactly one alignment set, 29% occur in exactly two alignment sets, and the remaining 10% occur in three or more alignment sets. Of all alignment sets, 32% include words in just two languages, 26% include words in exactly three languages, and the remaining 42% include words in four or more languages. The sets remain fixed during sampling and are treated by the model as observed data.

2.10.3 Tagset

The Multext-East corpus is manually annotated with detailed morphosyntactic information. In our experiments, we focus on the main syntactic category encoded as the first letter of the provided labels. The annotation distinguishes between 14 parts-of-speech, of which 11 are common for all languages in our experiments. Appendix A lists the tag repository for each of the eight languages.

	Number	Tags per token when lexicon contains ...				Trigram
	of Tokens	all words	count > 5	count > 10	top 100 words	Entropy
BG	101175	1.39	4.61	5.48	7.33	1.63
CS	102834	1.35	5.27	6.37	8.24	1.64
EN	118426	1.49	3.11	3.81	6.21	1.51
ET	94900	1.36	4.91	5.82	7.34	1.61
HU	98428	1.29	5.42	6.41	7.85	1.62
RO	118330	1.55	4.49	5.53	8.54	1.73
SL	116908	1.33	4.59	5.49	7.23	1.64
SR	112131	1.38	4.76	5.73	7.61	1.73

Table 2.3: Corpus size and tag/token ratio for each language in the set. The last column shows the trigram entropy for each language based on the annotations provided with the corpus. BG = Bulgarian, CS = Czech, EN = English, ET = Estonian, HU = Hungarian, RO = Romanian, SL = Slovene, SR = Serbian.

In our first experiment, we assume that a complete tag lexicon is available, so that the set of possible parts-of-speech for each word is known in advance. We use the tag dictionaries provided in the Multext-East corpus. The average number of possible tags per token is 1.39. We also experimented with incomplete tag dictionaries, where entries are only available for words appearing more than five or ten times in the corpus. For other words, the entire tagset of 14 tags is considered. In these two scenarios, the average per-token tag ambiguity is 4.65 and 5.58, respectively. Finally we also considered the case when lexicon entries are available for only the 100 most frequent words. In this case the average tags per token ambiguity is 7.54. Table 2.3 shows the specific tag/token ratio for each language for all scenarios.

In the Multext-East corpus, punctuation marks are not annotated with part-of-speech tags. We expand the tag repository by defining a separate tag for all punctuation marks. This allows the model to make use of any transition or coupling patterns involving punctuation marks. However, we do not consider punctuation tokens when computing model accuracy.

2.10.4 Monolingual Comparisons

As our monolingual baseline we use the unsupervised Bayesian hidden Markov model (HMM) of Goldwater and Griffiths [43]. This model, which they call BHMM1, modifies the standard HMM by adding priors and by performing Bayesian inference. Its performance is on par with state-of-the-art unsupervised models. The Bayesian HMM is a particularly informative baseline because our model reduces to this baseline when there are no alignments in the data. This implies that any performance gain over the baseline can only be attributed to the multilingual aspect of our model. We used our own implementation after verifying that its performance on the Penn Treebank corpus was identical to that reported by Goldwater and Griffiths.

To provide an additional point of comparison, we use a supervised hidden Markov model trained using the annotated corpus. We apply the standard maximum-likelihood estimation and perform inference using Viterbi decoding with pseudo-count smoothing for unknown words [95]. In appendix C we also report supervised results using the Stanford Tagger [117], version 1.6[7]. Although the results are slightly lower than our own supervised HMM implementation, we note that this system is not directly comparable to our set-up, as it does not allow the use of a tag dictionary to constrain part-of-speech selections.

2.10.5 Test Set Inference

We use the same procedure to apply all the models (the monolingual model, the bilingual merged node model, and the latent variable model) to test data. After training, trigram transition and word emission probabilities are computed, using the counts of tags assigned in the final training iteration. Similarly, the final sampled values of the hyperparameters are selected as smoothing parameters. We then apply Viterbi decoding to identify the highest probability tag sequences for each monolingual test set. We report results for multilingual and monolingual experiments averaged over five runs and for bilingual experiments averaged over three runs. The average standard-deviation of accuracy over multiple runs is less than 0.25 except when the lexicon is limited to the 100 most frequent words. In that case the standard deviation is 1.11 for monolingual model, 0.85 for merged node model

[7] `http://nlp.stanford.edu/software/tagger.shtml`

and 1.40 for latent variable model.

2.11 Results

In this section, we first report the performance for the two models on the full and reduced lexicon cases. Next, we report results for a semi-supervised experiment, where a subset of the languages have annotated text at training time. Finally, we investigate the sensitivity of both models to hyperparameter values and provide run time statistics for the latent variable model for increasing numbers of languages.

2.11.1 Full Lexicon Experiments

Our experiments show that both the merged node and latent variable models substantially improve tagging accuracy. Since the merged node model is restricted to pairs of languages, we provide average results over all possible pairings. In addition, we also consider two methods for combining predictions from multiple bilingual pairings: one using a voting scheme and the other employing an oracle to select the best pairings (see below for additional details).

As shown in Line 4 of table 2.4, the merged node model achieves, on average, 93.2% accuracy, a two percentage point improvement over the monolingual baseline.[8] The latent variable model — trained once on all eight languages — achieves 95% accuracy, nearly two percentage points higher than the bilingual merged node model. These two results correspond to error reductions of 23% and 43% respectively, and reduce the gap between unsupervised and supervised performance by over 30% and 60%.

As mentioned above, we also employ a voting scheme to combine information from multiple languages using the merged node model. Under this scheme, we

[8] The accuracy of the monolingual English tagger is relatively high compared to the 87% reported by [43] on the WSJ corpus. We attribute this discrepancy to the differences in tag inventory used in our data-set. For example, when *Particles* and *Prepositions* are merged in the WSJ corpus (as they happen to be in our tag inventory and corpus), the performance of Goldwater's model on WSJ is similar to what we report here.

	Avg	BG	CS	EN	ET	HU	RO	SL	SR
1. Random	83.3	82.5	86.9	80.7	84.0	85.7	78.2	84.5	83.5
2. Monolingual	91.2	88.7	93.9	95.8	92.7	95.3	91.1	87.4	84.5
3. MergedNode: *average*	**93.2**	91.3	96.9	95.9	93.3	96.7	91.9	89.3	90.2
4. LatentVariable	**95.0**	92.6	98.2	95.0	94.6	96.7	95.1	95.8	92.3
5. Supervised	97.3	96.8	98.6	97.2	97.0	97.8	97.7	97.0	96.6
6. MergedNode: *voting*	93.0	91.6	97.4	96.1	94.3	96.8	91.6	87.9	88.2
7. MergedNode: *best pair*	95.4	94.7	97.8	96.1	94.2	96.9	94.1	94.8	94.5

Table 2.4: Tagging accuracy with complete tag dictionaries. The first column reports average results across all languages (see table 2.3 for language name abbreviations). The latent variable model is trained using all eight languages, whereas the merged node models are trained on language pairs. In the latter case, results are given by averaging over all pairings (line 3), by having all bilingual models vote on each tag prediction (line 6), and by having an oracle select the best pairing for each target language (line 7). All differences between LatentVariable, MergedNode: *voting*, and Monolingual (lines 2, 4, and 6) are statistically significant at $p < 0.05$ according to a sign test. See table 2.3 for language name abbreviations.

train bilingual merged node models for each language pair. Then, when making tag predictions for a particular language — e.g., Romanian — we consider the preferences of each bilingual model trained with Romanian and a second language. The tag preferred by a plurality of models is selected. The results for this method are shown in line 6 of table 2.4, and do not differ significantly from the average bilingual performance. Thus, this simple method of combining information from multiple language does not measure up to the joint multilingual model performance.

We use the sign test to assess whether there are statistically significant differences in the accuracy of the tag predictions made by the monolingual baseline (line 2 of table 2.4), the latent variable model (line 4), and the voting-based merged node model (line 6). All differences in these rows are found to be statistically significant at $p < 0.05$. Note that we cannot use the sign test to compare the average performance of the bilingual model (line 3), since this result is an aggregate over accuracies for every language pair.

2.11.2 Reduced Lexicon Experiments

In realistic application scenarios, we may not have a tag dictionary with coverage across the entire lexicon. We consider three reduced lexicons: removing all words with counts of five or less; removing all words with counts of ten or less; and keeping only the top 100 most frequent words. Words that are removed from the lexicon can take any tag, increasing the overall difficulty of the task. These results are shown in table 2.5 and graphically summarized in Figure 2-3. In all cases, the monolingual model is less robust to reduction in lexicon coverage than the multilingual models. In the case of the 100 word lexicon, the latent variable model achieves accuracy of 57.9%, compared to 53.8% for the monolingual baseline. The merged node model, on the other hand, achieves a slightly higher average performance of 59.5%. In the two other scenarios, the latent variable model trained on all eight languages outperforms the bilingual merged node model, even when an oracle selects the best bilingual pairing for each target language. For example, using the lexicon with words that appear greater than five times, the monolingual baseline achieves 74.7% accuracy, the merged node model using the best possible pairings achieves 81.7% accuracy, and the full latent variable model achieves an accuracy of 82.8%.

Next we consider the performance of the bilingual merged node model when the lexicon is reduced for only one of the two languages. This condition may occur when dealing with two languages with asymmetric resources, in terms of unannotated text. As shown in table 2.6, the merged models on average scores 5.7 points higher than the monolingual model when both tag dictionaries are reduced, but 14.3 points higher when the partner language has a full tag dictionary. This suggests that the bilingual models effectively transfer the additional lexical information available for the resource-rich language to the resource-poor language, yielding substantial performance improvements.

Perhaps the most surprising result is that the resource-rich language gains as much on average from pairing with the resource-poor partner language as it would have gained from pairing with a language with a full lexicon. In both cases, an

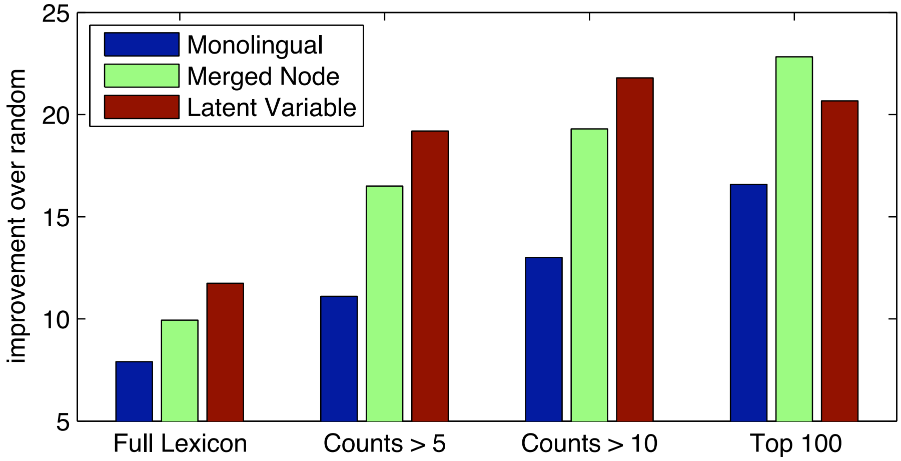


Figure 2-3: Summary of model performance in full and reduced lexicon conditions. Improvement over the random baseline is indicated for the monolingual baseline, the merged node model (average performance over all possible bilingual pairings), and the latent variable model (trained on all eight languages). "Counts $> x$" indicates that only words with counts greater than x were kept in the lexicon; "Top 100" keeps only the 100 most common words.

average accuracy of 93.2% is achieved, compared to the 91.1% monolingual baseline.

2.11.3 Indirect Supervision

Although the main focus of this thesis is unsupervised learning, we also provide some results indicating that multilingual learning can be applied to scenarios with varying amounts of annotated data. These scenarios are in fact quite realistic, as previously trained and highly accurate taggers will usually be available for at least some of the languages in a parallel corpus. We apply our latent variable model to these scenarios by simply treating the tags of annotated data (in any subset of languages) as fixed and observed throughout the sampling procedure. From a strictly probabilistic perspective this is the correct approach. However, we note that, in practice, heuristics and objective functions which place greater emphasis on the supervised portion of the data may yield better results. We do not explore that possibility here.

Table 2.7 gives results for two scenarios of indirect supervision: where only one

		Avg	BG	CS	EN	ET	HU	RO	SL	SR
Counts $>$ 5	Random	63.6	62.9	62	71.8	61.6	61.3	62.8	64.8	61.8
	Monolingual	74.8	73.5	72.2	87.3	72.5	73.5	77.1	75.7	66.3
	MergedNode: *average*	80.1	80.2	79.0	90.4	76.5	77.3	82.7	78.7	75.9
	LatentVariable	**82.8**	81.3	**83.0**	88.1	**80.6**	**80.8**	**86.1**	**83.6**	78.8
	MergedNode: *voting*	80.4	80.4	78.5	90.7	76.4	76.8	84.0	79.7	76.4
	MergedNode: *best pair*	81.7	**82.7**	79.7	**90.7**	77.5	78	84.4	80.9	**79.4**
Counts $>$ 10	Random	57.9	57.5	54.7	68.3	56	55.1	57.2	59.2	55.5
	Monolingual	70.9	71.9	66.7	84.4	68.3	69.0	73.0	70.4	63.7
	MergedNode: *average*	77.2	77.8	75.3	88.8	72.9	73.8	80.5	76.1	72.4
	LatentVariable	**79.7**	78.8†	**79.4**	86.1	**77.9**	**76.4**	**83.1**	**80.0**	75.9
	MergedNode: *voting*	77.5	78.4†	75.3	89.2	73.1	73.3	81.7	76.1	73.1
	MergedNode: *best pair*	79.0	**80.2**	76.7	**89.4**	74.9	75.2	82.1	77.6	**76.1**
Top 100	Random	37.3	36.7	32.1	48.9	36.6	36.4	33.7	39.8	33.8
	Monolingual	53.8	60.9‡	44.1	69.0	54.8*	56.8	51.4	49.4	44.0
	MergedNode: *average*	**59.6**	60.1	52.5	73.5	59.5	59.4	61.4	56.6	53.4
	LatentVariable	57.9	**65.5**	49.3	71.6	54.3*	51.0	57.5	53.9	**60.4**
	MergedNode: *voting*	62.4	61.5‡	55.4	74.8	62.2	60.9	64.3	62.3	57.5
	MergedNode: *best pair*	63.6	64.7	**55.3**	**77.4**	**61.5**	**60.2**	**69.3**	**63.1**	56.9

Table 2.5: Tagging accuracy in reduced lexicon conditions. "Counts $> x$" indicates that only words with counts greater than x were kept in the lexicon; "Top 100" keeps only the 100 most common words. The latent variable model is trained using all eight languages, whereas the merged node models are trained on language pairs. In the latter case, results are given by averaging over all pairings, by having all bilingual models vote on each tag prediction, and by having an oracle select the best pairing for each target language. Other than the three pairs of results marked with †, ‡, and $*$, all differences between "monolingual", "LatentVariable", and "MergedNode: *voting*" are statistically significant at $p < 0.05$ according to a sign test. See table 2.3 for language name abbreviations.

of the eight languages has annotated data, and where all *but* one of the languages has annotated data. In both cases, the unsupervised languages are provided with a 100 word lexicon, and all eight languages are trained together. When only one of the eight languages is supervised, the results vary depending on the choice of supervised language. When one of Bulgarian, Hungarian, or Romanian is supervised, no improvement is seen, on average, for the other seven languages. However, when Slovene is supervised, the improvement seen for the other languages is fairly substantial, with average accuracy rising to 64.8%, from 57.9% for the unsupervised latent variable model and 53.8% for the monolingual baseline. Perhaps unsurpris-

	Monolingual		Bilingual (Merged Node)			
	Reduced	Full	Both	Reduced	Unreduced	Both
			reduced	language	language	full
BG	60.9	88.7	60.1	71.3	91.6	91.3
CS	44.1	93.9	52.5	66.7	97.1	96.9
EN	69.0	95.8	73.5	82.4	95.8	95.9
ET	54.8	92.7	59.5	65.6	93.3	93.3
HU	56.8	95.3	59.4	63.0	96.7	96.7
RO	51.4	91.1	61.4	69.3	91.5	91.9
SL	49.4	87.4	56.6	63.3	89.1	89.3
SR	44.0	84.5	53.4	63.6	90.3	90.2
Avg.	53.8	91.2	59.5	68.1	93.2	93.2

Table 2.6: Various scenarios for reducing the tag dictionary to the 100 most frequent terms. See table 2.3 for language name abbreviations.

ingly, the results are more impressive when all *but* one of the languages is supervised. In this case, the average accuracy of the lone unsupervised language rises to 74.4%. Taken together, these results indicate that any mixture of supervised resources may be added to the mix in a very simple and straightforward way, often yielding substantial improvements for the other languages.

2.11.4 Hyperparameter Sensitivity and Runtime Statistics

Both models employ hyperparameters for the emission and transition distribution priors (θ_0 and ϕ_0 respectively) and the merged node model employs an additional hyperparameter for the coupling distribution prior (ω_0). These hyperparameters are all updated throughout the inference procedure. The latent variable model uses two additional hyperparameters that remained fixed: the concentration parameter of the Dirichlet process (α) and the parameter of the base distribution for superlingual tags (ψ_0). For the experiments described above we used the initialization values listed in Section 2.9.1. Here we investigate the sensitivity of the models to different initializations of θ_0, ϕ_0, and ω_0, and to different fixed values of α and ψ_0. Tables 2.8 and 2.9 show the results obtained for the merged node and latent variable models, respectively, using a full lexicon. We observe that across a wide range of values, both models yield very similar results. In addition, we note that the final sampled

		supervised language(s)...									
		BG	CS	EN	ET	HU	RO	SL	SR	All others	None
accuracy for...	BG		69.1	68.0	65.9	60.4	67.1	73.9	69.6	76.2	65.5
	CS	50.8		52.2	50.2	51.2	51.0	56.6	53.1	76.6	49.3
	EN	62.6	70.5		68.1	61.8	61.9	80.6	69.5	82.8	71.6
	ET	57.2	58.0	57.7		56.1	56.4	59.8	57.1	72.5	54.3
	HU	50.3	50.0	53.1	51.4		51.1	49.8	50.0	62.3	51.0
	RO	62.8	61.6	61.3	57.8	58.5		62.9	59.2	74.9	57.5
	SL	55.0	56.8	55.6	53.2	54.4	54.7		56.2	77.7	53.9
	SR	64.9	65.9	64.1	63.5	61.6	63.4	69.9		72.5	60.4
	Avg	57.7	61.7	58.9	58.6	57.7	57.9	64.8	59.2	74.4	57.9

Table 2.7: Performance of the latent variable model when some of the eight languages have supervised annotations and the others have only the most frequent 100 words lexicon. The first eight columns report results when only one of the eight languages is supervised. The penultimate column reports results when all but one of the languages are supervised. The final column reports results when *no* supervision is available (repeated from table 2.5 for convenience). See table 2.3 for language name abbreviations..

hyperparameter values for transition and emission distributions always fall below one, indicating that sparse priors are preferred.

As mentioned in Section 2.7 one of the key theoretical benefits of the latent variable approach is that the size of the model and its parameter space scale linearly with the number of languages. Here we provide empirical confirmation by running the latent variable model on all possible subsets of the eight languages, recording the time elapsed for each run[9]. Figure 2-4 shows the average running time as the number of languages is increased (averaged over all subsets of each size). We see that the model running time indeed scales linearly as languages are added, and that the per-language running time increases very slowly: when all eight languages are included, the time taken is roughly double that for eight monolingual models run serially. Both of our models scale well with tagset size and the number of examples. The time dependence on the former is cubic, as we use trigram models and employ Viterbi decoding to find optimal sequences at test-time. During the training time, however, the time scales linearly with the tagset size for the latent

[9] All experiments were single-threaded and run using an Intel Xeon 3.0 GHz processor

MergedNode: *h*yperparameter initializations							
ϕ_0	1.0	**0.1**	**0.01**	1.0	1.0	1.0	1.0
θ_0	1.0	1.0	1.0	**0.1**	**0.01**	1.0	1.0
ω_0	1.0	1.0	1.0	1.0	1.0	**0.1**	**0.01**
BG	91.3	91.3	91.3	91.3	91.2	91.1	91.3
CS	96.9	97.0	97.0	96.9	96.8	96.5	97.1
EN	95.9	95.9	95.9	95.9	95.9	95.9	95.9
ET	93.3	93.4	93.3	93.4	93.2	93.4	93.2
HU	96.7	96.7	96.7	96.7	96.7	96.7	96.8
RO	91.9	91.8	91.8	91.9	91.8	91.8	91.8
SL	89.3	89.3	89.3	89.3	89.4	89.3	89.3
SR	90.2	90.2	90.2	90.2	90.2	90.2	90.2
Avg	93.2	93.2	93.2	93.2	93.2	93.1	93.2

Table 2.8: Results for different initializations of the hyperparameters of the merged node model. ϕ_0, θ_0 and ω_0 are the hyperparameters for the transition, emission and coupling multinomials respectively. The results for each language are averaged over all possible pairings with the other languages. See table 2.3 for language name abbreviations.

variable model and quadratically for the merged node model. This is due to the use of Gibbs sampling that isolates the individual sampling decision on tags (for the latent variable model) and tag-pairs (for the merged node model). The dependence on the number of training examples is also linear for the same reason.

2.12 Analysis

In this section we provide further analysis of: (i) factors that influence the effectiveness of language pairings in bilingual models, (ii) the incremental value of adding more languages in the latent variable model, (iii) the gains of multilingual modeling, (iv) the superlingual tags and their corresponding cross-lingual patterns as learned by the latent variable model, and (v) whether multilingual data is more helpful than additional monolingual data.

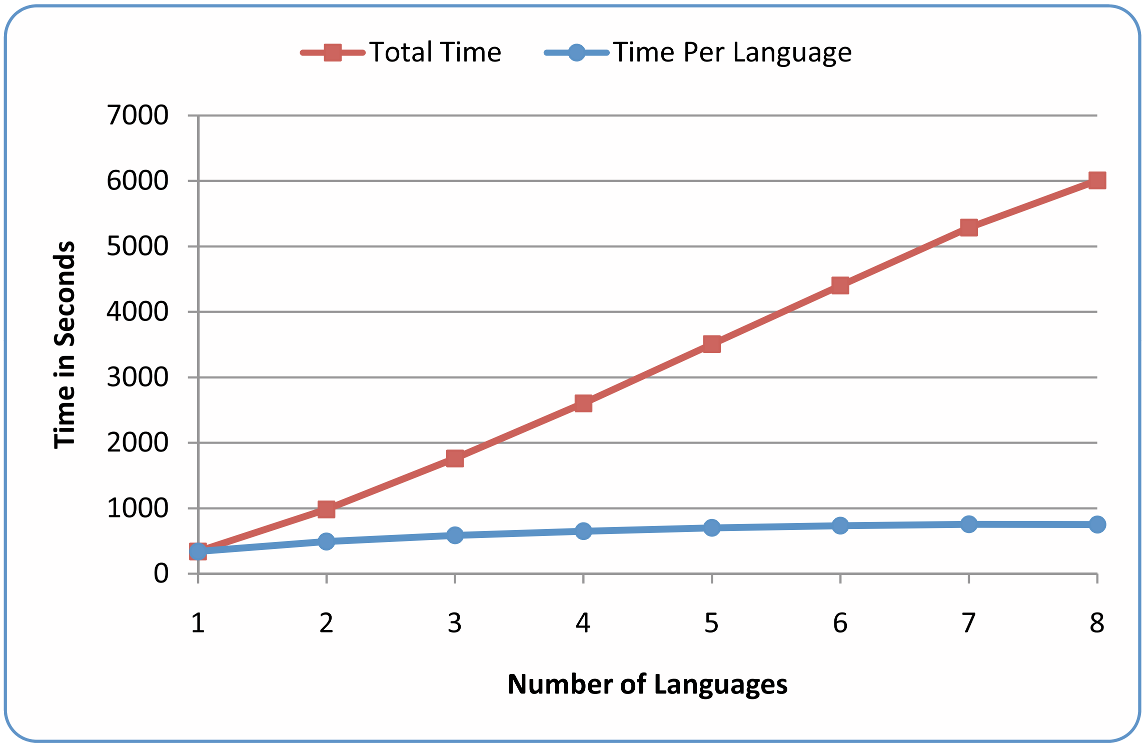


Figure 2-4: Average running time for 1000 iterations of the latent variable model. Results are averaged over all possible language subsets of each size. The top line shows the average running time for the entire subset, and the bottom line shows the running time divided by the number of languages.

LatentVariable: *h*yperparameter initializations & settings										
α	1.0	**0.1**	**10**	**100**	1.0	1.0	1.0	1.0	1.0	1.0
ψ_0	1.0	1.0	1.0	1.0	**0.1**	**0.01**	1.0	1.0	1.0	1.0
ϕ_0	1.0	1.0	1.0	1.0	1.0	1.0	**0.1**	**0.01**	1.0	1.0
θ_0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	**0.1**	**0.01**
BG	92.6	92.6	92.6	92.6	92.6	92.7	92.6	92.6	92.6	92.6
CS	98.2	98.1	98.2	98.2	98.1	98.1	98.2	98.1	98.2	98.1
EN	95.0	95.0	94.9	94.8	95.1	95.2	95.0	94.9	94.9	95.0
ET	94.6	95.0	95.0	94.9	94.2	94.8	95.0	94.9	94.9	94.5
HU	96.7	96.7	96.7	96.7	96.7	96.6	96.7	96.7	96.7	96.7
RO	95.1	95.0	95.1	95.1	95.2	95.1	95.0	94.9	95.1	95.0
SL	95.8	95.8	95.8	95.8	95.8	95.8	95.8	95.8	95.8	95.8
SR	92.3	92.3	92.3	92.3	92.4	92.4	92.3	92.3	92.3	92.3
Avg	95.0	95.1	95.1	95.0	95.0	95.1	95.1	95.0	95.1	95.0

Table 2.9: Results for different initializations and settings of hyperparameters of the latent variable model. ϕ_0 and θ_0 are the hyperparameters for the transition and emission multinomials respectively and are updated throughout inference. α and ψ_0 are the concentration parameter and base distribution parameter, respectively, for the Dirichlet process, and remain fixed. See table 2.3 for language name abbreviations.

2.12.1 Predicting Effective Language Pairings

We first analyze the cross-lingual variation in performance for different bilingual language pairings. As shown in table 2.10, the performance of the merged node model for each target language varies substantially across pairings. In addition, the identity of the optimally helpful language pairing also depends heavily on the target language in question. For instance, Slovene, achieves a large improvement when paired with Serbian (+7.4), a closely related Slavic language, but only a minor improvement when coupled with English (+1.8). On the other hand, for Bulgarian, the best performance is achieved when coupling with English (+6) rather than with closely related Slavic languages (+2.4 and +0). Thus, optimal pairings do not correspond simply to language relatedness. We note that when applying multilingual learning to morphological segmentation the best results were obtained for related languages, but only after incorporating declarative knowledge about their lower-level phonological relations using a prior which encourages phonologically close aligned morphemes [108]. Here too, a more complex model which models lower-level morphological relatedness (such as case) may yield better outcomes for closely related languages.

As an upper bound on the merged node model performance, line 7 of table 2.10 shows the results when selecting (with the help of an oracle) the best partner for each language. The average accuracy using this oracle is 95.4%, substantially higher than the average bilingual pairing accuracy of 93.2%, and even somewhat higher than the latent variable model performance of 95%. This gap in performance motivates a closer examination of the relationship between languages that constitute effective pairings.

MergedNode Model									
								coupled with...	
		Avg	BG	CS	EN	ET	HU	RO	SL	SR
accuracy for...	BG	91.3		90.2	**94.7**	92.3	90.6	91.2	91.1	88.7†
	CS	96.9	95.3		97.5	**97.8**	96.3	96.4	97.4	97.4
	EN	95.9	**96.1**	95.9†		95.8†	95.8†	95.8†	**96.1**	96.0
	ET	93.3	93.0	94.0	92.9†		92.2†	93.0	**94.2**	93.9
	HU	96.7	96.8	96.6	96.8	**96.9**		96.8	96.5	96.7
	RO	91.9	**94.1**	90.6†	92.0	91.3	90.3†		91.3	93.9
	SL	89.3	88.5	88.1	89.2	89.8	87.5†	87.5†		**94.8**
	SR	90.2	88.5	88.2	**94.5**	94.2	89.5	85.0	91.4	

Table 2.10: Merged node model accuracy for all language pairs. Each row corresponds to the performance of one language, each column indicates the language with which the performance was achieved. The best result for each language is indicated in bold. All results other than those marked with a † are significantly higher than the monolingual baseline at $p < 0.05$ according to a sign test. See table 2.3 for language name abbreviations.

Cross-lingual Entropy

In a previous publication [111] we proposed using *cross-lingual entropy* as a post-hoc explanation for variation in coupling performance. This measure calculates the entropy of a tagging decision in one language given the identity of an aligned tag in the other language. While cross-lingual entropy seemed to correlate well with relative performance for the four languages considered in that publication, we find that it does not correlate as strongly for all eight languages considered here. We computed the Pearson correlation coefficient [83] between the relative bilingual performance and cross-lingual entropy. For each target language, we rank the remaining seven languages based on two measures: how well the paired language contributes to improved performance of the target, and the cross-lingual entropy of the target language given the coupled language. We compute the Pearson correlation coefficient between these two rankings to assess their degree of overlap. See table D.1 in appendix D for a complete list of results. On average, the coefficient was 0.29, indicating a weak positive correlation between relative bilingual performance and cross-lingual entropy.

Alignment Density

We note that even if cross-lingual entropy had exhibited higher correlation with performance, it would be of little practical utility in an unsupervised scenario since its estimation requires a tagged corpus. Next we consider the density of pairwise lexical alignments between language pairs as a predictive measure of their coupled performance. Since alignments constitute the multilingual anchors of our models, as a practical matter greater alignment density should yield greater opportunities for cross-lingual transfer. From the linguistic viewpoint, this measure may also indirectly capture the correspondence between two languages. Moreover, this measure has the benefit of being computable from an untagged corpus, using automatically obtained GIZA++ alignments. As before, for each target language, we rank the other languages by relative bilingual performance, as well as by the percentage of words in the target language to which they provide alignments. Here we find an average Pearson coefficient of 0.42, indicating mild positive correlation. In fact, if we use alignment density as a criterion for selecting optimal pairing decisions for each target language, we obtain an average accuracy of 94.67% — higher than average bilingual performance, but still somewhat below the performance of the multilingual model.

Model Choice

The choice of model may also contribute to the patterns of variability we observe across language pairs. To test this hypothesis, we ran our latent variable model on all pairs of languages. The results of this experiment are shown in table 2.11. As in the case of the merged node model, the performance of each target language depends heavily on the choice of partner. However, the exact patterns of variability differ in this case from those observed for the merged node model. To measure this variability, we compare the pairing preferences for each language under each of the two models. More specifically, for each target language we rank the remaining seven languages by their contribution under each of our two models, and compute the Pearson coefficient between these two rankings. As seen in the last column of table D.1 in the appendix, we find a coefficient of 0.49 between the two rankings, indicating positive, though far from perfect, correlation.

LatentVariable Model										
									coupled with...	
		Avg	BG	CS	EN	ET	HU	RO	SL	SR
accuracy for...	BG	91.9		92.2	91.9	91.6	91.6	92.1	**92.3**	91.8
	CS	97.2	97.5		97.5	**97.6**	97.4	97.4	96.5	96.8
	EN	95.7	95.7†	95.7†		95.7†	95.6†	95.7†	95.7†	**95.8**†
	ET	93.9	**94.8**	94.3	93.4		92.3†	93.9	94.5	94.1
	HU	96.8	**97.0**	96.8	96.7	96.7		96.8	96.6	96.8
	RO	93.2	94.6	92.1	92.4	92.3	92.1		94.4	**94.7**
	SL	90.5	88.6	87.7	92.4	**95.2**	87.5†	87.6†		94.6
	SR	91.6	**94.7**	88.5	94.5	94.5	89.7	88.0	91.1	

Table 2.11: Accuracy of latent variable model when run on language pairs. Each row corresponds to the performance of one language, each column indicates the language with which the performance was achieved. The best result for each language is indicated in bold. All results other than those marked with a † are significantly higher than the monolingual baseline at $p < 0.05$ according to a sign test. See table 2.3 for language name abbreviations.

Utility of each Language as a Bilingual Partner

We also analyze the overall *helpfulness* of each language. As before, for each target language, we rank the remaining seven languages by the degree to which they contribute to increased target language performance when paired in a bilingual model. We can then ask whether the helpfulness rankings provided by each of the eight languages are correlated with one another — in other words, whether languages tend to be universally helpful (or unhelpful) or whether helpfulness depends heavily on the identity of the target language. We consider all pairs of target languages, and compute the Pearson rank correlation between their rankings of the six supplementary languages that they have in common (excluding the two target languages themselves). When we average these pair-wise rank correlations we obtain a coefficient of 0.20 for the merged node model and 0.21 for the latent variable model. These low correlations indicate that language helpfulness depends crucially on the target language in question. Nevertheless, we can still compute the average helpfulness of each language (across all target languages) to obtain something like a "universal" helpfulness ranking. See table E.1 in the appendix for this ranking. We can then ask whether this ranking correlates with language properties which might be predictive of general helpfulness. We compare the universal helpfulness rankings[10] to language rankings induced by tag-per-token ambiguity (the average number of tags allowed by the dictionary per token in the corpus) as well as trigram entropy (the entropy of the tag distribution given the previous two tags). In both cases we assign the highest rank to the language with *lowest* value, as we expect lower entropy and ambiguity to correlate with greater helpfulness. Contrary to expectations, the ranking induced by tag-per-token ambiguity actually correlates *negatively* with both universal helpfulness rankings by very small amounts (-0.28 for the merged node model and -0.23 for the latent variable model). For both models, Hungarian, which has the lowest tag-per-token ambiguity of all eight languages, had the *worst* universal

[10] We note that the universal helpfulness rankings obtained from each of the two multilingual models match each other only roughly: their correlation coefficient with one another is 0.50. In addition, "universal" in this context refers only to the eight languages under consideration and the rankings could very well change in a wider multilingual context.

helpfulness ranking. The correlations with trigram entropy were only a little more predictable. In the case of the latent variable model, there was no correlation at all between trigram entropy and universal helpfulness (-0.01). In the case of the merged node model, however, there was moderate positive correlation (0.43).

2.12.2 Adding Languages in the Latent Variable Model

While bilingual performance depends heavily on the choice of language pair, the latent variable model can easily incorporate all available languages, obviating the need for any choice. To test performance as the number of languages increases, we ran the latent variable model with all possible subsets of the eight languages in the full lexicon as well as all three reduced lexicon scenarios. Figures 2-5, 2-6, 2-7, and 2-8 plot the average accuracy as the number of available languages varies for all four lexicon scenarios (in decreasing order of the lexicon size). For comparison, the monolingual and average bilingual baseline results are given. In all scenarios, our latent variable model steadily gains in accuracy as the number of available languages increases, and in most scenarios sees an appreciable uptick when going from seven to eight languages. In the full lexicon case, the gap between supervised and unsupervised performance is cut by nearly two thirds under the unsupervised latent variable model with all eight languages.

Interestingly, as the lexicon is reduced in size, the performance of the bilingual merged node model gains relative to the latent variable model on pairs. In the full lexicon case, the latent variable model is clearly superior, whereas in the two moderately reduced lexicon cases, the performance on pairs is more or less the same for the two models. In the case of the drastically reduced lexicon (100 words), the merged node model is the clear winner. Thus, it seems that of the two models, the performance gains of the latent variable model are more sensitive to the size of the lexicon.

The same four figures (2-5, 2-6, 2-7, and 2-8) also show the multilingual performance broken down by language. All languages except for English tend to increase in accuracy as additional languages are added to the mix. Indeed, in the two cases of moderately reduced lexicons (Figures 2-6 and 2-7) all languages except for English show steady large gains which actually increase in size when going from seven to the full set of eight languages. In the full lexicon case (Figure 2-5), Estonian, Romanian, and Slovene display steady increases until the very end. Hungarian peaks at two languages, Bulgarian at three languages, and Czech and Serbian at seven languages. In the more drastic reduced lexicon case (Figure 2-8), the performance across languages is less consistent and the gains when languages are added are less stable. All languages report gains when going from one to two languages, but only half of them increase steadily up to eight languages. Two languages seem to trend downward after two or three languages, and the other two show mixed behavior.

In the full lexicon case (Figure 2-5), English is the only language which fails to improve. In the other scenarios, English gains initially but these gains are partially eroded when more languages are added. It is possible that English is an outlier since it has significantly lower tag transition entropy than any of the other languages (see table 2.3). Thus it may be that internal tag transitions are simply more informative for English than any information that can be gleaned from multilingual context.

2.12.3 Analysis of Multilingual Gains

In this section we seek to better understand the source of improvements for the latent variable model. Intuitively, we would expect the greatest benefits to accrue to test-set words which are frequently aligned in the parallel training corpus, since they have the direct influence of superlingual tags. Ideally we would see improvements for less frequently aligned words as well, through the indirect propagation of multilingual information.

First we examine the distribution of alignments by part-of-speech tag. For each part-of-speech, table B.3 (in appendix B) shows the percentage of occurrences with a direct edge from a superlingual tag. Determiners and articles, which exist for only two of the languages studied, are aligned about 50% of the time. In contrast, verbs are aligned about 73% of the time, while nouns are aligned over 85% of the time. Thus we see that alignments are unevenly distributed across parts-of-speech.

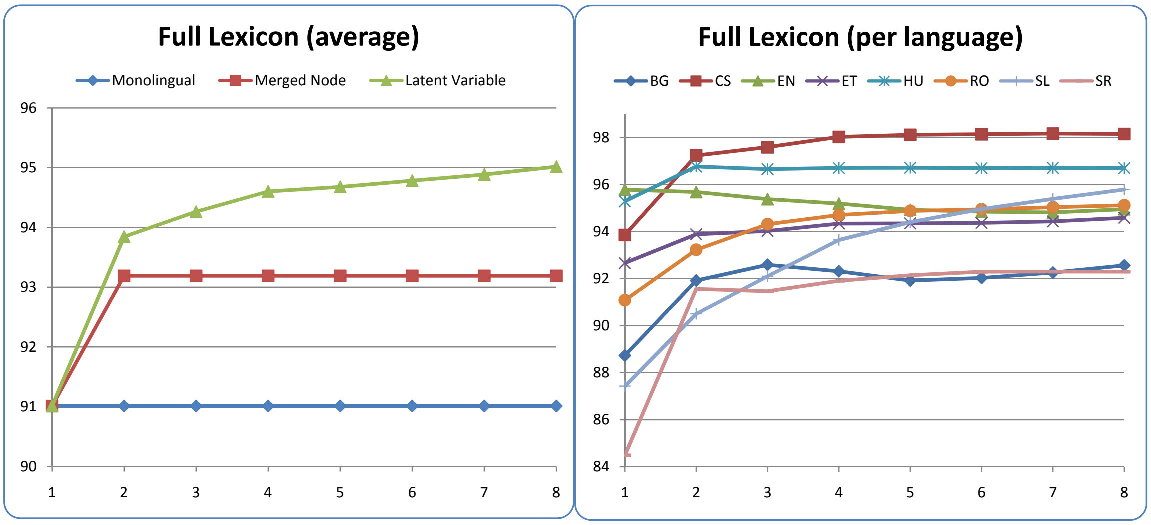


Figure 2-5: The performance of the latent variable model as the number of languages varies (averaged over all subsets of the eight languages for each size). LEFT: Average performance across all languages. Scores for monolingual and bilingual merged node models are given for comparison. RIGHT: The Performance for each individual language as the number of available languages varies.

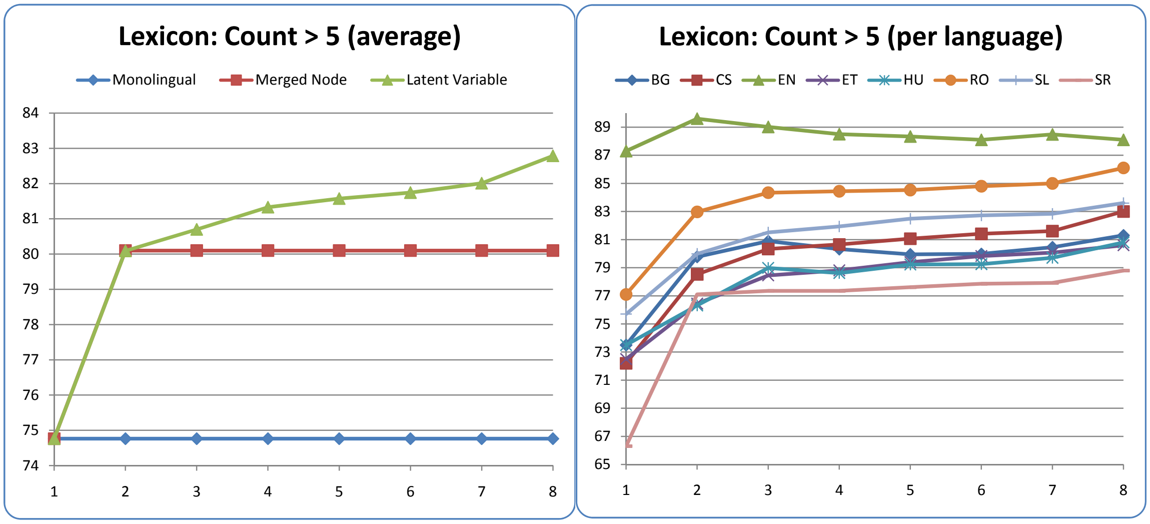


Figure 2-6: The performance of the latent variable model for the reduced lexicon scenario (Counts > 5), as the number of languages varies (averaged over all subsets of the eight languages for each size). LEFT: Average performance across all languages. Scores for monolingual and bilingual merged node models are given for comparison. RIGHT: The Performance for each individual language as the number of available languages varies.

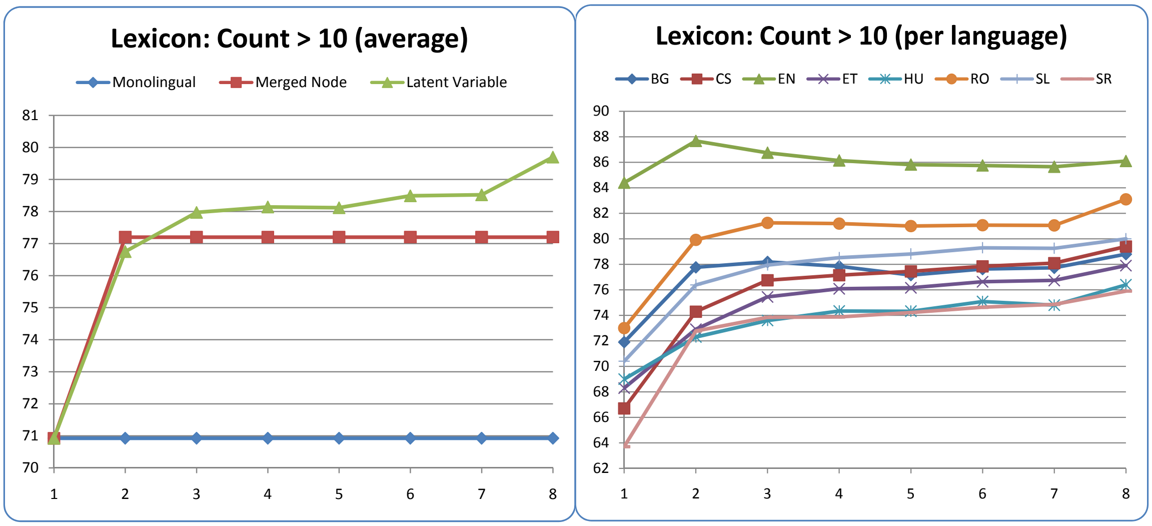


Figure 2-7: The performance of the latent variable model for the reduced lexicon scenario (Counts > 10), as the number of languages varies (averaged over all subsets of the eight languages for each size). LEFT: Average performance across all languages. Scores for monolingual and bilingual merged node models are given for comparison. RIGHT: The Performance for each individual language as the number of available languages varies.

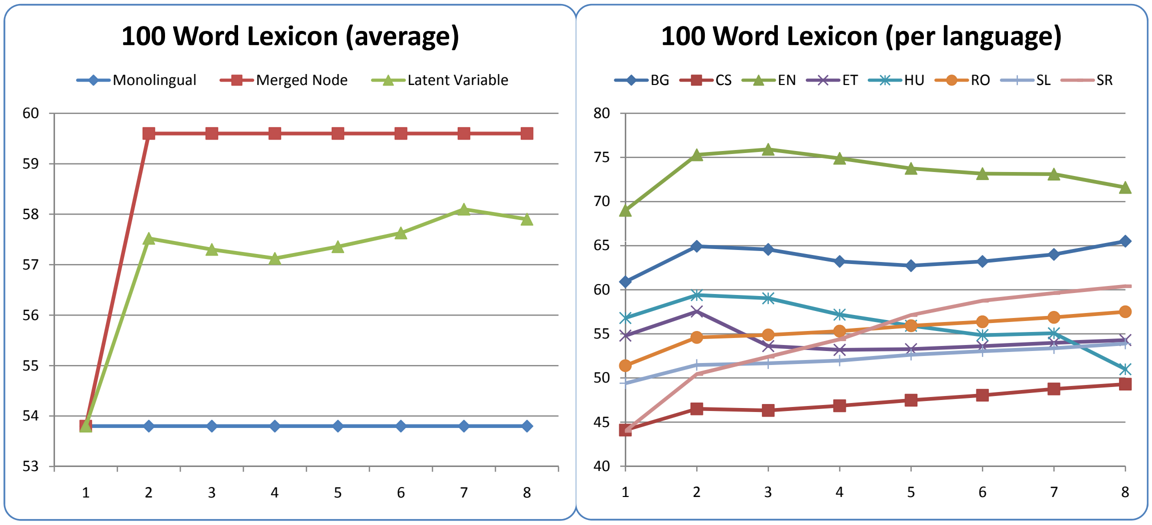


Figure 2-8: The performance of the latent variable model for the reduced lexicon scenario (100 words), as the number of languages varies (averaged over all subsets of the eight languages for each size). LEFT: Average performance across all languages. Scores for monolingual and bilingual merged node models are given for comparison. RIGHT: The Performance for each individual language as the number of available languages varies.

Next we analyze the performance of the latent variable model (when trained on all 8 languages) for words with varying degrees of alignment density. For each word in the training corpus, we count the frequency for which it occurs with a superlingual tag. We then order the test-set words in increasing order of these frequencies and calculate the performance of the model (as well as the baseline) on each initial subset of words. Figure 2-9 shows the results. As can be seen, the relative multilingual performance tends to be greatest for frequently aligned words. For words which align fewer than 30% of the time, there is virtually no difference in performance. Thereafter, multilingual performance systematically diverges from the baseline.

Finally, we break down the relative performance of the latent variable model for both in-vocabulary and out-of-vocabulary words. Recall that in all testing scenarios, we provide our models with a seed dictionary which lists the possible parts-of-speech for some subset of words. Thus, we can partition the test-set into two portions: those words for which an entry in the tag lexicon is available, and those words for which an entry is unavailable. Table 2.12 breaks down the performance of the latent variable model (when trained on all 8 languages) on in-vocabulary and out-of-vocabulary words. In nearly all cases, multilingual performance is superior to monolingual performance. However, the greatest relative gains are seen for out-of-vocabulary words.

2.12.4 Superlingual Tag Values

In this section we analyze the superlingual tags and their corresponding part-of-speech distributions, as learned by the latent variable model. Recall that each superlingual tag intuitively represents a discovered *multilingual context* and that it is through these tags that multilingual information is propagated. More formally, each superlingual tag provides a complete distribution over parts-of-speech for each language, allowing the encoding of both primary and secondary preferences separately for each language. These preferences then interact with the language-specific context (i.e. the surrounding parts-of-speech and the corresponding word). We

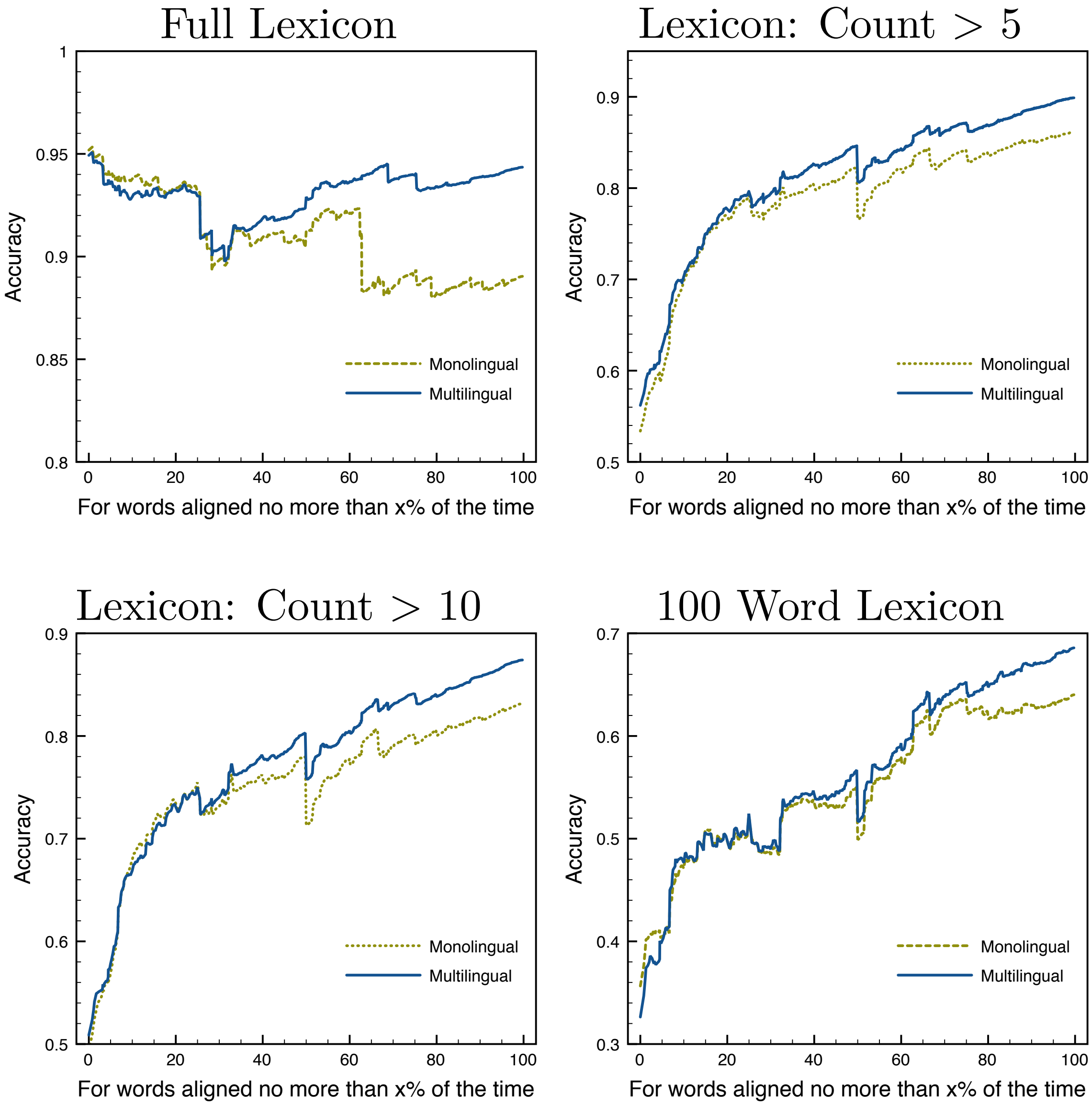


Figure 2-9: The average performance of the latent variable model and baseline for words with different alignment frequencies. The words are ordered by increasing frequency of alignment, and performance is measured on each initial subset of words. The horizontal axis gives the maximum alignment frequency of each subset and the vertical axis gives test-set tag accuracy for that subset. E.g. at point 50 along the horizontal axis we only consider words which align 50% or less of the time. The final right-hand point gives the performance on all words.

		All	BG	CS	EN	ET	HU	RO	SL	SR
Counts > 5	MONO: in	90.21	89.57	91.25	93.90	88.13	94.40	87.87	92.12	84.01
	MULTI: in	92.91	89.98	96.83	92.29	92.54	94.31	93.95	94.89	89.25
	MONO: out	23.08	26.07	21.25	31.03	21.20	22.79	25.85	20.70	19.55
	MULTI: out	46.01	50.29	44.14	51.85	39.54	46.61	50.27	44.82	42.40
Counts > 10	MONO: in	91.43	89.66	91.41	94.66	89.39	95.53	87.90	92.63	89.99
	MULTI: in	92.84	89.89	97.04	92.37	92.56	94.25	93.83	94.78	88.91
	MONO: out	26.41	30.39	25.18	34.48	23.51	25.59	29.97	22.19	22.82
	MULTI: out	48.91	52.60	48.48	54.56	45.71	44.90	52.95	46.83	47.24
100 Words	MONO: in	89.30	87.53	87.80	95.57	91.74	96.04	90.48	92.82	70.56
	MULTI: in	92.84	86.84	97.07	93.37	94.47	95.74	95.13	95.71	85.74
	MONO: out	26.28	37.15	19.25	35.13	28.31	30.17	20.69	18.24	23.81
	MULTI: out	31.41	49.48	24.55	24.45	27.58	24.15	30.66	22.41	46.94

Table 2.12: In-vocabulary vs out-of-vocabulary performance for the latent variable model in the three reduced lexicon scenarios. The monolingual baseline performance is given for comparison. See table 2.3 for language name abbreviations.

place a Dirichlet process prior on the superlingual tags, so the number of sampled values is dictated by the complexity of the data. In fact, as shown in table 2.13, the number of sampled superlingual tags steadily increases with the number of languages. As multilingual contexts becomes more complex and diverse, additional superlingual tags are needed.

Number languages	2	3	4	5	6	7	8
Number superlingual tag values	11.07	12.57	13.87	15.07	15.79	16.13	16.50

Table 2.13: Average number of sampled superlingual tag values as the number of languages increases.

Next we analyze the part-of-speech tag distributions associated with superlingual tag values. Most superlingual tag values correspond to low entropy tag distributions, with a single dominant part-of-speech tag across all languages. See, for example, the distributions associated with superlingual tag value 6 in table 2.14, all of which favor nouns by large margins. Similar sets of distributions occur favoring verbs, adjectives, and the other primary part-of-speech categories. In fact, among the seventeen sampled superlingual tag values, nine belong to this type, and they cover 80% of actual superlingual tag instances. The remaining superlingual tags correspond to more complex cross-lingual patterns. The associated tag distributions in those cases favor different part-of-speech tags in various languages and tend to have higher entropy, with the probability mass spread more evenly over two or three tags. One such example is the set of distributions associated with the superlingual tag value 14 in table 2.14, which seems to be a mixed noun/verb class. In six out of eight languages the most favored tag is verb, while a strong secondary choice in these cases is noun. However, for Estonian and Hungarian, this preference is reversed, with nouns being given higher probability. This superlingual tag may have captured the phenomenon of "light verbs," whereby verbs in one language correspond to a combination of a noun and verb in another language. For example the English verb *whisper*/V, when translated into Urdu, becomes the collocation *whisper*/N *do*/V. In these cases, verbs and nouns will often be aligned to one another, requiring a more complex superlingual tag. The analysis of these examples shows that the superlingual tags effectively learns both simple and complex cross-lingual patterns

tag value 6	bg	$P(N) = 0.91,\ P(A) = 0.04,\ ...$	tag value 14	bg	$P(V) = 0.66,\ P(N) = 0.21,\ ...$
	cs	$P(N) = 0.92,\ P(A) = 0.03,\ ...$		cs	$P(V) = 0.60,\ P(N) = 0.22,\ ...$
	en	$P(N) = 0.97,\ P(V) = 0.00,\ ...$		en	$P(V) = 0.55,\ P(N) = 0.25,\ ...$
	et	$P(N) = 0.91,\ P(V) = 0.03,\ ...$		et	$P(N) = 0.52,\ P(V) = 0.29,\ ...$
	hu	$P(N) = 0.85,\ P(A) = 0.06,\ ...$		hu	$P(N) = 0.44,\ P(V) = 0.34,\ ...$
	ro	$P(N) = 0.90,\ P(A) = 0.04,\ ...$		ro	$P(V) = 0.45,\ P(N) = 0.33,\ ...$
	sl	$P(N) = 0.94,\ P(A) = 0.03,\ ...$		sl	$P(V) = 0.55,\ P(N) = 0.24,\ ...$
	sr	$P(N) = 0.92,\ P(A) = 0.03,\ ...$		sr	$P(V) = 0.49,\ P(N) = 0.26,\ ...$

Table 2.14: Part-of-speech tag distributions associated with two superlingual latent tag values. Probabilities of only the two most probable tags for each language are shown. See table 2.3 for language name abbreviations.

2.12.5 Performance with Reduced Data

One potential objection to the claims made in this brief section is that the improved results may be due merely to the addition of more data, so that the multilingual aspect of the model may be irrelevant. We test this idea by evaluating the monolingual, merged node, and latent variable systems on training sets in which the number of examples is reduced by half. The multilingual models in this setting have access to exactly half as much data as the monolingual model in the original experiment. As shown in table 2.15, both the monolingual baseline and our models are quite insensitive to this drop in data. In fact, both of our models, when trained on half of the corpus, still outperform the monolingual model trained on the entire corpus. This indicates that the performance gains demonstrated by multilingual learning cannot be explained merely by the addition of more data.

	Avg	BG	CS	EN	ET	HU	RO	SL	SR
Monolingual: full data	91.2	88.7	93.9	95.8	92.7	95.3	91.1	87.4	84.5
Monolingual: half data	91.0	88.8	93.8	95.7	92.6	95.3	90.2	87.5	84.5
MergedNode: (*avg.*) full data	93.2	91.3	96.9	95.9	93.3	96.7	91.9	89.3	90.2
MergedNode: (*avg.*) half data	93.0	91.1	96.6	95.7	92.7	96.7	92.0	88.9	89.9
LatentVariable: full data	95.0	92.6	98.2	95.0	94.6	96.7	95.1	95.8	92.3
LatentVariable: half data	94.7	92.6	97.8	94.7	93.9	96.7	94.4	95.4	92.2

Table 2.15: Tagging accuracy on reduced training dataset, with complete tag dictionaries; results on the full training dataset are repeated here for comparison. The first column reports average results across all languages. See table 2.3 for language name abbreviations.

2.13 Conclusions

The key hypothesis of multilingual learning is that by combining cues from multiple languages, the structure of each becomes more apparent. We considered two ways of applying this intuition to the problem of unsupervised part-of-speech tagging: a model that directly merges tag structures for a pair of languages into a single sequence and a second model which instead incorporates multilingual context using latent variables.

Our results demonstrate that by incorporating multilingual evidence we can achieve impressive performance gains across a range of scenarios. When a full lexicon is available, our two models cut the gap between unsupervised and supervised performance by nearly one third (merged node model, averaged over all pairs) and two thirds (latent variable model, using all eight languages). For all but one language, we observe performance gains as additional languages are added. The sole exception is English, which only gains from additional languages in reduced lexicon settings.

In most scenarios, the latent variable model achieves better performance than the merged node model, and has the additional advantage of scaling gracefully with the number of languages. These observations suggest that the non-parametric latent variable structure provides a more flexible paradigm for incorporating multilingual cues. However, the benefit of the latent variable model relative to the merged node model (even when running both models on pairs of languages) seems to decrease with the size of the lexicon. Thus, in practical scenarios where only a small lexicon or no lexicon is available, the merged node model may represent a better choice.

Our experiments have shown that performance can vary greatly depending on the choice of additional languages. It is difficult to predict *a priori* which languages constitute good combinations. In particular, language relatedness itself cannot be used as a consistent predictor as sometimes closely related languages constitute beneficial couplings and sometimes unrelated languages are more helpful. We identify a number of features which correlate with bilingual performance, though we observe that these features interact in complex ways. Fortunately, our latent variable model allows us to bypass this question by simply using all available languages.

Limitations and Future Work

In both of our models lexical alignments play a crucial role, as they determine the topology of the model for each sentence. In fact, we observed a positive correlation between alignment density and bilingual performance, indicating the importance of high quality alignments. In our experiments, we considered the alignment structure an observed variable, produced by standard MT tools which operate over pairs of languages. An interesting alternative would be to incorporate alignment structure into the model itself, to find alignments best tuned for tagging accuracy based on the evidence of multiple languages rather than pairs.

Another limitation of the two models is that they only consider one-to-one lexical alignments. When pairing isolating and synthetic languages[11] it should be beneficial to align short analytical phrases consisting of multiple words to single morpheme-rich words in the other language. To do so would involve flexibly aligning and chunking the parallel sentences throughout the learning process.

Finally, we consider two technical limitations of the latent variable model. The first is that we employ only a single distribution over superlingual tags (drawn from a Dirichlet process). However, not all superlingual tags have edges into all languages. In fact, the *majority* of superlingual tags in our corpus point to no more than half of all eight languages. It is likely that the alignments over each subset of the languages will carry with them a unique distribution over parts-of-speech. In other words, the very fact that an alignment only occurs between two particular languages, for example, might provide some useful information for part-of-speech selection. In future work, we can address this concern by positing a separate distribution over superlingual tags for each unique subset of languages.

[11] Isolating languages are those with a morpheme to word ratio close to one, and synthetic languages are those which allow multiple morphemes to be easily combined into single words. English is an example of an isolating language, whereas Hungarian is a synthetic language.

The second limitation is that our model locally normalizes all probability distributions. In its current form, the probability of tag y_i (with an edge from superlingual tag s) is given by the renormalized product:

$$P(y_i \mid y_{i-1}, s) = \frac{P(y_i \mid y_{i-1}) \cdot P(y_i \mid s)}{Z},$$

where the first factor is the language-specific transition distribution and the second factor is the distribution over parts-of-speech given by superlingual tag s. As discussed earlier in the chapter, this factorization allows our model to trade-off monolingual cues against multilingual cues. Either distribution can rule out particular tag assignments by assigning them very low probability. However, because each of these two distributions is locally normalized, the model is unable to express its *confidence* in one over the other. A more general formulation would replace these two distributions with unnormalized scores:

$$P(y_i \mid y_{i-1}, s) = \frac{score(y_i \mid y_{i-1}) \cdot score(y_i \mid s)}{Z}$$

The total magnitude of transition and superlingual scores could thus vary independently, allowing the model to express its *confidence* in each source of information. An even more radical generalization of our model would be to eschew local normalization in its entirety, instead using a globally normalized Markov Random Field. The main challenge in this setting is computational. Applying inference to undirected models with many complex latent variables is still an open research problem [82].

A more general direction for future work is to incorporate even more sources of multilingual information, such as additional languages and declarative knowledge of their typological properties [26]. In this chapter we showed that performance improves as the number of languages increases. We were limited by our corpus to eight languages, but we envision future work on massively parallel corpora involving dozens of languages as well as learning from languages with non-parallel data.

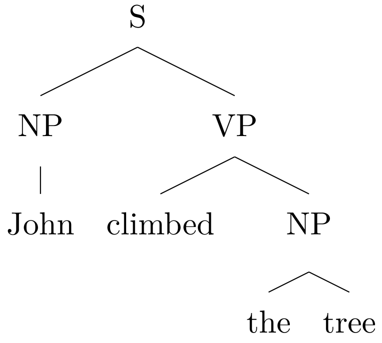
Chapter 3

Unsupervised Multilingual Grammar Induction

In the previous chapter we considered the task of part-of-speech tagging. In that task, the *structure* of latent variables was determined by the word-aligned sentences, and was thus considered fixed and observed. In contrast, this chapter considers a more complex task, where latent structure itself must be predicted. In particular, we consider the task of constituency bracketing. The goal is to predict the nested bracketing of each sentence which reflects its underlying syntax. Thus, for the sentence *John climbed the tree*, our goal would be to produce the bracketing:

[John [climbed [the tree]]]

which reflects the underlying syntactic structure:



In the unsupervised scenario, hand-annotated training data is not available. Instead, we must rely on the patterns of the words themselves as an implicit guide to deeper structure. This has proven to be quite difficult in the monolingual setting [20, 64].

3.1 Chapter Overview

Section 3.2 gives a broad introduction to the chapter. We argue that a multilingual approach will lead to more accurate parse predictions. The key challenge is to capture cross-lingual regularities while still allowing significant divergence between parallel trees. We briefly describe our approach and summarize our experimental findings. Section 3.3 compares our approach to previous unsupervised grammar induction work. Section 3.4 describes our modeling approach in great detail, and section 3.5 describes our inference algorithm. Section 3.6 describes our experiments on three bilingual corpora and reports our results. Section 3.7 completes the chapter with some concluding remarks.

3.2 Introduction

In this chapter, we explore the application of multilingual learning to unsupervised grammar induction. Our goal is to improve parsing performance on monolingual test data by using unsupervised bilingual cues at training time. Following previous work on monolingual grammar induction [20, 64], we focus on unlabeled constituency brackets.

The key premise of our approach is that ambiguous syntactic structures in one language may correspond to less uncertain structures in the other language. For instance, the English sentence:

I saw the student from MIT

exhibits the classical problem of prepositional phrase attachment ambiguity [25].

The prepositional phrase *from MIT* may form a constituent with the adjacent noun phrase, yielding the parse:

I [*saw* [*the student* [*from MIT*]]],

meaning that the student herself is from MIT. In theory, though, the prepositional phrase may also directly modify the entire verb phrase:

I [*saw* [*the student*] [*from MIT*]],

meaning that the act of seeing the student was undertaken from MIT. The Urdu translation of this sentence, which can be glossed literally as:

I [[*MIT of student*] *saw*],

displays no such ambiguity. An explicit genitive phrase *MIT of student* is used, forming an unambiguous noun phrase. Knowing the word-level correspondences between these sentences should therefore help us resolve the English prepositional phrase attachment ambiguity in favor of the noun phrase attachment. In addition, systematic differences in word order can also be informative. Without much knowledge of Urdu grammar, one might have mistakenly placed the first two words *I MIT* into a single bracket on their own. However, after observing that the corresponding words in the English sentence appear respectively in the first and last positions, we can safely rule out this possibility.

One of the general aims of unsupervised multilingual learning is to exploit cross-lingual patterns discovered in data, while still allowing a wide range of language-specific idiosyncrasies. Especially at the syntactic level, languages differ greatly in their expression of similar meanings. Thus, one of the key challenges here is one of *representation*: How can we simultaneously parse two parallel sentences, represent what is common between them, while still allowing their syntactic structures to diverge in significant ways?

To answer this question, we have adapted a computational formalism known as *unordered tree alignment* [60] to our probabilistic setting. Under this formalism, any two trees can be embedded in an *alignment tree*. This alignment tree allows arbitrary parts of the two trees to diverge in structure, permitting language-specific grammatical structure to be preserved. A computational advantage of this formalism is that it allows us compute the marginal probability of tree pairs and to sample alignments in polynomial time, using a dynamic program.

We formulate a generative Bayesian model which seeks to explain the observed parallel data through a combination of bilingual and monolingual parameters. Our model views each pair of sentences as having been generated as follows: First an alignment tree is drawn. Each node in this alignment tree contains either a solitary monolingual constituent or a pair of coupled bilingual constituents. For each solitary monolingual constituent, a sequence of part-of-speech tags is drawn from a language-specific distribution. For each pair of coupled bilingual constituents, a pair of part-of-speech sequences are drawn jointly from a cross-lingual distribution. Word-level alignments are then drawn based on the tree alignment. Finally, parallel sentences are assembled from these generated part-of-speech sequences and word-level alignments.

To perform inference under this model, we use a Metropolis-Hastings within-Gibbs sampler. We sample pairs of trees and then compute marginalized probabilities over all possible alignments using dynamic programming.

We test the effectiveness of our bilingual grammar induction model on three corpora of parallel text: English-Korean, English-Urdu and English-Chinese. The model is trained using bilingual data with automatically induced word-level alignments, but is tested on purely monolingual data for each language. In all cases, our model outperforms a state-of-the-art baseline: the Constituent Context Model (CCM) [64], sometimes by substantial margins. On average, over all the testing scenarios that we studied, our model achieves an absolute increase in F-measure of 8.8 points, and a 19% reduction in error relative to a theoretical upper bound.

3.3 Related Work

The unsupervised grammar induction task has been studied extensively, mostly in a monolingual setting [20, 113, 64, 102]. While Probabilistic Context-free Grammars (PCFG) perform poorly on this task, the CCM [64] has achieved large gains in performance and remains the state-of-the-art probabilistic model for unsupervised constituency parsing. We therefore use the CCM as our basic model of monolingual syntax.

While there has been some previous work on bilingual context-free grammar parsing, it has mainly focused on improving machine translation systems rather than monolingual parsing accuracy. Research in this direction was pioneered by [122], who developed Inversion Transduction Grammars to capture cross-lingual grammar variations such as phrase reorderings. More general formalisms (such as Synchronous Grammars) for the same purpose were later developed [123, 22, 78, 32, 129, 11]. We know of only one study which evaluates these bilingual grammar formalisms on the task of grammar induction itself [104]. Both our model and even the monolingual CCM baseline yield far higher performance on the same Korean-English corpus. In our model, we seek to learn syntactic correspondences between the languages while using word-level alignments as a guide to finding constituent-level alignments. The tree alignment formalism we employ is more flexible than Inversion Transduction Grammars (as well as Synchronous Grammars) in that it allows a node in any part of either tree to remain unaligned. This results in the possibility of nodes aligning across different heights of the two trees (see figure 3-1).

Our approach is closer in spirit to the unsupervised bilingual parsing model developed by Kuhn [70], which aims to improve monolingual performance. Assuming that trees induced over parallel sentences have to exhibit certain structural regularities, Kuhn manually specifies a set of rules for determining when parsing decisions in the two languages are inconsistent with GIZA++ word-level alignments. By incorporating these constraints into the EM algorithm he was able to improve performance over a monolingual unsupervised PCFG. Still, the performance falls short of state-of-the-art monolingual models such as the CCM.

More recently, there has been a body of work attempting to improve *supervised* parsing performance by exploiting syntactically annotated parallel data. One strand of this work has been pursued in the projection framework, which assumes that syntactic annotation is available only in a source language [58, 124]. Syntactic trees from the source language are transferred onto a target language via the aligned parallel corpus. The projected annotations are used to train a parser for the target language. As in our approach, these methods explore bilingual correspondences between syntactic structures. However, such correspondences are encoded manually or trained from annotated corpora, whereas we induce them automatically using only raw parallel text.

An alternative supervision scenario considers the case where syntactic annotations are available for both languages. Burkett and Klein [17] develop an algorithm for simultaneously training English and Chinese parsers on a bilingual tree bank. Their method proceeds by first training standard supervised parsers for each language. They then define a log-linear reranking model which considers the highest probability parses for each language, and ranks them according to their agreement with one another. The reranking model treats tree-to-tree node alignments as a hidden variable. No structure over the node-alignments is assumed except that they are one-to-one. As a result, summing over all possible alignments is #P-complete, and they must resort to approximations. In contrast, our tree alignment formalism permits the marginalization and sampling of tree alignments in polynomial time. In subsequent work, Burkett et al. [18] consider the scenario where supervised parsers exist for both languages but are improved by the addition of unannotated bilingual parallel text. Most recently, Burkett et al. [16] develop a model for joint bilingual parsing and word alignment. These methods simultaneously learns to parse each language and to induce the connection between derived parses. The evaluation of these algorithms have demonstrated the value of cross-lingual constraints: parsers trained on bilingual annotated data yield improvement over monolingual counterparts. In contrast to this line of work, we assume no annotated texts in either

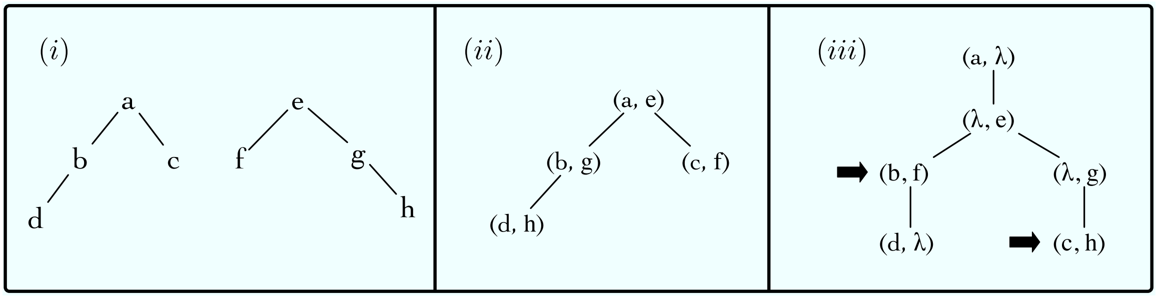


Figure 3-1: A pair of trees (i) and two possible alignment trees. In (ii), no empty spaces are inserted, but the order of one of the original tree's siblings has been reversed. In (iii), only two pairs of nodes have been aligned (indicated by arrows) and many empty spaces inserted.

language.

Finally, we note three recent papers on multilingual dependency parsing. The first of these ties the parameters of multilingual parsers through a shared logistic normal prior [23]. While the primary performance gains occur when tying related parameters within a language, some additional benefit is observed through bilingual tying, even in the absence of a parallel corpus. The second paper, which appeared after the initial publication of this work, considers the tying of parameters across a broad set of languages [6]. Interestingly, the best results were found when the prior over parameters mirrored the phylogenetic relationship of the languages. Finally, in a very recent publication Naseem et al. [84] use hand-specified universal rules of syntax which are probabilistically refined separately for each language. Although no explicit multilingual modeling is performed, they report the best unsupervised results for six different languages.

3.4 Model

We propose an unsupervised Bayesian model for learning bilingual syntactic structure using parallel corpora. Our key premise is that difficult-to-learn syntactic structures of one language may correspond to simpler or less uncertain structures in the other language. We treat the part-of-speech tag sequences of parallel sentences, as well as their word-level alignments, as observed data. We obtain these word-level alignments automatically using GIZA++ [90].

Our model seeks to explain this observed data through a generative process whereby two aligned parse trees are produced jointly. Though they are aligned, arbitrary parts of the two trees are permitted to diverge, accommodating language-specific grammatical structure. In effect, our model loosely binds the two trees: node-to-node alignments need only be used where repeated bilingual patterns can be discovered in the data.

3.4.1 Tree Alignments

We achieve this loose binding of trees by adapting *unordered tree alignment* [60] to a probabilistic setting. Under this formalism, any two trees T_1 and T_2 can be aligned through the following steps:

1. Insert empty nodes (labeled with λ) into T_1
2. Insert empty nodes (labeled with λ) into T_2
3. Reorder sibling nodes in T_1
4. Reorder sibling nodes in T_2
5. Repeat steps 1-4 until the resulting trees T_1' and T_2' are identical in structure
6. Overlay T_1' and T_2' to obtain an *alignment tree* $\mathcal{A}$

The alignment tree $\mathcal{A}$ embeds the original two trees within it. Each node consists of a pair

$$(x, y),$$

where x and y are corresponding nodes in T_1' and T_2', respectively. If x and y are both original nodes from T_1 and T_2, then we say that these nodes are aligned. If however, $x \in T_1$ and $y = \lambda$, then we say that x remains unaligned. Similarly, if $x = \lambda$ and $y \in T_2$ then we say that y remains unaligned.

Intuitively, an alignment $\mathcal{A}$ can allow arbitrary parts of each tree to remain unaligned to the other tree. However, alignments must respect the basic hierarchical structure of each tree. For example, assume that the pair of nodes (x_1, y_1) are aligned, and a second pair of nodes (x_2, y_2) are aligned as well. If x_1 is an ancestor of x_2 in the original tree T_1, then y_1 must be an ancestor of y_2 in tree T_2 as well.

The flexibility of the tree alignment formalism can be demonstrated by two extreme cases: (1) an alignment between two trees may actually align *none* of their individual nodes, instead inserting an empty space λ for each of the original two trees' nodes; (2) if the original trees are already structurally identical up to sibling order, the alignment may match their nodes exactly, without inserting any empty spaces. See Figure 3-1 for an example.

Tree alignment can be viewed as a somewhat more restrictive variant of the well known *tree edit-distance* formalism [114]. In fact, each alignment corresponds to an edit-sequence in which all insertions precede all deletions [60]. However, this restriction yields computational benefits. Computing the optimal edit-distance between unordered trees is NP-hard [130]. In contrast, trees T_1 and T_2 with bounded degree can be optimally aligned (i.e. aligned with as few empty node insertions as possible) in time $O(|T_1| \cdot |T_2|)$ using a dynamic program. As we will see in section 3.5, our inference procedure relies heavily on similar dynamic programs.

3.4.2 CCM Overview

As our basic model of syntactic structure, we adopt the Constituent-Context Model (CCM) of Klein and Manning [64]. In this section, we summarize that model in order to provide the necessary background for our bilingual model.

The CCM is a generative model of the part-of-speech sequences of observed sentences, ignoring the words themselves. For example, the sentence *John climbed the tree* would be considered the following observation:

NNP VBD DT NN,

where NNP denotes a personal noun, VBD denotes a past-tense verb, DT denotes a definite determiner, and NN denotes a common noun.

Trees, Constituents, and Distituents

According to the CCM, the above sequence was probabilistically generated by an underlying binary tree structure, in this case:

$$T = [\underline{\quad}_1\ [\underline{\quad}_2\ [\underline{\quad}_3\ \underline{\quad}_4]]]$$

The CCM assumes a prior uniform distribution over all such tree-structures (up to some very large size). Next, we consider every span of leaves in this tree, making a crucial distinction between *constituents* and *distituents.* A constituent is a span of leaves which are exactly dominated by a single node in the tree. In a slight abuse of notation, we will write $(i, j) \in T$ if a node $x \in T$ exactly dominates the leaves i through j. Thus, beside the leaf nodes themselves (which are constituents, but are ignored here for ease of exposition), tree T contains three constituents:

$$\underline{\quad}_1\ \underline{\quad}_2\ \underline{\quad}_3\ \underline{\quad}_4 \qquad\qquad \underline{\quad}_2\ \underline{\quad}_3\ \underline{\quad}_4 \qquad\qquad \underline{\quad}_3\ \underline{\quad}_4$$

Every other leaf-span in the tree is a *distituent.* Thus tree T contains three distituents:

$$\underline{\quad}_1\ \underline{\quad}_2 \qquad\qquad \underline{\quad}_1\ \underline{\quad}_2\ \underline{\quad}_3 \qquad\qquad \underline{\quad}_2\ \underline{\quad}_3$$

Yields

For every constituent, we draw a constituent-*yield*: a sequence of parts-of-speech to label the corresponding leaf nodes. Thus for tree T we draw three constituent-yields:

$$y(1,4) \ = \ \textsc{nnp vbd dt nn} \qquad\qquad y(2,4) \ = \ \textsc{vbd dt nn} \qquad\qquad y(3,4) \ = \ \textsc{dt nn}$$

Likewise, for every distituent, we draw a distituent-*yield.* Thus for tree T we draw three distituent-yields:

$$y(1,2) \ = \ \textsc{nnp vbd} \qquad\qquad y(1,3) \ = \ \textsc{nnp vbd dt} \qquad\qquad y(2,3) \ = \ \textsc{vbd dt}$$

Constituent yields are drawn from a multinomial distribution π^C over all part-of-speech sequences (up to some large fixed length), and distituent yields are drawn from a corresponding distribution π^D. The notation $y(i, j)$ denotes the yield spanning leaves i through j (inclusive).

Contexts

Next, constituent and distituent *contexts* are drawn. The context of a leaf-span is the pair of parts-of-speech labeling the leaves to the immediate left and right of the span (substituting a special symbol # when the span includes the left-most or right-most leaves of the tree). Thus for tree T we generate the following three constituent-contexts:

$$c(1,4) = (\#, \#) \qquad c(2,4) = (\textsc{nnp}, \#) \qquad c(3,4) = (\textsc{vbd}, \#)$$

Likewise, we generate the following three distituent-contexts:

$$c(1,2) = (\#, \textsc{dt}) \qquad c(1,3) = (\#, \textsc{nn}) \qquad c(2,3) = (\textsc{nnp}, \textsc{nn})$$

The constituent-contexts are drawn from a multinomial distribution ϕ^C over all part-of-speech pairs, and distituent-contexts are drawn from a corresponding distribution ϕ^D. The notation $c(i, j)$ denotes the context for leaf span i through j (inclusive).

Over-generation

Note that the CCM *over-generates* each observed part-of-speech sequence. In the example above, we independently generated each of the following values:

$$\begin{array}{rcccccccl}
y(1,4) & = & & \textsc{nnp} & \textsc{vbd} & \textsc{dt} & \textsc{nn} & & \text{// Constituent yields} \\
y(2,4) & = & & \_\_ & \textsc{vbd} & \textsc{dt} & \textsc{nn} & & \\
y(3,4) & = & & \_\_ & \_\_ & \textsc{dt} & \textsc{nn} & & \\
y(1,2) & = & & \textsc{nnp} & \textsc{vbd} & \_\_ & \_\_ & & \text{// Distituent yields} \\
y(1,3) & = & & \textsc{nnp} & \textsc{vbd} & \textsc{dt} & \_\_ & & \\
y(2,3) & = & & \_\_ & \textsc{vbd} & \textsc{dt} & \_\_ & & \\
c(1,4) & = & \# & \_\_ & \_\_ & \_\_ & \_\_ & \# & \text{// Constituent contexts} \\
c(2,4) & = & & \textsc{nnp} & \_\_ & \_\_ & \_\_ & \# & \\
c(3,4) & = & & \_\_ & \textsc{vbd} & \_\_ & \_\_ & \# & \\
c(1,2) & = & \# & \_\_ & \_\_ & \textsc{dt} & \_\_ & & \text{// Distituent contexts} \\
c(1,3) & = & \# & \_\_ & \_\_ & \_\_ & \textsc{nn} & & \\
c(2,3) & = & & \textsc{nnp} & \_\_ & \_\_ & \textsc{nn} & &
\end{array}$$

These variables are all consistent with one another, and taken together, provide a complete labeling of tree T:

$$[\textsc{nnp}\ [\textsc{vbd}\ [\textsc{dt}\ \textsc{nn}]]],$$

which corresponds to the correct parse of our original sentence:

$$[\textit{John}\ [\textit{climbed}\ [\textit{the tree}]]].$$

According to the CCM, the probability of the generated variables is simply a product of independent multinomials:

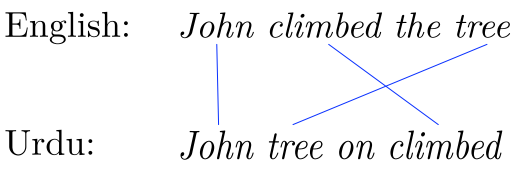
$$P(T) \prod_{(i,j)\in T} \pi^C[y(i,j)]\ \phi^C[c(i,j)] \prod_{(i,j)\notin T} \pi^D[y(i,j)]\ \phi^D[c(i,j)]$$

Under this model, it is obvious that non-zero probability will also be assigned to sets of variables which do *not* yield a consistent part-of-speech sequence. Thus, as a generative model of sentences / part-of-speech sequences, CCM is deficient. Alternatively, we can view CCM as assigning zero probability to inconsistent sets of generated variables by fiat. On this view, probabilities over consistent sets of variables must then be renormalized by some global constant.

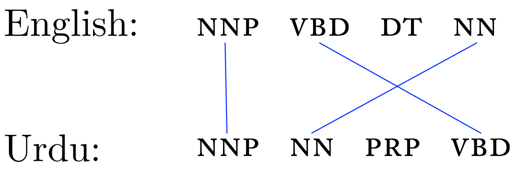
Despite this deficiency, the unsupervised performance of the CCM on English Wall Street Journal text is far higher than that of an unsupervised Probabilistic Context-free Grammar (PCFG) [64]. In fact, the CCM is still among the best-performing unsupervised probabilistic constituency parsers reported in the literature. As such, we use it as the basis of our bilingual model, which we describe in the next section.

3.4.3 Extension to Bilingual Setting

In the bilingual setting we assume that our corpus consists of translated sentence-pairs, along with word-level alignments. For example, we might observe the English sentence discussed in the previous section, but this time with an Urdu counterpart (glossed into English for convenience): [1]



As was the case for the CCM, we disregard the words, and treat the sentence as a pair of partially aligned part-of-speech sequences:



As in the monolingual case, our goal is to model these sequences as arising from latent tree structures, T_1 for the English sentence, and T_2 for the Urdu sentence. We assume that, just as the *words* of the two sentences are aligned, so too are the underlying trees. In fact, we assume that the word-level alignments are themselves a probabilistic byproduct of the tree alignment. See section 3.4.1 above for an overview of the tree alignment formalism which we use.

[1] In the Urdu sentence *tree on* is a postpositional phrase and is the object of the verb *climbed*.

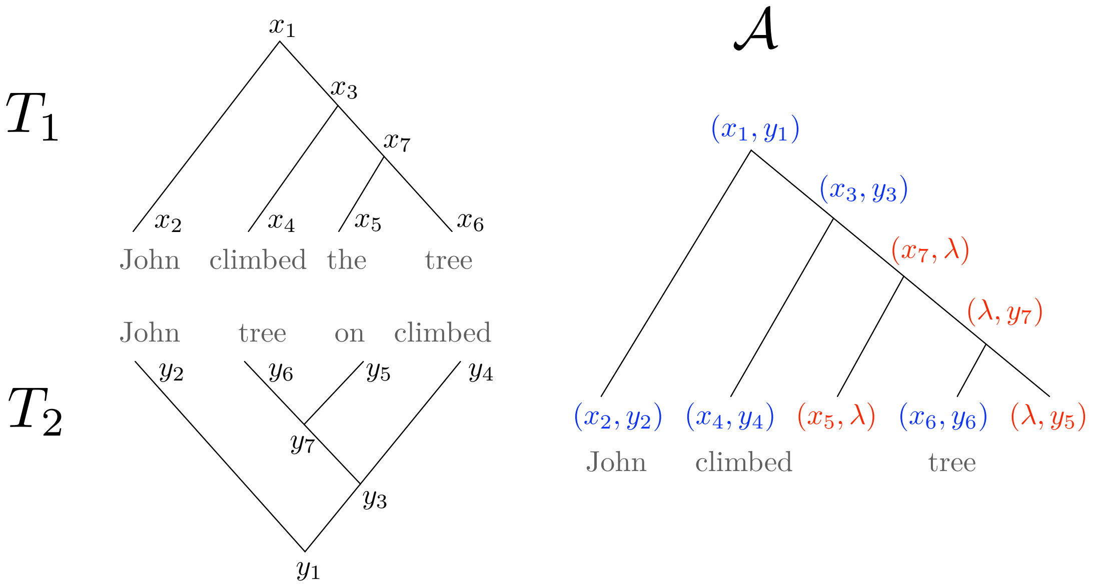


Figure 3-2: Tree pair T_1, T_2 with tree alignment $\mathcal{A}$.

Aligned Trees

Formally, we assume an underlying triple $(T_1, T_2, \mathcal{A})$, where $\mathcal{A}$ is the *alignment tree* between T_1 and T_2. Recall that every node in $\mathcal{A}$ consists of a pair (x, y), where x is either a node in T_1 or the empty symbol λ, and likewise y is either a node in T_2 or λ. Intuitively, two nodes $x \in T_1$ and $y \in T_2$ should be aligned if and only if the respective sentence fragments which they dominate convey more or less the same information. In our example, the underlying aligned trees would be those shown in figure 3.4.3. Note that while five node-pairs are aligned, four nodes remain unaligned: (1) the Urdu postpositional phrase *tree on*, (2) the English definite noun phrase *the tree*, (3) the English word *the*, and (4) the Urdu word *on*. As in the monolingual case, we assume a uniform prior distribution over trees and their alignment.

Bilingual Yields

We now generate yields and contexts for each constituent and distituent in the two trees. We use separate distributions for each language, with one crucial exception:

For aligned node-pairs $(x, y) \in \mathcal{A}$, we draw a bilingual yield-pair from a single joint distribution ω. For example, for the verb phrase node-pair (x_3, y_3) in our running example (figure 3.4.3), we would jointly draw:

$$(\textsc{vbd dt nn}, \quad \textsc{nn prp vbd}) \qquad \sim \quad \omega$$

In all other cases we use language-specific distributions. For example, to generate yields for the two unaligned constituents $x_7 \in T_1$ (*the tree* in English) and $y_7 \in T_2$ (*tree on* in Urdu), we draw:

$$\begin{aligned} \textsc{dt nn} \qquad & \sim \quad \pi_1^C \\ \textsc{nn prp} \qquad & \sim \quad \pi_2^C \end{aligned}$$

And likewise to generate the respective contexts, we draw:

$$\begin{aligned} (\textsc{vbd}, \#) \qquad & \sim \quad \phi_1^C \\ (\textsc{nnp}, \textsc{vbd}) \qquad & \sim \quad \phi_2^C \end{aligned}$$

All *distituent* yields and all contexts are drawn according to the appropriate language-specific distributions.

Word Alignments

Finally, the observed word alignments are generated. Our model views these word alignments as a consequence of the latent tree alignments. In particular, for each aligned node-pair $(x, y) \in \mathcal{A}$, we first generate its *Giza score.* The *Giza score* measures the degree to which the words dominated by nodes x and y are aligned to one another, rather than to words under other nodes.

More precisely, let m be the number of aligned words (w_1, w_2) such that w_1 is dominated by $x \in T_1$ and w_2 is dominated by $y \in T_2$. In our running example, $m = 2$ for aligned node-pair (x_3, y_3) since two words are aligned across those nodes.

Let n be the number of aligned word-pairs (w_1, w_2) such that either (a) w_1 is dominated by x but w_2 is *not* dominated by y, or (b) w_2 is dominated by y but w_1 is *not* dominated by x. In our running example, $n = 0$ for all aligned node-pairs.

Finally, we define the *Giza score* for node-pair (x, y) to simply be $(m-n)$. Higher Giza scores for aligned nodes indicate that the word alignments are relatively more consistent with the node alignment.

For an unaligned node $(x, \lambda) \in \mathcal{A}$, let n be the number of words dominated by x which are aligned to *any* other word. We then define the *Giza score* of (x, λ) to be $0 - n$. Intuitively, words dominated by unaligned nodes should be very sparsely aligned. Thus, ideally the Giza score of unaligned nodes should be zero, or some negative number of low magnitude.

According to our model, the Giza score $s_{(x,y)}$ for each pair of aligned nodes $(x, y) \in \mathcal{A}$ is drawn according to:

$$s_{(x,y)} \quad \sim \quad Gz,$$

where Gz is a discrete distribution over a subset of the integers $\{-K, \ldots, -1, 0, 1, \ldots, K\}$.

The Giza score $s_{(x,\lambda)}$ for *unaligned* node (x, λ) is drawn according to:

$$s_{(x,\lambda)} \quad \sim \quad Gz',$$

where Gz' is discrete distribution over $\{-K, \ldots, -1, 0\}$.[2]

Finally, word alignments consistent with the Giza scores are drawn from a uniform distribution.

In the next two sections, we describe our model more programmatically by listing the parameters and the generative process.

3.4.4 Parameters

In this section we list and describe the parameters of our model, all of which are multinomial distributions:

[2] By definition, the giza score for unaligned nodes cannot be greater than zero.

π_1^C	–	Distribution over constituent-yields of language 1.
π_1^D	–	Distribution over distituent-yields of language 1.
ϕ_1^C	–	Distribution over constituent-contexts of language 1.
ϕ_1^D	–	Distribution over distituent-contexts of language 1.
π_2^C	–	Distribution over constituent-yields of language 2.
π_2^D	–	Distribution over distituent-yields of language 2.
ϕ_2^C	–	Distribution over constituent-contexts of language 2.
ϕ_2^D	–	Distribution over distituent-contexts of language 2.
ω	–	Distribution over bilingual *pairs* of constituent yields.
Gz	–	Distribution over Giza scores: $\{-K, \ldots, -1, 0, 1, \ldots, K\}$.
Gz'	–	Distribution over Giza scores: $\{-K, \ldots, -1, 0\}$.

Briefly, *constituents* are spans of leaves in a tree which are fully and exactly dominated by a node, whereas *distituents* are spans which no single node fully and exactly dominates. *Yields* are labelings of a span of leaves with part-of-speech tags. *Contexts* are labelings of the pair of leaves to the immediate left and right of a span with part-of-speech tags. Fuller descriptions with a running example are given in the previous section.

The first two sets of distributions correspond exactly to the parameters of the CCM. Parameter ω can be thought of as a "coupling parameter" which measures the compatibility of aligned bilingual yield-pairs. The final parameter measures the compatibility of tree alignments with the observed lexical GIZA++ alignments. Intuitively, aligned nodes should have a high density of word-level alignments between them, and unaligned nodes should have few lexical alignments. See the end of the previous section for a formal definition of *Giza score*.

3.4.5 Generative Process

Now we describe the stochastic process whereby the observed parallel sentences and their lexical alignments are generated, according to our model.

We formulate our model in the hierarchical Bayesian framework where the parameters are themselves viewed as random variables. Thus, as the first step in the generative process, all the multinomial parameters listed in the previous section are drawn from their conjugate priors (Dirichlet distributions of appropriate dimension). Once the parameters are drawn, each pair of word-aligned parallel sentences is generated.

Aligned Tree-pair Generation

The first step in sentence generation is to draw a pair of aligned tree structures. We define the prior distribution over these structures to be uniform over all *consistent* triples $(T_1, T_2, \mathcal{A})$, where consistency requires

1. that T_1 and T_2 each be bounded in size (by some very large fixed value),
2. that $\mathcal{A}$ be an alignment tree for T_1 and T_2 (defined in section 3.4.1),
3. and that $\mathcal{A}$ contain no doubly-empty nodes (λ, λ).

Sentence-pair Generation

Given the aligned tree pair $(T_1, T_2, \mathcal{A})$, the sentence generation proceeds as follows:

1. For each unaligned node $(x, \lambda) \in \mathcal{A}$, with $x \in T_1$ dominating span (i, j), draw:

$$\begin{array}{lll} y_1(i,j) & \sim \ \pi_1^C & \text{// constituent-yield for } x \in T_1 \\ c_1(i,j) & \sim \ \phi_1^C & \text{// constituent-context for } x \in T_1 \\ s_{(x,\lambda)} & \sim \ Gz' & \text{// Giza score for } (x, \lambda) \end{array}$$

2. For each unaligned node $(\lambda, y) \in \mathcal{A}$, with $y \in T_2$ dominating span (k, l), draw:

$$\begin{array}{lll} y_2(k,l) & \sim \ \pi_2^C & \text{// constituent-yield for } y \in T_2 \\ c_2(k,l) & \sim \ \phi_2^C & \text{// constituent-context for } y \in T_2 \\ s_{(\lambda,y)} & \sim \ Gz' & \text{// Giza score for } (\lambda, y) \end{array}$$

3. For each aligned node $(x, y) \in \mathcal{A}$, with $x \in T_1$ dominating span (i, j) and $y \in T_2$ dominating span (k, l), draw:

$$\begin{array}{lcll} y_1(i,j),\ y_2(k,l) & \sim & \omega & \text{// constituent-yields for } x \in T_1 \text{ and } y \in T_2 \\ c_1(i,j) & \sim & \phi_1^C & \text{// constituent-context for } x \in T_1 \\ c_2(k,l) & \sim & \phi_2^C & \text{// constituent-context for } y \in T_2 \\ s_{(x,y)} & \sim & Gz & \text{// Giza score for } (x,y) \end{array}$$

4. For each leaf-span $(i, j) \notin T_1$ (i.e. *not* dominated by a node), draw:

$$\begin{array}{lcll} y_1(i,j) & \sim & \pi_1^D & \text{// distituent-yield for } T_1 \\ c_1(i,j) & \sim & \phi_1^D & \text{// distituent-context for } T_1 \end{array}$$

5. For each leaf-span $(k, l) \notin T_2$ (i.e. *not* dominated by a node), draw:

$$\begin{array}{lcll} y_2(k,l) & \sim & \pi_2^D & \text{// distituent-yield for } T_2 \\ c_2(k,l) & \sim & \phi_2^D & \text{// distituent-context for } T_2 \end{array}$$

6. Assemble sentences pair from the yields and contexts
7. Draw lexical alignments consistent with the Giza scores, according to a uniform distribution.

In the next section we turn to the problem of inference under this model when only the part-of-speech tag sequences of parallel sentences and their word alignments are observed.

3.5 Inference

The goal of our inference procedure is to obtain CCM parameters for each language that can be applied to monolingual test data. Ideally, we would choose parameters that have the highest marginal probability, conditioned on the observed bilingual part-of-speech sequences $\mathbf{s}_1, \mathbf{s}_2$ and word alignments $\mathbf{a}$:

$$\hat{\pi}, \hat{\phi} = \underset{\pi,\phi}{\operatorname{argmax}} \int P(\pi, \phi, \omega, Gz, Gz'\mathbf{T}_1, \mathbf{T}_2, \boldsymbol{\mathcal{A}} \mid \mathbf{s}_1, \mathbf{s}_2, \mathbf{a})\; d\omega\, dGz\, dGz'\, d\mathbf{T}_1\, d\mathbf{T}_2\, d\boldsymbol{\mathcal{A}}, \tag{3.1}$$

where $\hat{\pi} = (\pi_1^C, \pi_1^D, \pi_2^C, \pi_2^D)$ is the set of *yield* parameters, $\hat{\phi} = (\phi_1^C, \phi_1^D, \phi_2^C, \phi_2^D)$ is the set of *context* parameters, and $\mathbf{T}_1, \mathbf{T}_2, \boldsymbol{\mathcal{A}}$ are the sets of trees and their alignments over the observed sentences.

While the structure of our model permits us to decompose the joint probability, it is not possible to analytically marginalize all of the hidden variables. We resort to standard Monte Carlo approximation, in which an integral is approximated by a finite sum. In particular, we sample posterior values of the hidden aligned trees: $\mathbf{T}_1, \mathbf{T}_2, \boldsymbol{\mathcal{A}}$, and replace their integral in equation 3.1 with a sum over the samples. As the number of samples goes to infinity, the approximation converges to the true value of the integral [80].

Since simultaneously sampling latent trees for all sentence-pairs is not feasible, we use Gibbs sampling to draw individual variables one at a time [40]. Gibbs sampling begins by randomly initializing unobserved random variables; at each iteration, each random variable u_i is then sampled from the conditional distribution $P(u_i|u_{-i})$, where u_{-i} refers to all variables $u_{j\neq i}$. By repeatedly sampling individual hidden variables according to their conditional distributions, we obtain a Markov chain whose stationary distribution is the desired joint distribution over the variables $P(\mathbf{u})$ [42]. When possible, we avoid explicitly sampling variables which are not of direct interest, but rather integrate over them. This technique is known as *collapsed sampling*; it is guaranteed never to increase sampling variance, and will often reduce it [74].

In particular, for each sentence pair (s_1, s_2) with word alignment a, we sample an aligned tree-pair $(T_1, T_2, \mathcal{A})$. To do so, we perform a Metropolis-within-Gibbs sampling step: The trees (T_1, T_2) are first sampled from a simpler *proposal distribution* and then are accepted or rejected on the basis of their true marginal probability. Only afterwards is the alignment $\mathcal{A}$ between them sampled.

Algorithm 3: Gibbs sampler for bilingual grammar induction.

```
Input: Bilingual corpus consisting of part-of-speech sequence-pairs $(s_1, s_2)$
       with corresponding lexical alignments $a$
Output: 1000 samples of aligned tree-pairs $(T_1, T_2, \mathcal{A})$ for each sentence pair.
Initialize aligned tree-pairs $(T_1, T_2, \mathcal{A})$;
for $r \leftarrow 1$ to 1000 do
    for word-aligned sentence pair $(s_1, s_2, a)$ do
        Sample tree-pair $(T_1, T_2)^*$ from proposal distribution $Q$ // Section 3.5.1
        Sample Bernoulli $b$ according to acceptance ratio          // Section 3.5.2
        if $b = 1$ then
            $(T_1, T_2)^{(r)} \leftarrow (T_1, T_2)^*$
        else
            $(T_1, T_2)^{(r)} \leftarrow (T_1, T_2)^{(r-1)}$
        Sample tree alignment $\mathcal{A}^{(r)}$ for $(T_1, T_2)^{(r)}$              // Section 3.5.3
```

Throughout sampling, we marginalize out the the parameters $(\pi, \phi, \omega, Gz, Gz')$, using standard closed-form integrals. Algorithm 3 gives an overview of our sampling algorithm. In the remainder of the section we describe the individual sampling steps.

3.5.1 Sampling Trees

For the i^{th} word-aligned sentence pair (s_1, s_2, a) we wish to sample an aligned tree-pair $(T_1, T_2, \mathcal{A})$ from:

$$P(T_1, T_2, \mathcal{A} \mid s_1, s_2, a, (s_1, s_2, a, T_1, T_2, \mathcal{A})_{-i}),$$

where the notation x_{-i} refers to all instances of x besides x_i. Since an exponential number of aligned tree-pairs are possible for each instance, our sampling algorithm needs to factor into a series of smaller moves. We know of no simple factorization for aligned tree pairs. We therefore develop a Metropolis-Hastings sampler [54] which allows us to first sample the trees themselves, and only afterwards sample the alignment between them.

In general, Metropolis-Hastings is used when sampling from a posterior $P(u_i|u_{-i})$

is difficult. Instead of directly sampling a new $u_i^{(r)}$ at round r, one instead samples a new value u_i^* from a simpler *proposal distribution* $Q(u_i^* \mid u_i', u_{-i})$, where u_i' denotes the previously sampled value for u_i (shorthand for $u_i^{(r-1)}$). In its simplest form, the proposal distribution is often a Gaussian with mean set to u_i'. After the proposed value u_i^* is drawn, an *acceptance ratio* is computed:

$$\alpha = \frac{P(u_i^* \mid u_{-i})\ Q(u_i' \mid u_i^*, u_{-i})}{P(u_i' \mid u_{-i})\ Q(u_i^* \mid u_i', u_{-i})}$$

A random value b is then drawn from a Bernoulli with parameter $\min(\alpha, 1)$. If $b = 1$, we accept the proposed value: $u_i^{(r)} \leftarrow u_i^*$. Otherwise, we retain our previous sample: $u_i^{(r)} \leftarrow u_i'$. The Markov chain induced by this sampler will eventually converge to the true posterior $P(u_i \mid u_{-i})$ [54]. However, using a "random walk" proposal distribution can often lead to very slow mixing of the Markov chain.

If the proposal distribution Q does *not* depend on the previous value u_i', then we say that this is an *independent* Metropolis-Hastings sampler. If Q is a good (if biased) approximation to the posterior, then we can avoid the slow mixing behavior of the random walk.

We develop an independent Metropolis-Hastings sampler with a proposal distribution that treats each language as fully independent of the other:

$$\begin{aligned} &Q(T_1, T_2 \mid s_1, s_2, (s_1, s_2, T_1, T_2)_{-i}) = \\ &\qquad Q_1(T_1 \mid s_1, (s_1, T_1)_{-i}) \cdot Q_2(T_2 \mid s_2, (s_2, T_2)_{-i}) \end{aligned}$$

Each distribution Q_ℓ is what the posterior for tree T_ℓ *would have been* if (i) the tree alignments $\boldsymbol{\mathcal{A}}$ *only* contained unaligned nodes, and (ii) word alignments $\mathbf{a}$ were all empty. In other words, our proposal distribution mimics the monolingual CCM and ignores any explicit cross-lingual information.[3]

To sample from this proposal distribution we build on a well-known tree sampling algorithm for PCFGs [44, 62]. Our algorithm proceeds in two steps. First, we

[3] Cross-lingual information still exercises implicit influence on Q by way of the sampled values for other trees $(T_1)_{-i}$ and $(T_2)_{-i}$.

compute the marginal probability of each span in the sentence, using a dynamic program which sums over all possible subtrees that dominate the given span. The resulting table is similar to the "inside" table of the inside-outside algorithm for PCFGs [71]. Using this table, we proceed to sample the tree in top-down fashion by recursively sampling individual split points in the sentence.

More formally, consider a sentence $s = w_1, \ldots, w_n$. As before, we use abbreviations to denote the yield and context of each span:

$$\begin{aligned} y(i,j) &= w_i, \ldots, w_j \\ c(i,j) &= (w_{i-1}, w_{j+1}) \end{aligned}$$

Recall that we can write the monolingual CCM probability $P(s,T)$ as a product of constituent and distituent parameters:

$$P(T) \prod_{(i,j)\in T} \pi^C\big[y(i,j)\big]\ \phi^C\big[c(i,j)\big] \prod_{(i,j)\notin T} \pi^D\big[y(i,j)\big]\ \phi^D\big[c(i,j)\big] \tag{3.2}$$

Following Klein [63] (Appendix A.1), we rewrite this probability so that it factors over constituent spans (i.e. nodes of T):

$$K(s) \prod_{(i,j)\in T} \beta(i,j), \tag{3.3}$$

where $\beta(i,j)$ is a fraction of constituent parameters over distituent parameters for span (i,j):

$$\beta(i,j) = \frac{\pi^C\big[y(i,j)\big]\ \phi^C\big[c(i,j)\big]}{\pi^D\big[y(i,j)\big]\ \phi^D\big[c(i,j)\big]}, \tag{3.4}$$

and $K(s)$ is a sentence-specific constant (distituent parameters over all spans and the constant tree probability):

$$K(s) = P(T) \prod_{0<i\leq j\leq n} \pi^D\big[y(i,j)\big]\ \phi^D\big[c(i,j)\big] \tag{3.5}$$

We then define the *inside score* of span (i,j) to be its unnormalized CCM

Algorithm 4: `split`(i, j) recursively samples a binary tree over span (i, j)

```
if $i = j$ then return $\{(i, j)\}$
for $k \leftarrow i$ to $j - 1$ do
    $p_k \leftarrow \mathrm{I}(i, k)\ \mathrm{I}(k + 1, j)$
$z \leftarrow \sum_{i \leq k < j} p_k$
sample $k \sim \text{discrete}[p_i/z, \ldots, p_{j-1}/z]$
return $\{(i, k), (k + 1, j)\} \cup$ split$(i, k)$ $\cup$ split$(k + 1, j)$
```

marginal probability:

$$\mathrm{I}(i, j) = \sum_{T'} \prod_{(a,b) \in T'} \beta(a, b),$$

where the sum is over all binary tree structures T' with $(j - i)$ leaves. These values can be computed recursively in $O(n^3)$ time by summing over all split-points of each span:

$$I(i, j) \quad \leftarrow \quad \beta(i, j) \sum_{i \leq k < j} \mathrm{I}(i, k)\ \mathrm{I}(k + 1, j)$$

Once the table $\mathrm{I}(i, j)$ has been computed, we can sample a tree T by making top-down split decisions over the sentence. Algorithm 4 defines the recursive function `split`. To sample a complete binary tree for sentence $s = w_1, \ldots, w_n$ we call `split`$(1, n)$.

To see that this function samples T according to $P(T|s)$, we cast each tree as a unique series of split decisions $T = d_1, d_2, \ldots, d_m$. For each decision d_r, we deterministically select a span (i, j) from a set of available spans S_r.[4] We then choose some split-point $d_r = k \in \{i, \ldots, j - 1\}$ and update the available spans: $S_{r+1} \leftarrow S_r - \{(i, j)\} \cup \{(i, k), (k + 1, j)\}$. S_1 is initialized with the full sentence span $(1, n)$, and the tree is complete when the set S_r contains only singleton spans (i, i).

[4] For example, by lexicographically ordering S_r and choosing the minimal element.

We can now rewrite the tree probability in terms of split decisions:

$$P(T,s) \propto P(T|s) = P(d_1,\ldots,d_m \mid s) = \prod_{r=1}^{m} P(d_r \mid d_1,\ldots,d_{r-1},s)$$

Assume that d_r is a decision to split span (i,j) into (i,k) and $(k+1,j)$. Then we can marginalize over all possible completions of the decision process:

$$\begin{aligned} P(d_r \mid d_1,\ldots,d_{r-1},s) &= \sum_{d_{r+1},\ldots,d_m} P(d_r,\ldots,d_m \mid d_1,\ldots,d_{r-1},s) \\ &\propto \sum_{T' \text{ over } (i,k)} \prod_{(a,b)\in T'} \beta(a,b) \sum_{T'' \text{ over } (k+1,j)} \prod_{(a,b)\in T''} \beta(a,b) \\ &= \mathrm{I}(i,k)\ \mathrm{I}(k+1,j) \end{aligned}$$

To conclude the section, we note that definition 3.4 depends on the multinomial parameter values π and ϕ. As mentioned earlier, we avoid explicitly sampling these parameters, instead marginalizing them out. Since we use (conjugate) Dirichlet priors, we can employ the standard closed-forms for the posteriors. For example, we can write the posterior of a constituent-yield c as:

$$P(c \mid \ldots) = \frac{N(c)+\alpha_0}{\sum_{c'} N(c')+\alpha_0},$$

where the ellipsis "$\ldots$" denotes the other sampled constituent-yields, $N(c)$ denotes the number of times c appears among them, and α_0 is the symmetric Dirichlet hyperparameter. Intuitively, we can think of the hyperparameter α_0 as a smoothing pseudo-count for infrequently observed constituent yields. See section 3.6.2 for the hyperparameter values used in our experiments.

3.5.2 Computing Acceptance Ratios

After a new tree pair has been sampled from our proposal distribution Q, we need to compute an *acceptance ratio*. This ratio compares the true posterior and the proposal probabilities of the new pair $(T_1^*, T_2^*)_i$ and the previously sampled $(T_1', T_2')_i$:[5]

$$\alpha = \frac{P(T_1^*, T_2^* \mid s_1, s_2, a)\; Q(T_1', T_2' \mid s_1, s_2)}{P(T_1', T_2' \mid s_1, s_2, a)\; Q(T_1^*, T_2^* \mid s_1, s_2)}$$

Recall that our proposal distribution Q decomposes into separate monolingual CCM probabilities for each tree (equation 3.5.1). Thus, we can easily compute $Q(T_1, T_2|s_1, s_2)$ as a product of multinomial posteriors (with the parameters marginalized out). However, to compute the true model posterior for trees (T_1, T_2), we must marginalize over all possible tree alignments:

$$\sum_{\mathcal{A}} P(T_1, T_2, \mathcal{A} \mid s_1, s_2, a)$$

Fortunately, for any given pair of trees T_1 and T_2 this marginalization can be computed using a dynamic program in time $O(|T_1||T_2|)$. Here we provide a very brief sketch. For every pair of nodes $x \in T_1, y \in T_2$, a table stores the marginal probability of the subtrees rooted at x and y, respectively. A dynamic program builds this table from the bottom up: For each node pair x, y, we sum the probabilities of all local alignment configurations, each multiplied by the appropriate marginals already computed in the table for lower-level node pairs. This algorithm is an adaptation of the dynamic program presented in [60] for finding minimum cost alignment trees (Fig. 5 of that publication).

3.5.3 Sampling Tree Alignments

Once a pair of trees (T_1, T_2) has been sampled, we can proceed to sample an alignment tree $\mathcal{A}|T_1, T_2$.[6] We sample individual alignment decisions from the top down, at each step using the alignment marginals for the remaining subtrees (already computed using the dynamic program sketched in the previous section). Once the triple

[5] The conditional dependence on the other sampled aligned trees $(T_1, T_2, \mathcal{A})_{-i}$ has been suppressed for notational convenience.

[6] Sampling the alignment tree is important, as it provides us with counts of aligned constituents for the coupling parameter.

$(T_1, T_2, \mathcal{A})$ has been sampled, we move on to the next parallel sentence.

3.6 Experiments

We test our model on three corpora of bilingual parallel sentences: English-Korean, English-Urdu, and English-Chinese. Though the model is trained using parallel data, during testing it has access only to monolingual data. This set-up ensures that we are testing our model's ability to learn better parameters at training time, rather than its ability to exploit parallel data at test time. Following [64], we restrict our model to binary trees, though we note that the alignment trees do not follow this restriction.

3.6.1 Data and Baseline

The Penn Korean Treebank [52] consists of 5,083 Korean sentences translated into English for the purposes of language training in a military setting. Both the Korean and English sentences are annotated with syntactic trees. We use the first 4,000 sentences for training and the last 1,083 sentences for testing. We note that in the Korean data, a separate tag is given for each morpheme. We simply concatenate all the morpheme tags given for each word and treat the concatenation as a single tag. This procedure results in 199 different tags. The English-Urdu parallel corpus[7] consists of 4,325 sentences from the first three sections of the Penn Treebank and their Urdu translations annotated at the part-of-speech level. The Urdu side of this corpus does not provide tree annotations so here we can test parse accuracy only on English. We use the remaining sections of the Penn Treebank for English testing. The English-Chinese treebank [10] consists of 3,850 Chinese newswire sentences translated into English. Both the English and Chinese sentences are annotated with parse trees. We use the first 4/5 for training and the final 1/5 for testing.

During preprocessing of the corpora we remove all punctuation marks and spe-

[7] `http://www.crulp.org`

cial symbols, following the setup in previous grammar induction work [64]. To obtain lexical alignments between the parallel sentences we employ GIZA++ [90]. We use intersection alignments, which are one-to-one alignments produced by taking the intersection of one-to-many alignments in each direction. These one-to-one intersection alignments tend to have higher precision.

We initialize the trees by making uniform split decisions recursively from the top down for sentences in both languages. Then for each pair of parallel sentences we randomly sample an initial alignment tree for the two sampled trees.

We implement a Bayesian version of the CCM as a baseline. This model uses the same inference procedure as our bilingual model (Gibbs sampling). In fact, our model reduces to this Bayesian CCM when it is assumed that no nodes between the two parallel trees are ever aligned and when word-level alignments are ignored. We also reimplemented the original EM version of CCM and found virtually no difference in performance when using EM or Gibbs sampling. In both cases our implementation achieves F-measure in the range of 69-70% on WSJ10, broadly in line with the performance reported by [64].

3.6.2 Hyperparameters

Klein [63] reports using smoothing pseudo-counts of 2 for constituent yields and contexts and 8 for distituent yields and contexts. In our Bayesian model, these similar smoothing counts occur as the parameters of the Dirichlet priors. For Korean we found that the baseline performed well using these values. However, on our English and Chinese data, we found that somewhat higher smoothing values worked best, so we utilized values of 20 and 80 for constituent and distituent smoothing counts, respectively.

Our model additionally requires hyperparameter values for:

- ω – The coupling distribution for aligned constituent yields
- Gz – The distribution over giza scores for aligned nodes
- Gz' – The distributions over giza scores for unaligned nodes

For ω we used a symmetric Dirichlet prior with parameter 1. Recall that both Gz and Gz' are distributions over Giza scores, which respectively range over of the sets $\{-K, \ldots, -1, 0, 1, \ldots, K\}$ and $\{-K, \ldots, -1, 0\}$. In our experiments, we set $K = 3$.

In order to create a strong inductive bias towards high Giza scores, we used non-symmetric Dirichlet priors for these distributions. In the case of Gz (giza score for aligned nodes), we set the hyperparameters to 1,000 for negative values and zero, and 1,000,000 for positive values. In the case of Gz', we set the hyperparameters to 1,000 for negative scores and 1,000,000 for zero itself. This very strong prior bias encodes our intuition that syntactic alignments which respect lexical alignments should be preferred. Our method is not sensitive to these exact values and any reasonably strong bias gave similar results.

In all our experiments, we consider the hyperparameters to be fixed and observed values.

3.6.3 Results

As mentioned previously, we test our model only on monolingual data, where the parallel sentences are not provided to the model. To predict the bracketings of these monolingual test sentences, we take the counts accumulated in the final round of sampling over the training data and perform a maximum likelihood estimate of the monolingual CCM parameters. These parameters are then used to produce the highest probability bracketing of the test set.

To evaluate both our model as well as the baseline, we use (unlabeled) bracket precision, recall, and F-measure [64]. More formally, we consider both the gold-standard tree and the predicted tree to be sets of constituent spans. Thus, for the sentence *John climbed the tree*, the gold-standard tree for the correct bracketing:

[John [climbed [the tree]]]

would be $T^* = \{(1, 4), (2, 4), (3, 4)\}$. The tree for the incorrect bracketing:

[[John climbed] the tree]

	Max Sent. Length		Monolingual			Bilingual			Upper Bound
	Test	Train	Precision	Recall	F1	Precision	Recall	F1	F1
EN with KR	10	10	52.74	39.53	45.19	57.76	43.30	49.50	85.6
		20	41.87	31.38	35.87	61.66	46.22	52.83	85.6
		30	33.43	25.06	28.65	64.41	48.28	**55.19**	85.6
	20	20	35.12	25.12	29.29	56.96	40.74	47.50	83.3
		30	26.26	18.78	21.90	60.07	42.96	**50.09**	83.3
	30	30	23.95	16.81	19.76	58.01	40.73	**47.86**	82.4
KR with EN	10	10	71.07	62.55	66.54	75.63	66.56	70.81	93.6
		20	71.35	62.79	66.80	77.61	68.30	72.66	93.6
		30	71.37	62.81	66.82	77.87	68.53	**72.91**	93.6
	20	20	64.28	54.73	59.12	70.44	59.98	64.79	91.9
		30	64.29	54.75	59.14	70.81	60.30	**65.13**	91.9
	30	30	63.63	54.17	58.52	70.11	59.70	**64.49**	91.9
EN with CH	10	10	50.09	34.18	40.63	37.46	25.56	30.39	81.0
		20	58.86	40.17	47.75	50.24	34.29	40.76	81.0
		30	64.81	44.22	52.57	68.24	46.57	**55.36**	81.0
	20	20	41.90	30.52	35.31	38.64	28.15	32.57	84.3
		30	52.83	38.49	44.53	58.50	42.62	**49.31**	84.3
	30	30	46.35	33.67	39.00	51.40	37.33	**43.25**	84.1
CH with EN	10	10	39.87	27.71	32.69	40.62	28.23	33.31	81.9
		20	43.44	30.19	35.62	47.54	33.03	38.98	81.9
		30	43.63	30.32	35.77	54.09	37.59	**44.36**	81.9
	20	20	29.80	23.46	26.25	36.93	29.07	32.53	88.0
		30	30.05	23.65	26.47	43.99	34.63	**38.75**	88.0
	30	30	24.46	19.41	21.64	39.61	31.43	**35.05**	88.4
EN with UR	10	10	57.98	45.68	51.10	73.43	57.85	64.71	88.1
		20	70.57	55.60	62.20	80.24	63.22	**70.72**	88.1
		30	75.39	59.40	66.45	79.04	62.28	69.67	88.1
	20	20	57.78	43.86	49.87	67.26	51.06	**58.05**	86.3
		30	63.12	47.91	54.47	64.45	48.92	55.62	86.3
	30	30	57.36	43.02	49.17	57.97	43.48	**49.69**	85.7

Table 3.1: Unlabeled precision, recall and F-measure for the monolingual baseline and the bilingual model on several test sets. We report results for different combinations of maximum sentence length in both the training and test sets. The right most column, in all cases, contains the maximum F-measure achievable using binary trees. The best performance for each test-length is highlighted in bold.

would be $T = \{(1,4),(1,2)\}$. Following previous work, we include the whole-sentence brackets but ignore single-word brackets. Thus, in this example precision would be 1/2 and recall would be 1/4.

Klein [64] notes that the CCM performance drops precipitously on long sentences. In order to compare monolingual and bilingual results across different sentence lengths, we consider various corpus subsets. In particular, for each corpus we extract subsets with maximum sentence lengths of 10, 20, and 30 for both the training and testing portions.

For each corpus, we then train and test both the CCM and our bilingual model on each of the sub-corpora (i.e. sentences with maximum length 10, 20, and 30). We also consider the scenario where each of the two models is trained on longer sentences but only tested on shorter sentences. For example, we would train the models on sentences up to length 30, but only test on sentences up to length 10 or length 20. We average all results over 10 separate sampling runs.

We report the upper bound on F-measure obtainable by binary trees. To do so, we binarize the gold-standard trees and compute the precision of the resulting constituents (recall remains at 100%).

Table 3.1 gives the full results of our experiments. In all testing scenarios the bilingual model outperforms its monolingual counterpart in terms of both precision and recall. On average across all scenarios, the bilingual model gains 10.2 percentage points in precision, 7.7 in recall, and 8.8 in F-measure. The gap between monolingual performance and the binary tree upper bound is reduced by over 19%.

The extent of the gain varies across pairings. For instance, the smallest improvement is observed for English when trained with Urdu. The Korean-English pairing results in substantial improvements for Korean and quite large improvements for English, for which the absolute gain reaches 28 points in F-measure. In the case of Chinese and English, the gains for English are fairly minimal whereas those for Chinese are quite substantial. This asymmetry should not be surprising, as Chinese on its own seems to be quite a bit more difficult to parse than English.

We investigate the impact of sentence length for both the training and testing

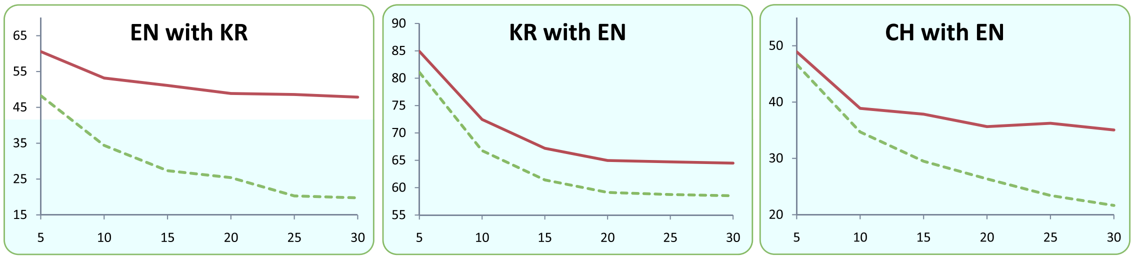


Figure 3-3: The F-measure of the CCM baseline (dotted line) and bilingual model (solid line) plotted on the y-axis, as the maximum sentence length in the test set is increased (x-axis). Results are averaged over all training scenarios given in Table 3.1.

sets. For our model, adding sentences of greater length to the training set always leads to increases in parse accuracy for short sentences. For the baseline, however, adding this additional training data significantly degrades performance in the case of English paired with Korean.

Figure 3-3 summarizes the performance of our model for different sentence lengths on several of the test-sets. As shown in the figure, the largest improvements over the baseline tend to occur at longer sentence lengths.

3.7 Conclusions

In this chapter we presented a probabilistic model for bilingual grammar induction. The key challenge we confronted was in finding ways to represent cross-lingual regularities in syntactic patterns while still allowing significant language-specific divergence.

We addressed this challenge by adapting a computational formalism known as *unordered tree alignment* [60] to a Bayesian probabilistic setting. Under this formalism, any two trees can be embedded in an *alignment tree*. The alignment tree allows arbitrary parts of the two trees to diverge in structure, permitting language-specific grammatical structure to be preserved.

We found the computational properties of this formalism to be a major advantage. Tree alignments must remain monotonic in the hierarchical structure of the aligned trees. As a result, we could use dynamic programming to efficiently marginalize over alignments as well as sample them, both in polynomial time in the size of the trees.

We built our probabilistic model on the basis of the Constituent-Context Model of Klein and Manning [64]. Although this model gives state-of-the-art results for English grammar induction, it is formulated as a deficient model which overgenerates the observed data. Unfortunately, our bilingual formulation inherits this deficiency.

Experimentally, we saw significant improvements over the monolingual baseline across three language pairings and a large range of experimental settings. Although this is encouraging, performance still remains very low in most instances. On the bright side, there remains much room for improvement on this most difficult of tasks.

Chapter 4

Lost Language Decipherment

In the previous two chapters, we examined the classical NLP tasks of part-of-speech tagging and grammar induction. We showed that multilingual analysis leads to significant performance gains over monolingual models. However, in both cases we assumed the existence of multilingual parallel texts. For most tasks and languages, this is indeed a realistic assumption. In contrast, this chapter examines a problem for which parallel text is typically not available: lost language decipherment. Instead of using parallel text to induce cross-lingual regularities, we instead look for *language-wide* structural similarities between the lost language and a living relative.

We make several assumptions in our approach to this task. Crucially, we assume that a known related language has already been identified (or hypothesized). Another assumption is that the writing systems of both languages are more or less *alphabetic* in nature.[1] Because the languages are related, we can expect a stable mapping between their letters to exist, due to the presence of *cognates*. Cognates are pairs of words which descend from a common word in a shared ancestral language.

Our definition of the computational decipherment task closely follows the setup typically faced by human decipherers [100]. Our input consists of texts in a lost language and a corpus of non-parallel data in the known related language. The

[1] To be more precise, the two writing systems dealt with here are of the *abjad* type, as vowels are not fully represented. For syllabic and logographic systems, modifications to our approach may be required.

decipherment itself involves two related sub-tasks:

1. finding the mapping between alphabets of the known and lost languages, and
2. translating words in the lost language into corresponding cognates of the known language.

4.1 Chapter Overview

Section 4.2 gives a broad introduction to the chapter. We list some of the intuitions that have guided human decipherers. We argue that language-wide similarities between the known and lost language can be automatically discovered by encoding these intuitions in a probabilistic model. We give a summary of the model and of our experimental results. Section 4.3 describes previous work related to language decipherment. Section 4.4 gives some background information on the discovery and decipherment of the Ugaritic language. Section 4.5 considers some assumptions made in our formulation of the task. Section 4.6 fully describes our model, and section 4.7 details our inference algorithm. Section 4.8 gives our experiments and results, and section 4.10 closes the chapter with some discussion and directions for future work.

4.2 Introduction

Dozens of lost languages have been deciphered by humans in the last two centuries. In each case, the decipherment has been considered a major intellectual breakthrough, often the culmination of decades of scholarly efforts. Computers have played no role in the decipherment any of these languages. Andrew Robinson, a noted author on writing systems and lost languages, represents the skeptical scholarly view that computers do not possess the "logic and intuition" required to unravel the mysteries of ancient scripts.[2] In this chapter, we demonstrate that at least some

[2] *"Successful archaeological decipherment has turned out to require a synthesis of logic and intuition …that computers do not (and presumably cannot) possess."* [100]

of this logic and intuition can be successfully modeled, allowing computational tools to be used in the decipherment process.

While there is no single formula that human decipherers have employed, manual efforts have focused on several guiding principles. A common starting point is to compare letter and word frequencies between the lost and known languages. In the presence of cognates the correct mapping between the languages will reveal similarities in frequency, both at the character and lexical level. In addition, morphological analysis plays a crucial role here, as highly frequent morpheme correspondences can be particularly revealing. In fact, these three strands of analysis (character frequency, morphology, and lexical frequency) are intertwined throughout the human decipherment process. Partial knowledge of each drives discovery in the others.

We capture these intuitions in a generative Bayesian model. This model assumes that words in the lost language are composed of morphemes which were generated with latent counterparts in the known language. We model bilingual morpheme pairs as arising through a series of Dirichlet processes. This allows us to assign probabilities based both on character-level correspondences (using a character-edit base distribution) as well as higher-level morpheme correspondences. In addition, our model carries out an implicit morphological analysis of the lost language, utilizing the known morphological structure of the related language. This model structure allows us to capture the interplay between the character- and morpheme-level correspondences that humans have used in the manual decipherment process.

In addition, we introduce a novel technique for imposing structural sparsity constraints on character-level mappings. We assume that an accurate alphabetic mapping between related languages will be sparse in the following way: each letter will map to a very limited subset of letters in the other language. We capture this intuition by adapting the so-called “spike-and-slab” prior to the Dirichlet-multinomial setting. For each pair of characters in the two languages, we posit an indicator variable which controls the prior likelihood of those characters substituting for one another. We define a joint prior over these indicator variables which encourages sparse settings.

We applied our model to a corpus of Ugaritic, an ancient Semitic language discovered in 1928. Ugaritic was manually deciphered in 1932, using knowledge of Hebrew, a related language. We compare our method against the only existing decipherment baseline, an HMM-based character substitution cipher [65, 66]. The baseline correctly maps the majority of letters — 23 out of 30 — to their correct Hebrew counterparts, but only correctly translates 29% of all cognates. In comparison, our method yields correct mappings for 28 of 30 letters, and correctly translates 63% of all cognates into their Hebrew counterparts.

4.3 Related Work

Our work on decipherment has connections to several lines of work in statistical NLP. First, our work relates to research on automatic cognate identification. Early work on this task assumed the existence of bilingual dictionaries and a complete table of sound correspondences [75, 46]. The goal was to predict whether a given pair of words, which are known to be a translation pair, had descended from a common ancestral word. Kondrak [69] extended this line of work by removing the assumption that cognates must have identical meanings. He instead measures semantic similarity using glosses from a dictionary. Bergsma and Kondrak [7] consider an extension beyond language pairs to include evidence from multiple languages. They employ an integer linear programming framework to globally constrain cognate decisions by a large set of languages. More recently, Hall and Klein [51] presented a Bayesian model for cognate induction from unaligned word lists. They model the latent phylogenetic structure of the languages in order to induce more accurate cognate predictions. They assume that the languages in question share a single writing system, and that all the words given have at least one cognate.

In contrast to this line of work, we do not assume access to any sort of dictionary for our lost language, nor do we know the phonetic values of the symbols. In fact, we deal with two entirely distinct writing systems and actual archeological texts rather than artificial word lists.

A second related line of work is lexicon induction from non-parallel corpora. While this research has similar goals, it typically builds on information or resources unavailable for ancient texts, such as comparable corpora, a seed lexicon, and cognate information [37, 96, 68, 48]. Moreover, distributional methods that rely on co-occurrence analysis operate over large corpora, which are typically unavailable for a lost language.

Two recent papers complementary to ours work are Penn and Choma [93] and Bouchard-Cote et al. [12]. In the first paper, Penn and Choma propose a quantitative method for the automatic classification of writing systems. In fact, this chapter assumes throughout that the basic nature of the lost-language writing system is known (i.e. that it is more or less alphabetic in nature – technically an *abjad*). The results presented by Penn and Choma support the plausibility of this assumption. In the second paper, Bouchard-Cote et al present a probabilistic model of diachronic phonology. As input, they assume a list of cognates across several Romance languages, and predict latent ancestral forms. In essence, this work assumes as *input* what our model produces as *output*. Thus, after automatically deducing cognates between Ugaritic and Hebrew, we could theoretically use the model of Bouchard-Cote et al to induce the latent ancestral forms in Proto-Semitic which led to these cognates.

Finally, Knight and Yamada [65] and Knight et al. [66] describe a computational HMM-based method for deciphering an unknown script that represents a known spoken language. This method “makes the text speak” by gleaning character-to-sound mappings from non-parallel character and sound sequences. It does not relate words in different languages, thus it cannot encode deciphering constraints similar to the ones considered in this paper. More importantly, this method had not been applied to archaeological data. While lost languages are gaining increasing interest in the NLP community [67], there have been no successful attempts of their automatic decipherment.

4.4 Background on Ugaritic

In this section we give some background information on the Ugaritic language. We first describe the story of its decipherment, and then briefly list some of its linguistic properties.

4.4.1 Manual Decipherment of Ugaritic

The Ugaritic tablets (dating from the 14th through 12 centuries BCE) were first discovered in Syria in 1928 [105, 120]. At the time of their discovery, the cuneiform writing on the tablets was of an unknown type. Charles Virolleaud, who led the initial decipherment effort, recognized that the script was likely alphabetic, since the inscribed words consisted of only thirty distinct symbols. The location of the tablets discovery further suggested that Ugaritic was likely to have been a Semitic language from the Western branch, with properties similar to Hebrew and Aramaic. This realization was crucial for deciphering the Ugaritic script. In fact, German cryptographer and Semitic scholar Hans Bauer decoded the first two Ugaritic letters—*mem* and *lambda*—by mapping them to Hebrew letters with similar occurrence patterns in prefixes and suffixes. Bootstrapping from this finding, Bauer found words in the tablets that were likely to serve as cognates to Hebrew words—e.g., the Ugaritic word for *king* matches its Hebrew equivalent. Through this process a few more letters were decoded, but the Ugaritic texts were still unreadable. What made the final decipherment possible was a sheer stroke of luck—Bauer guessed that a word inscribed on an ax discovered in the Ras Shamra excavations was the Ugaritic word for *ax*. Bauer's guess was correct, though he selected the wrong phonetic sequence. Edouard Dhorme, another cryptographer and Semitic scholar, later corrected the reading, expanding a set of translated words. Discoveries of additional tablets allowed Bauer, Dhorme and Virolleaud to revise their hypothesis, completing the initial decipherment. Since these initial decipherment results, scholars have spent decades mapping the individual words of the Ugaritic vocabulary to cognates in other Semitic languages. The translation of the Ugaritic tablets remains a lively and controversial field of study.

4.4.2 Linguistic Features of Ugaritic

Ugaritic shares many features with other ancient Semitic languages, following the same word order, gender, number, and case structure [55]. It is a morphologically rich language, with triliteral roots and many prefixes and suffixes.

At the same time, it exhibits a number of features that distinguish it from Hebrew. Ugaritic has a bigger phonemic inventory than Hebrew, yielding a bigger alphabet – 30 letters vs. 23 in Hebrew. Another distinguishing feature of Ugaritic is that vowels are only indicated for diphthongs or when following the glottal stop (through the use of three distinct glottal stop characters) while in Hebrew many long vowels are written using homorganic consonants. Ugaritic also does not have articles, while Hebrew nouns and adjectives take definite articles which are realized as prefixes. These differences result in significant divergence between Hebrew and Ugaritic cognates, thereby complicating the decipherment process.

4.5 Problem Formulation

We are given a corpus in a lost language and a non-parallel corpus in a related language from the same language family. Our primary goal is to translate words in the unknown language by mapping them to cognates in the known language. As part of this process, we induce a lower-level mapping between the letters of the two alphabets, capturing the regular phonetic correspondences found in cognates.

We make several assumptions about the writing system of the lost language. First, we assume that the writing system is alphabetic in nature. In general, this assumption can be easily validated by counting the number of symbols found in the written record. Next, we assume that the corpus has been transcribed into electronic format, where the graphemes present in the physical text have been unambiguously identified. Finally, we assume that words are explicitly separated in the text, either by white space or a special symbol.

We also make a mild assumption about the morphology of the lost language. We posit that each word consists of a stem, prefix, and suffix, where the latter two may be omitted. This assumption captures a wide range of human languages and a variety of morphological systems. While the correct morphological analysis of words in the lost language must be learned, we assume that the inventory and frequencies of prefixes and suffixes in the known language are given.

In summary, the observed input to the model consists of two elements: (i) a list of unanalyzed word types derived from a corpus in the lost language, and (ii) a morphologically analyzed lexicon in a known related language derived from a separate corpus, in our case non-parallel.

4.6 Model

In this section we describe our model for lost language decipherment. This model is designed to encode various intuitions that humans have used in lost language decipherment. We first describe some of these intuitions.

4.6.1 Intuitions

Our goal is to incorporate the logic and intuition used by human decipherers in an unsupervised statistical model. To make these intuitions concrete, consider the following toy example, consisting of a lost language much like English, but written using numerals:

- 15234 *(asked)*
- 1525 *(asks)*
- 4352 *(desk)*

Analyzing the undeciphered corpus, we might first notice a pair of endings, -34, and -5, which both occur after the initial sequence 152- (and may likewise occur at the end of a variety of words in the corpus). If we know this lost language to be closely related to English, we can surmise that these two endings correspond to the English verbal suffixes -*ed* and -*s*. Using this knowledge, we can hypothesize the following character correspondences: (3 = *e*), (4 = *d*), (5 = *s*). We now know that (4352 = *des*2) and we can use our knowledge of the English lexicon to hypothesize that this word is *desk*, thereby learning the correspondence (2 = *k*). Finally, we can use similar reasoning to reveal that the initial character sequence 152- corresponds to the English verb *ask*.

As this example illustrates, human decipherment efforts proceed by discovering both character-level and morpheme-level correspondences. This interplay implicitly relies on a morphological analysis of words in the lost language, while utilizing knowledge of the known language's lexicon and morphology.

One final intuition our model captures is the sparsity of the alphabetic correspondence between related languages. We know from comparative linguistics that the correct mapping will preserve regular phonetic relationships between the two languages (as exemplified by cognates). As a result, each character in one language will map to a small number of characters in the other language (typically one, but sometimes two or three). By incorporating this structural sparsity intuition, we can allow the model to focus on on a smaller set of linguistically valid hypotheses.

Below we give an overview of our model, which is designed to capture these linguistic intuitions.

4.6.2 Model Structure

We start with the assumption that some number of observed word-forms in the lost language are cognate to words in the known language. Our model posits that these lost language words are composed of a sequence of morphemes (prefix, stem, suffix) each of which was probabilistically generated jointly with a latent counterpart in the known language.

Our goal is to find the morphemic boundaries and known language counterparts that lead to consistent correspondences both at the character and morpheme level. The technical challenge is that each level of correspondence (character and morpheme) can completely describe the observed data. A probabilistic mechanism based simply on one leaves no room for the other to play a role. We resolve this tension by employing a hierarchical non-parametric Bayesian model: the distributions over bilingual morpheme pairs assign probability based on recurrent patterns at the morpheme level. These distributions are themselves drawn from a prior probabilistic process which favors distributions with consistent character-level correspondences.

We now give a top-down formal description of the model. See figure 4-1 for an accompanying graphical overview. There are four basic layers in the generative process:

1. **Structural sparsity**: draw a set of indicator variables $\vec{\lambda}$, each corresponding to a character substitution (u, h).

2. **String-edit distribution**: draw a *base distribution* G_0 parameterized by weights on character-level edit operations.

3. **Morpheme-pair distributions**: draw a set of distributions on bilingual morpheme pairs $G^{stm}, G_1^{pre}, G_1^{suf}, \ldots$ from the Dirichlet process $\mathrm{DP}(G_0, \alpha_0)$.

4. **Word generation**: draw cognate-pairs in the lost and known language, as well as words in the lost language with no cognates.

We now go through each step in more detail.

Structural Sparsity

The first step of the generative process provides a control on the structural sparsity of character-substitution probabilities. By "structural sparsity" in this context, we refer to the desire that each character in the lost and known languages map to a very limited number of characters in the other language.

For each pair of characters (u, h) (where u and h range over characters in the lost and known languages, respectively) we posit a 0-1 indicator variable $\lambda_{(u,h)}$. Intuitively, we would like $\lambda_{(u,h)} = 1$ to indicate that u and h are *reflexes* of one another. That is, that the phonemes these two characters represent descend from a

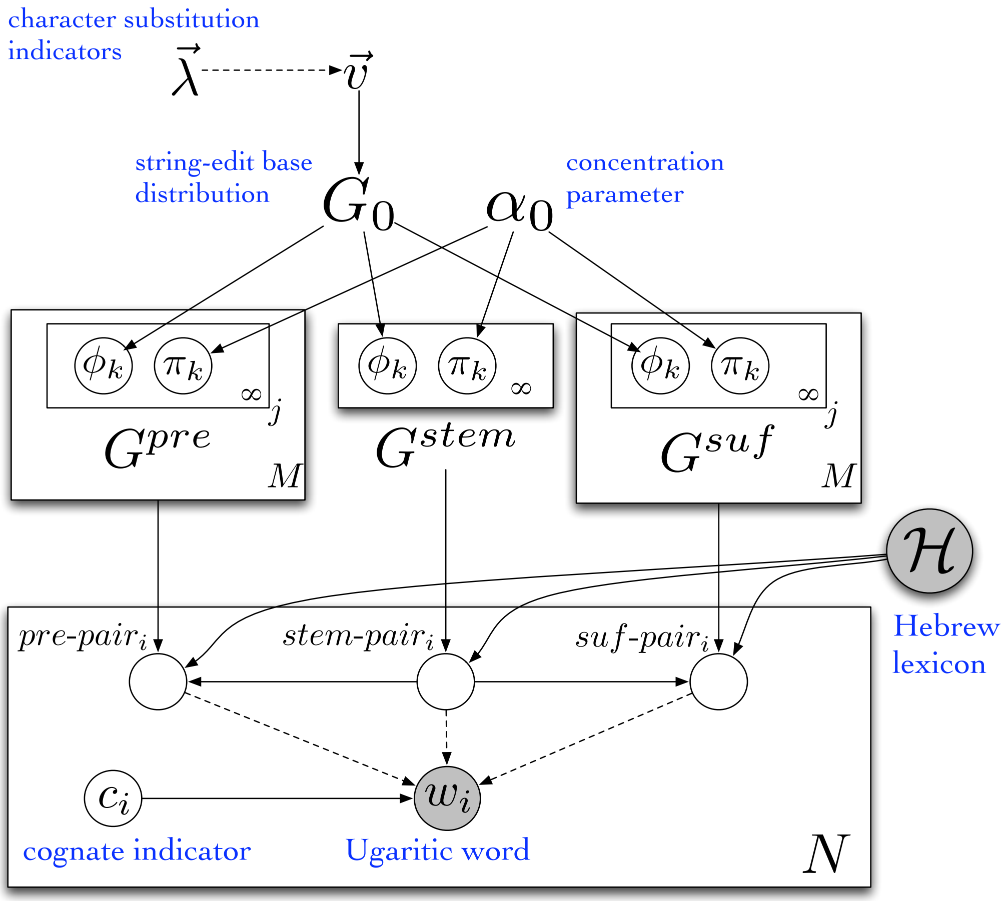


Figure 4-1: **Plate diagram of the decipherment model.** Observed variables are shaded in grey; full lines indicate probabilistic dependencies; dotted lines indicate deterministic dependencies; boxes indicate repeated variables, with the value in the bottom-right of each box indicating the number of repetitions. The structural sparsity indicator variables $\vec{\lambda}$ determine the values of the base distribution hyperparameters $\vec{v}$. The base distribution G_0 defines probabilities over string-pairs based on character-level edit operations. The stem-pair distribution G^{stm} is drawn from the Dirichlet process $\mathrm{DP}(G_0, \alpha_0)$ and is characterized by an infinite sequence of string-pairs (ϕ_k) and an accompanying sequence of weights (π_k). For each of M parts-of-speech, distributions over bilingual prefix-pairs and suffix pairs, G^{pre} and G^{suf} are likewise drawn. For each of N Ugaritic word-forms, an indicator variable c_i is drawn. If $c_i = 0$, the Ugaritic word w_i is drawn from an Ugaritic character-level language model. Otherwise, bilingual Ugaritic-Hebrew morpheme pairs are drawn, which deterministically yield the Ugaritic word w_i.

common phoneme in an ancestor language and that therefore u and h are likely to substitute for one another in cognate-pairs. Thus $\big\{(u,h) \mid \lambda_{(u,h)} = 1\big\}$ represents the set of historically valid alphabetic mappings. We next define a joint prior over these variables $\vec{\lambda}$ which encourages sparse character mappings. With m lost-language characters and n known-language characters, we can view the set of possible values for $\vec{\lambda}$ as the set of all binary $m \times n$ matrices. Thus, the prior $P(\vec{\lambda})$ should define a distribution over such matrices which encourages both row and column sparsity:

$$\forall u : \sum_h \lambda_{(u,h)} \approx 1, \quad \forall h : \sum_u \lambda_{(u,h)} \approx 1.$$

Defining a normalized probability distribution over binary matrices that achieves this effect is difficult. Instead, we define our prior indirectly through a real-valued positive function g:

$$P(\vec{\lambda}) = \frac{g(\vec{\lambda})}{Z} \tag{4.1}$$

The value Z is a normalization term which depends only on the matrix dimensions m and n (i.e. the number of letters in the two alphabets):[3]

$$Z = \sum_{\vec{\lambda}} g(\vec{\lambda})$$

We define g in terms of two vector-valued feature functions $f(\vec{\lambda})$ and $f'(\vec{\lambda})$ along with a weight vector $\vec{w}$:

$$g(\vec{\lambda}) = \exp\Big(f(\vec{\lambda}) \cdot \vec{w} + f'(\vec{\lambda}) \cdot \vec{w}\Big) \tag{4.2}$$

Intuitively, f and f' count the number of sparsity violations for the lost and known languages, respectively, and $\vec{w}$ penalizes these violations. More precisely, we count the number of known-language characters to which each lost-language letter u maps:

[3] Since computing Z is intractable, we develop an inference algorithm below (section 4.7.4) which only requires computation of the unnormalized function $g(\vec{\lambda})$.

$c(u) = \sum_h \lambda_{(u,h)}$.[4] We then define a vector of features which count how many lost-language characters u map to exactly i known-language characters beyond some allowed budget b_i:

$$f(\vec{\lambda})_i = \max\bigl(0,\ \bigl|\{u : c(u) = i\}\bigr| - b_i\bigr) \tag{4.3}$$

In similar fashion we define f', merely swapping the roles of u and h and defining corresponding budget values b'_i. Finally, we set:

$$\begin{aligned}
&\vec{w} = (w_0 = -\infty, w_1 = 0, w_2 = -50, w_{>2} = -\infty)\\
&\vec{b} = (b_i = 0, \forall i)\\
&\vec{b'} = (b'_0 = 0, b'_1 = 0, b'_2 = 7, b'_3 = 1, b'_{>3} = 0)
\end{aligned}$$

The asymmetry in our definition of $\vec{b}$ and $\vec{b'}$ results from our observation that the Ugaritic script contains seven characters more than Hebrew. Thus, we allow up to seven Hebrew letters to map to two Ugaritic letters without penalty, and we allow one Hebrew letter to map to three Ugaritic letters without penalty. In the reverse direction, no such allowances are made. Every Ugaritic letter which maps to more than one Hebrew letter is immediately penalized.

String-edit Distribution

The next step in the generative process is to draw our base measure G_0, which defines a distribution over all string pairs $(\mathbf{u}, \mathbf{h})$ (where $\mathbf{u}$ is composed of lost-language characters and $\mathbf{h}$ is composed of known-language characters). Distribution G_0 assigns probabilities based on character-level edit operations. These operations consist of substitutions (u, h), insertions (ϵ, h), deletions (u, ϵ), and a stop symbol (ϵ, ϵ).

Under G_0, each edit operation e is assigned a weight $0 \leq \rho_e \leq 1$. We partition

[4] The functional dependence of $c(u)$ on $\vec{\lambda}$ is suppressed for notational clarity.

edit operations into three categories,

$$\begin{aligned} SUB &= \big\{(u,h) : \forall u \forall h\big\} \cup \big\{(\epsilon,\epsilon)\big\} \\ INS &= \big\{(\epsilon,h) : \forall h\big\} \\ DEL &= \big\{(u,\epsilon) : \forall u\big\}, \end{aligned}$$

and we require the weights corresponding to each set to sum to one:

$$\sum_{e \in SUB} \rho_e = 1, \ \sum_{e \in INS} \rho_e = 1, \ \sum_{e \in DEL} \rho_e = 1$$

In addition, G_0 provides a fixed distribution q over the *number* of insertions and deletions occurring in any single edit-sequence. Probabilities over edit-sequences are then defined according to G_0 by:

$$\frac{q\left(n_I, n_D\right)}{n_S^{[n_I+n_D]}} \cdot \prod_i \rho_{e_i}, \tag{4.4}$$

where n_I, n_D, n_S are the number of insertions, deletions, and substitutions in the edit-sequence, and the notation $x^{[n]}$ represents the rising factorial: $x(x+1)\cdots(x+n-1)$.

This function can be shown to be a probability mass function over all possible edit-sequences through the following generative process:

1. Draw substitutions according to $\{\rho_e : e \in SUB\}$ until the end symbol (ϵ,ϵ) is drawn.

2. Draw the number of insertions and deletions n_I, n_D according to q.

3. Draw n_I insertions according to $\{\rho_e : e \in INS\}$ and n_D deletions according to $\{\rho_e : e \in DEL\}$.

4. Place the insertions and deletions among the substitutions with uniform probability (i.e. with probability $\frac{1}{n_S^{[n_I+n_D]}}$)

The traditional probabilistic string-edit formulation (e.g. [98]) places all types of edit-operations under a single multinomial distribution. While this results in a simpler distribution over edit-sequences (i.e. simply $\prod_i \rho_i$), it has certain drawbacks which our formulation overcomes. In particular, we believe that it is important to provide an explicit distribution over the number of insertions and deletions and to prevent insertion and deletions from competing with substitutions for probability mass.

Note that each edit-sequence yields a string pair through projection and ϵ-removal:

$$\begin{aligned} y_1(\vec{e}) &\triangleq (e_1)_1 \cdots (e_k)_1 = \mathbf{u} \\ y_2(\vec{e}) &\triangleq (e_1)_2 \cdots (e_k)_2 = \mathbf{h} \\ y(\vec{e}) &\triangleq \big(y_1(\vec{e}),\ y_2(\vec{e})\big) = (\mathbf{u}, \mathbf{h}) \end{aligned} \tag{4.5}$$

If so desired, we can define an explicit distribution on string-pairs $(\mathbf{u}, \mathbf{h})$ by summing over all possible edit-sequences which yield $(\mathbf{u}, \mathbf{h})$:

$$P(\mathbf{u}, \mathbf{h}) = \sum_{\vec{e} : y(\vec{e}) = (\mathbf{u}, \mathbf{h})} P(\vec{e})$$

However, in the remainder of this thesis we will simply define G_0 as a distribution over edit-sequences $\vec{e}$.

In setting the value of q (the distribution over the number of insertions and deletions), we observe that the average Ugaritic word is over two letters longer than the average Hebrew word. Thus, occurrences of Hebrew character insertions are *a priori* likely, and Ugaritic character deletions are very unlikely. In our experiments, we simply set q to disallow Ugaritic deletions, and to allow up to one Hebrew insertion per morpheme (with probability 0.5).

As G_0 consists of three multinomial distributions (over SUB, INS, and DEL), we draw the three corresponding sets of weights from conjugate-prior Dirichlet dis-

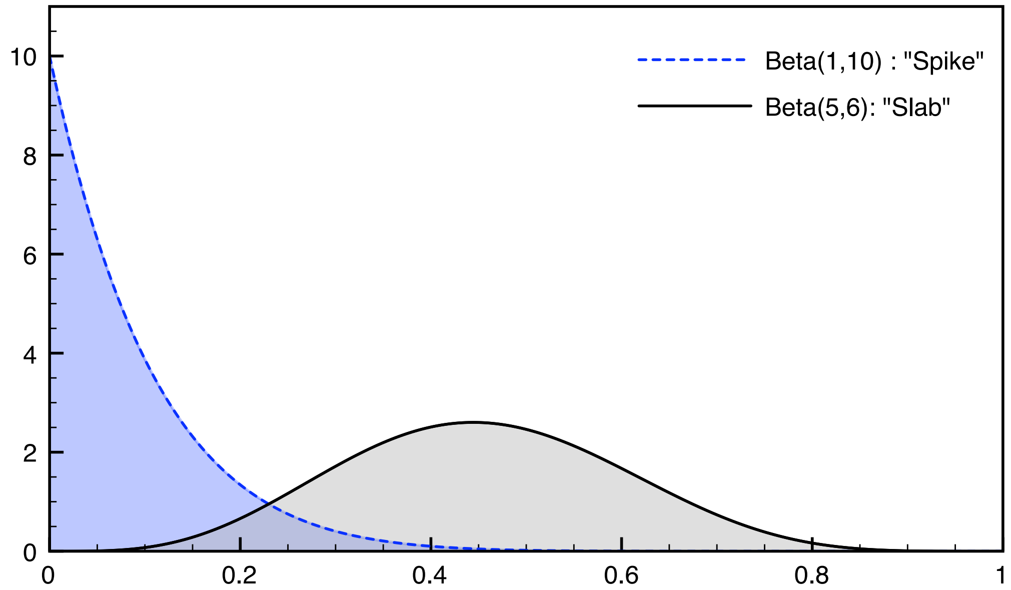


Figure 4-2: **"Spike-and-slab" effect** on character-edit probabilities $\vec{\rho}$. Assume three edit operations with indicator variables $\vec{\lambda} = (0,1,1)$ and resulting hyperparameters $\vec{v} = (1,5,5)$. Then the marginal priors are $\rho_{e_1} \sim \text{Beta}(1, 5+5)$ and $\rho_{e_2}, \rho_{e_3} \sim \text{Beta}(5, 1+5)$.

tributions.

$$\{\rho_e : e \in INS\} \sim \text{Dirichlet}(\mathbf{1}) \tag{4.6}$$

$$\{\rho_e : e \in DEL\} \sim \text{Dirichlet}(\mathbf{1}) \tag{4.7}$$

$$\{\rho_e : e \in SUB\} \sim \text{Dirichlet}(\vec{v}) \tag{4.8}$$

For insertions and deletions we simply set all the Dirichlet hyperparameters to 1. In the case of substitutions, we employ the previously sampled sparsity indicator variables to deterministically set the hyperparameter vector $\vec{v}$. In particular, each hyperparameter value v_e corresponds to a character edit-operation e and is set according to the indicator variable λ_e:

$$v_e = \begin{cases} 1 & \text{if } \lambda_e = 0, \\ K & \text{if } \lambda_e = 1. \end{cases} \tag{4.9}$$

where K is some constant value > 1 (set to 50 in our experiments). The resulting effect is that when $\lambda_e = 0$, the marginal prior density of the corresponding edit weight ρ_e spikes at 0. When $\lambda_e = 1$, the corresponding marginal prior density remains relatively flat and unconstrained. Figure 4-2 illustrates this effect graphically.

For similar applications of "spike and slab" priors, see [59] in the regression scenario and [119] in the context of topic models.

Morpheme-pair Distributions

Next we draw a series of distributions which *directly* assign probability to morpheme pairs (or more precisely to edit sequences which yield morpheme pairs). The previously drawn base measure G_0, along with a fixed concentration parameter α_0, define a *Dirichlet process* [2]: $\mathrm{DP}(G_0, \alpha_0)$, which provides a probability distribution over all possible morpheme-pair distributions. Distributions drawn from this Dirichlet process assign large probability mass to a small number of morpheme pairs, while remaining sensitive to the character-level substitution probabilities of the base distribution.

Our model distinguishes between three types of morphemes: prefixes, stems, and suffixes. Since each part-of-speech in a language carries with it unique prefix and suffix frequencies, we generate distinct prefix and suffix distributions for each of M parts-of-speech, and a single distribution over all stem-pairs:

$$\begin{array}{lll} G^{stm} & \sim & \mathrm{DP}(G_0, \alpha_0) \\ \forall j \in 1 \ldots M: \quad G_j^{pre} & \sim & \mathrm{DP}(G_0, \alpha_0) \\ \forall j \in 1 \ldots M: \quad G_j^{suf} & \sim & \mathrm{DP}(G_0, \alpha_0) \end{array}$$

While we avoid dealing directly with these distributions in our inference procedure, they can be viewed as arising from a stick-breaking process [103]:

1. Draw an infinite sequence of i.i.d. edit-sequences from the base distribution:

   $\phi_1, \phi_2, \ldots \sim G_0$

2. Draw an infinite sequence of i.i.d. weights:

   $\pi'_1, \pi'_2, \ldots \sim \text{Beta}(1, \alpha_0)$
3. Normalize the weights:

   $\pi_k = \pi'_k \cdot \sum_{i<k} (1 - \pi'_i)$
4. Define a probability density function over edit-sequences using the dirac delta function[5]:

$$p(\vec{e}) = \sum_{k=1}^{\infty} \pi_k \cdot \delta_{\phi_k = \vec{e}} \tag{4.10}$$

Word Generation

Once the morpheme-pair distributions have been drawn, actual words may now be generated. To generate word $\mathbf{u}_i$ of the N word-forms observed in the lost-language texts, we first draw a cognate indicator variable c_i. This variable determines whether $\mathbf{u}_i$ is to be generated along with a known-language cognate, or alone as a non-cognate. We model c_i as a simple Bernoulli random variable, with fixed parameter $P(c_i = 1)$. As discussed below in section 4.9.3, this cognate prior may be varied to induce different prediction thresholds. In our main experiments, we simply set $P(c_i = 1) = 0.5$. If $c_i = 1$, then a cognate word pair $(\mathbf{u}_i, \mathbf{h}_i)$ is generated according

[5] $\delta_b = (1 \text{ if } b, 0 \text{ if } \neg b)$

to our model as follows:

$$\begin{array}{lcll}
\vec{e}_{stm} & \sim & G^{stm} & \text{// draw stem edit-sequence} \\
\text{pos}_i & \leftarrow & j = \text{pos}(\vec{e}_{stm}) & \text{// determine part-of-speech } j \text{ (see below)} \\
\vec{e}_{pre} & \sim & G_j^{pre} & \text{// draw prefix edit-sequence} \\
\vec{e}_{suf} & \sim & G_j^{suf} & \text{// draw suffix edit-sequence} \\
\big(\mathbf{u}^{pre}, \mathbf{h}^{pre}\big) & \leftarrow & y(\vec{e}_{pre}) & \text{// yield morpheme pairs (equation 4.5)} \\
\big(\mathbf{u}^{stm}, \mathbf{h}^{stm}\big) & \leftarrow & y(\vec{e}_{stm}) & \\
\big(\mathbf{u}^{suf}, \mathbf{h}^{suf}\big) & \leftarrow & y(\vec{e}_{suf}) & \\
\mathbf{u}_i & \leftarrow & \mathbf{u}^{pre}\,\mathbf{u}^{stm}\,\mathbf{u}^{suf} & \text{// concatenate morphemes} \\
\mathbf{h}_i & \leftarrow & \mathbf{h}^{pre}\,\mathbf{h}^{stm}\,\mathbf{h}^{suf} &
\end{array}$$

Besides observing the resulting lost-language word $\mathbf{u}_i$, we also assume the existence of a known-language lexicon $\mathcal{H}$. This lexicon provides us with knowledge of all possible stems, prefixes, and suffixes, for each part-of-speech j in the known language: $\mathcal{H}_j^{stm}, \mathcal{H}_j^{pre}, \mathcal{H}_j^{suf}$. We treat this knowledge as a hard constraint on the set of possible values for the latent cognate word $\mathbf{h}_i$. In particular, we treat as an observation that for some part-of-speech j:

$$\mathbf{h}^{pre} \in \mathcal{H}_j^{pre} \quad \wedge \quad \mathbf{h}^{stm} \in \mathcal{H}_j^{stm} \quad \wedge \quad \mathbf{h}^{suf} \in \mathcal{H}_j^{suf}.$$

Probabilistically, we view the part-of-speech assignment j as being determined by the generated stem edit-sequence: [6]

$$\text{pos}(\vec{e}_{stm}) \;\triangleq\; j \;\text{ such that } y_2(\vec{e}_{stm}) \in G_j^{stm}$$

[6] If more than one part-of-speech meet this criterion, then we assume a uniform distribution over all such values.

The prefix and suffix morpheme-pairs are then drawn from the appropriate distributions G_j^{pre} and G_j^{suf}. In this way, the prefix and suffix both probabilistically depend on the stem (by way of its part-of-speech). We will see in section 4.7.1 what role all these observations play in our inference algorithm.

If $c_i = 0$, then $\mathbf{u}_i$ was generated without a known-language cognate. We assume that the lone lost-language word was generated according to a unigram character-level language model

$$P(\mathbf{u}_i | c_i = 0) = P(\#) \cdot \prod_j P\left(\mathbf{u}_i[j]\right), \tag{4.11}$$

where # is a special end-word character and $\mathbf{u}_i[j]$ denotes the j^{th} character in word $\mathbf{u}_i$. We note that this differs somewhat from the model presented in our previous publication [109], where a uniform distribution was assumed over letters of the lost language for the purpose of noncognate generation. See section 4.9.3 below for a more detailed discussion of this issue.

In summary, this model structure captures both character and lexical level correspondences, while utilizing morphological and part-of-speech knowledge of the known language. An additional feature of this multi-layered model structure is that each distribution over morpheme pairs is derived from the single character-level base distribution G_0. As a result, any character-level mappings learned from one type of morphological correspondence will be propagated to all other morpheme distributions. Finally, the character-level mappings discovered by the model are encouraged to obey linguistically motivated structural sparsity constraints. In the next section we describe our inference procedure at length.

4.7 Inference

For each word-form $\mathbf{u}_i$ in our undeciphered language we wish to first predict a cognate indicator variable c_i and, if $c_i = 1$, then the corresponding morphemes in

Algorithm 5: Gibbs sampler for lost language decipherment.

```
Input: Lost-language word forms $\mathbf{u}_1, \ldots, \mathbf{u}_N$ and known-language lexicon $\mathcal{H}$
Output: 1000 samples of latent variables
Initialize latent variables;
for $r \leftarrow 1$ to 1000 do
    for $i \leftarrow 1$ to $N$ do
        Sample word analysis $\left[\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}\right]_i^{(r)}$    // Section 4.7.1
        Sample cognate indicator $c_i^{(r)}$    // Section 4.7.2
    Resample Chinese restaurant tables;    // Section 4.7.3
    foreach lost-language character $u$ do
        Sample sparsity indicators $\left\{\lambda_{(u,h)} : \forall h\right\}^{(r)}$    // Section 4.7.4
    foreach known-language character $h$ do
        Sample sparsity indicators $\left\{\lambda_{(u,h)} : \forall u\right\}^{(r)}$    // Section 4.7.4
```

the known language $\left(\mathbf{h}^{pre}\,\mathbf{h}^{stm}\,\mathbf{h}^{suf}\right)_i$. Ideally we would predict each word analysis with highest posterior marginal probability under our model given the observed undeciphered corpus $\mathbf{u}_1, \ldots, \mathbf{u}_N$ and known-language lexicon $\mathcal{H}$. In order to do so, we need to integrate out all other latent variables in our model:

- Structural sparsity indicator variables $\vec{\lambda}$
- String-edit base distribution G_0
- Morpheme-pair distributions $G^{stm}, G_1^{pre}, G_1^{suf}, \ldots$
- Morphological segmentations
- Latent cognates of all other words $\mathbf{u}_{k \neq i}$

As these integrals are intractable to compute exactly, we resort to the standard Monte Carlo approximation. We collect samples of the variables over which we wish to marginalize but for which we cannot compute closed-form integrals. We then approximate the marginal probabilities for each undeciphered word $\mathbf{u}_i$ by summing over all the samples, and predicting the analysis with highest posterior probability.

In our sampling algorithm, we avoid sampling the base distribution G_0 and the morpheme-pair distributions (G^{stm} etc.), instead marginalizing them out using analytical closed forms. We explicitly sample the sparsity indicator variables $\vec{\lambda}$, the cognate indicator variables c_i, and latent word analyses (segmentations and cognate counterparts). To sample these variables tractably, we use Gibbs sampling to sample each latent variable conditioned on our current sample of the others. Although the samples are no longer independent, they form a Markov chain whose stationary distribution is the true joint distribution defined by the model [40].

See algorithm 5 for a high-level overview of our Gibbs sampler. In the following sections we provide details for each sampling step.

4.7.1 Sampling Word Analyses

For each lost-language word $\mathbf{u}_i$ with corresponding cognate indicator $c_i = 1$, we sample a segmentation $\mathbf{u}^{pre}\,\mathbf{u}^{stm}\,\mathbf{u}^{suf}$ with corresponding cognate morphemes $\mathbf{h}^{pre}\,\mathbf{h}^{stm}\,\mathbf{h}^{suf}$.

More precisely, we sample three edit-sequences $\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}$ which yield:

$$\begin{aligned} y(\vec{e}_{pre}) &= \big(\mathbf{u}^{pre}, \mathbf{h}^{pre}\big) \\ y(\vec{e}_{stm}) &= \big(\mathbf{u}^{stm}, \mathbf{h}^{stm}\big) \\ y(\vec{e}_{suf}) &= \big(\mathbf{u}^{suf}, \mathbf{h}^{suf}\big), \end{aligned}$$

with the hard constraint that the resulting analysis must be consistent (i) with the observed lost-language word-form $\mathbf{u}_i$:

$$\mathbf{u}_i \quad = \quad \mathbf{u}^{pre}\,\mathbf{u}^{stm}\,\mathbf{u}^{suf},$$

and (ii) with the observed known-language lexicon $\mathcal{H}$:

$$\mathbf{h}^{pre} \in \mathcal{H}^{pre}_j \;\wedge\; \mathbf{h}^{stm} \in \mathcal{H}^{stm}_j \;\wedge\; \mathbf{h}^{suf} \in \mathcal{H}^{suf}_j \quad \text{(for some part-of-speech } j\text{)}.$$

We break this task down into two steps, (i) first sampling a segmentation and part-of-speech, and (ii) then sampling the actual edit-sequences (yielding the corresponding cognate morphemes).

Sampling Segmentations and Parts-of-speech

We sample a segmentation and part-of-speech by simply enumerating all possibilities, calculating their posterior probabilities, and sampling from the resulting discrete distribution.

We now show how to calculate the posterior for a each segmentation $\mathbf{u}_i = \mathbf{u}^{pre}\,\mathbf{u}^{stm}, \mathbf{u}^{suf}$ and part-of-speech j:[7]

$$\begin{aligned}
& P(\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, \text{pos}_i = j \mid \mathbf{u}_i, \mathcal{H}) \\
= & \sum_{\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}} P(\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}, \mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, j \mid \mathbf{u}_i, \mathcal{H}) \\
\propto & \sum_{\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}} P(\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}, \mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, j) \;\; P(\mathcal{H} \mid \vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}, j) \\
= & \sum_{\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf} \in C_1 \times C_2 \times C_3} P(\vec{e}_{pre}, \vec{e}_{suf} \mid j) \;\; P(\vec{e}_{stm}) \; P(\mathcal{H}) \\
\propto & \sum_{\vec{e}_{pre} \in C_1} P(\vec{e}_{pre} \mid j) \sum_{\vec{e}_{stm} \in C_2} P(\vec{e}_{stm}) \sum_{\vec{e}_{suf} \in C_3} P(\vec{e}_{suf} \mid j)
\end{aligned} \tag{4.12}$$

where C_1, C_2, C_3 are the sets of all edit-sequences (i) yielding the respective morphemes $\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}$ and (ii) consistent with the known-language lexicon for part-

[7] For notational clarity, we leave the conditioning on the other sampled variables implicit throughout this section.

of-speech j. More precisely:

$$C_1 = \left\{\vec{e} \mid y_1(\vec{e}) = \mathbf{u}^{pre} \ \wedge \ y_2(\vec{e}) \in \mathcal{H}_j^{pre}\right\}$$
$$C_2 = \left\{\vec{e} \mid y_1(\vec{e}) = \mathbf{u}^{stm} \ \wedge \ y_2(\vec{e}) \in \mathcal{H}_j^{stm}\right\}$$
$$C_3 = \left\{\vec{e} \mid y_1(\vec{e}) = \mathbf{u}^{suf} \ \wedge \ y_2(\vec{e}) \in \mathcal{H}_j^{suf}\right\}$$

Computing Edit-sequence Probabilities

We defer for now the question of how to efficiently sum over these sets, and instead start by deriving the individual probability terms in equation 4.12. Let us consider the posterior probability of a stem edit-sequence $\vec{e}$. We first note that were we to Gibbs-sample the parameters of the stem morpheme-pair distribution G^{stm}, $P(\vec{e})$ would be directly given by equation 4.10. Instead, we marginalize out these parameters using the standard Chinese Restaurant Process closed form [34]:

$$P(\vec{e}) = \frac{N_{stm}(\vec{e}) + \alpha_0 \cdot F(\vec{e})}{N_{cog} + \alpha_0}, \tag{4.13}$$

where $N_{stm}(\vec{e})$ gives the number of other cognated words for which $\vec{e}$ appears as the stem: $\left|\{k \neq i \mid (\vec{e}_{stm})_k = \vec{e} \wedge c_k = 1\}\right|$, and N_{cog} gives the total number of other cognated words: $\left|\{k \neq i \mid c_k = 1\}\right|$.[8] The function $F(\vec{e})$ gives the probability assigned to $\vec{e}$ by the string-edit base distribution. If the multinomial parameters of the base distribution were Gibbs-sampled, then F would be given by equation 4.4. Instead, we employ the standard marginalized posterior distribution for multinomials with Dirichlet hyperparameters to obtain:

$$F(\vec{e}) = \frac{q(n_I, n_D)}{n_S^{[n_I+n_D]}} \cdot \prod_{e \in \vec{e}} f(e), \tag{4.14}$$

[8] Both according to the most recently sampled latent values for words $\mathbf{u}_{k\neq i}$.

with:

$$f(e) = \begin{cases} \frac{N(e)+1}{N_I+\sum_{\text{ins } e'} 1} & \text{if } e = (\epsilon, h), \\ \frac{N(e)+1}{N_D+\sum_{\text{del } e'} 1} & \text{if } e = (u, \epsilon), \\ \frac{N(e)+v_e}{N_S+\sum_{\text{sub } e'} v_{e'}} & \text{if } e = (u, h) \text{ or } e = (\epsilon, \epsilon) \end{cases}$$

where $N(e)$ denotes the number of times the edit-operation e has occurred in the *unique* edit-sequences obtained from each morpheme-pair distribution $(G^{stm}, G_1^{pre}, \ldots)$; or more precisely, the number of times e appears among the sets:

$$\begin{aligned} E^{stm} &= \big\{(\vec{e}_{stm})_{k \neq i}\big\} \\ E_j^{pre} &= \big\{(\vec{e}_{pre})_{k \neq i} \mid \text{pos}_k = j\big\},\ \forall j \in 1 \ldots M \\ E_j^{suf} &= \big\{(\vec{e}_{suf})_{k \neq i} \mid \text{pos}_k = j\big\},\ \forall j \in 1 \ldots M. \end{aligned}$$

And N_I, N_D, N_S give the *total* number of respective insertions, deletions, and substitutions among those sets. We emphasize that these sets only distinguish between *unique* edit-sequences (e.g. the edit-sequence $(\vec{e}_{stm})_m = (\vec{e}_{stm})_n$ for $m \neq n$ would only occur once in E^{stm}). Only a single instance of each edit-sequence type is drawn from the base distribution itself – the first person at each Chinese restaurant table, as it were. Finally, for an explanation of the first factor in equation 4.14 we refer the reader back to equation 4.4.

Similar reasoning applies in deriving the probability terms for prefix and suffix edit-sequences, with the exception that counts are now restricted to a individual parts-of-speech (since we have different G_j^{pre} and G_j^{suf} for each part-of-speech j). E.g. for prefixes:

$$P(\vec{e} \mid j) = \frac{N_{pre,j}(\vec{e}) + \alpha_0 \cdot F(\vec{e})}{N_{cog,j} + \alpha_0}$$

where $N_{pre,j}(\vec{e})$ gives the number of other cognated words with part-of-speech j for which sequence $\vec{e}$ appears as the prefix: $\big|\{k \neq i \mid (\vec{e}_{pre})_k = \vec{e} \ \wedge\ c_k = 1 \ \wedge\ \text{pos}_k = j\}\big|$, and N_{cog} gives the total number of other cognated words with part-of-speech j: $\big|\{k \neq i \mid c_k = 1 \ \wedge\ \text{pos}_k = j\}\big|$. Note that all edit-sequence posteriors depend on the function $F(\vec{e})$ since all morpheme-pair distributions derive from the same base measure G_0.

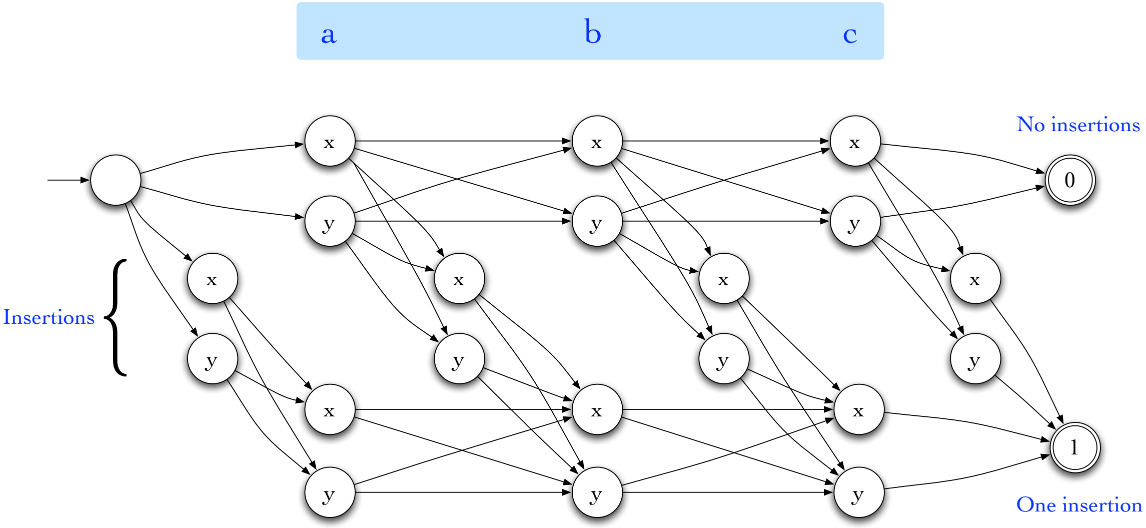


Figure 4-3: **WFSA** $A(abb)$**.** States correspond to following sets of edits: $\{(a,x),(a,y),(b,x),(b,y),(\epsilon,x),(\epsilon,y),(\epsilon,\epsilon)\}$. The top two rows correspond to substitution states for which no insertion has yet occurred. The middle two rows correspond to insertion states. The bottom two rows correspond to substitution states for which an insertion has already occurred.

Summing Probabilities with Finite-state Machines

The individual probability terms in equation 4.12 can thus be computed by simply caching counts from each word's sampled analysis. However, to compute segmentation probabilities we still need to efficiently sum these terms over the sets C_1, C_2, and C_3. For example, plugging equation 4.13 into the middle sum of equation 4.12 yields the following computation (after removing the constant denominator):

$$\sum_{\vec{e}\in C_2} N_{stm}(\vec{e}) \quad + \quad \alpha_0 \sum_{\vec{e}\in C_2} F(\vec{e}) \tag{4.15}$$

Recall that set C_2 contains all edit-sequences $\vec{e}$ yielding a particular lost-language morpheme $\mathbf{u}^{stm}$ along with any $\mathbf{h}^{stm} \in \mathcal{H}_j^{stm}$. The left-hand sum is easy to compute, as we can ignore any edit-sequence absent in our current sample. (We simply enumerate the edit-sequences which have already been observed for identical stems:

$\{(\vec{e}_{stm})_{k \neq i} \mid \mathbf{u}_k^{stm} = \mathbf{u}_i^{stm}\}$, and count how many times each appears.)

The right-hand sum is more difficult. Its computation requires summing over all edit-sequences in C_2, even those never seen before. A brute force computation would require: (i) the enumeration of all possible edit-sequences which yield $\mathbf{u}^{stm}$ (exponential in the length of $\mathbf{u}^{stm}$), and (ii) the removal of any such edit-sequence which yields a value $\mathbf{h}^{stm} \notin \mathcal{H}_j^{stm}$.

Fortunately, the function $F(\vec{e})$ (defined in equation 4.14) nearly factors over the individual edits $e \in \vec{e}$. This fact will allow us to compactly represent the set C_2 as a weighted finite-state acceptor (WFSA). We start by constructing a WFSA $A(\mathbf{u})$ which accepts *any string* $\mathbf{h}$ which can be jointly generated with $\mathbf{u}$ through a sequence of edits $\vec{e}$. In other words, the *language* of $A(\mathbf{u})$ is the set of strings $\{\mathbf{h} \mid \exists \vec{e} \ : \ y(\vec{e}) = (\mathbf{u}, \mathbf{h})\}$. Each state s of $A(\mathbf{u})$ will correspond to a single edit-operation e and incoming arcs will be weighted by the appropriate factors of equation 4.14:

- If $e = (\epsilon, h)$, incoming arcs accept the symbol h and are weighted by $f(e)$.

- If $e = (u, \epsilon)$, incoming arcs accept the symbol ϵ and are weighted by $f(e)$.

- If $e = (u, h)$, incoming arcs accept the symbol h and are weighted by $f(e)$.

- If $e = (\epsilon, \epsilon)$, s is an end-state, incoming arcs accept ϵ, and are weighted by $f(e) \cdot \frac{q(n_I, n_D)}{n_S^{[n_I + n_D]}}$.

The final case requires some clarification, as the values n_I, n_D, and n_S count the total number of insertions, deletions, and substitutions throughout the entire edit-sequence. This is the only item in equation 4.14 which doesn't factor over individual edits. As a result, the states in the WFSA need to keep track of the number of previously performed insertions and deletions. Since our distribution $q(n_I, n_D)$ only allows a single insertion (and no deletions) per morpheme, we can simply double our set of substitution states to track whether the allowed insertion has occurred yet or not and provide a unique end-state for each possibility. See figure 4-3 for an example.

Thus, every path of arcs $\vec{a}$ through $A(\mathbf{u})$ corresponds to an edit-sequence $\vec{e}$, with path arc-weights which satisfy: $\prod_{a\in\vec{a}} w(a) = F(\vec{e})$. However, $A(\mathbf{u})$ is not yet restrictive enough for our purposes. It accepts some strings $\mathbf{h} \notin \mathcal{H}_j^{stm}$. Set C_2, on the other hand, is restricted to edit-sequences which yield actual known-language morphemes $\mathbf{h}^{stm} \in \mathcal{H}_j^{stm}$. To add this restriction, we construct a *lexicon* acceptor A_j^{stm}. This WFSA accepts all and only those strings in the lexicon $\mathcal{H}_j^{stm}$ (with all arc weights set to 1).

We construct this lexicon WFSA by enumerating each morpheme $\mathbf{h} \in \mathcal{H}_j^{stm}$ and adding a separate path through A_j^{stm} which sequentially accepts the letters of $\mathbf{h}$. Initially, A_j^{stm} will be quite large due to many redundant states (e.g. the paths accepting strings *yyy* and *yyyx* would be entirely disjoint). However, after its initial construction, we apply the Hopcroft minimization algorithm [57] which yields an equivalent, but optimally compact, WFSA (e.g. the paths accepting strings *yyyy* and *yyyx* would now share an initial prefix).

Next, we intersect the optimized A_j^{stm} with $A(\mathbf{u})$ to produce a new WFSA $A_j^{stm}(\mathbf{u})$. This operation essentially *prunes* our original $A(\mathbf{u})$ and restricts its paths to those which correspond to edit-sequences which yield a string in the lexicon $\mathcal{H}_j^{stm}$. (The original arc weights of $A(\mathbf{u})$ remain unaffected.) Finally, we are ready to compute the right-hand sum in equation 4.15 as the sum of path-weights:

$$\sum_{\vec{e}\in C_2} F(\vec{e}) = \sum_{\vec{a}} \prod_{a\in\vec{a}} w(a)$$

where each $\vec{a}$ is a unique sequence of arcs from the start-state of $A_j^{stm}(\mathbf{u})$ to an end-state. Since this WFSA contains no cycles, its path-weight sum can easily be computed with a dynamic program. To each state s_k, we associate a value β_k with the following recursive definition:

$$\beta_k = \begin{cases} 1 & \text{if } s_k \text{ is an end state,} \\ \sum_{a\,:\, s_k \to s_\ell} w(a)\cdot\beta_\ell & \text{otherwise.} \end{cases} \tag{4.16}$$

Intuitively, β_k gives the total path-weight of paths starting at state s_k and ending at an end-state. Values can easily be computed by starting with the end-states and stepping backwards along arcs. The resulting value β_0, corresponding to the start-state s_0, gives the desired sum. This construction and computation can be similarly carried out for the other two sums in equation 4.12.

Sampling Edit-sequences

Once we've sampled a segmentation $\mathbf{u}_i = \mathbf{u}^{pre}\,\mathbf{u}^{stm}, \mathbf{u}^{suf}$ and part-of-speech $\text{pos}_i = j$, we turn to the next step: sampling the actual edit-sequences $\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}$. We start by examining their posterior probabilities:

$$\begin{aligned}
&P(\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf} \mid \mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, j, \mathcal{H}) \\
&\quad\propto \quad P(\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, \mathcal{H} \mid \vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}, j)\; P(\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf} \mid j) \\
&\quad\propto \quad \begin{cases} P(\vec{e}_{pre}|j)P(\vec{e}_{stm})P(\vec{e}_{suf}|j) & \text{if } \vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf} \in C_1 \times C_2 \times C_3, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

The sets C_1, C_2, C_3 are defined as in equation 4.12 to include all edit-sequences which yield the fixed lost-language morphemes $\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}$ along with known-language morphemes in the vocabulary $\mathcal{H}$. Now, instead of summing over these sets, we instead need to sample from them. We focus here on the sampling procedure for $\vec{e}_{stm}$, but similar reasoning applies to $\vec{e}_{pre}$ and $\vec{e}_{suf}$.

The individual probability term $P(\vec{e})$ has already been derived in equation 4.13, but we repeat it here for convenience:

$$P(\vec{e}) \;\propto\; N_{stm}(\vec{e}) + \alpha_0 \cdot F(\vec{e})$$

We break our sampling procedure into two steps. First we decide whether to draw our sample $\vec{e}$ according to the left-hand term (from an existing table in the "Chinese restaurant"), or according to the right-hand term (the base distribution). To make this decision, we calculate $p = \sum_{\vec{e} \in C_2} N_{stm}(\vec{e})$ and $Q = p + \alpha_0 \cdot \sum_{\vec{e} \in C_2} F(\vec{e})$. To calculate p, we need merely consider values which have already been observed for identical stems: $\left\{(\vec{e}_{stm})_{k \neq i} \mid \mathbf{u}_k^{stm} = \mathbf{u}_i^{stm}\right\}$. To calculate Q, we sum all the path-weight of WFSA $A_j^{stm}(\mathbf{u}^{stm})$ using the dynamic program derived in the previous section.

We then sample a Bernoulli random variable with parameter p/Q. If we draw a heads, we proceed to sample from the already observed edit-sequences $\left\{(\vec{e}_{stm})_{k \neq i} \mid \mathbf{u}_k^{stm} = \mathbf{u}_i^{stm}\right\}$ in proportion to the number of times they have each been observed: $N_{stm}(\vec{e})$. If the Bernoulli comes up tails, we proceed to sample from all edit-sequences $\vec{e} \in C_2$ according to $F(\vec{e})$. In other words, we wish to draw $\vec{e}$ with probability:

$$p(\vec{e}) = \begin{cases} \frac{F(\vec{e})}{Z} & \text{if } \vec{e} \in C_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

As before, we utilize our WFSA $A_j^{stm}(\mathbf{u}^{stm})$. Recall that each unique path through this WFSA corresponds to an edit-sequence $\vec{e} \in C_2$ with path-weight $F(\vec{e})$. As before, we employ the values β_k defined recursively in equation 4.16. This time, we sample a path arc-by-arc from the start-state s_0 until an end-state is reached. When we are in state s_k, we sample the next arc $a : s_k \to s_\ell$ according to:

$$P(a : s_k \to s_\ell) \; \propto \; w(a) \cdot \beta_\ell$$

This procedure results in a draw of $\vec{e} \in C_2$ with the desired probability. To see this, we can rewrite $p(\vec{e})$ using the chain rule as $\prod_i p(e_i \mid e_1, \ldots, e_{i-1})$. We can further rewrite each conditional probability by marginalizing over all possible edit-sequence completions, and then plugging in the definitions of $p(\vec{e})$ (equation 4.17) and $F(\vec{e})$

(equation 4.14):

$$\begin{aligned} p(e_i \mid e_1, \ldots, e_{i-1}) &= \sum_{e_{i+1} \cdots e_m} p(e_i, \ldots, e_m \mid e_1, \ldots, e_{i-1}) \\ &\propto \sum_{e_{i+1} \cdots e_m \,:\, e_1 \cdots e_m \in C_2} \left[\frac{q(n_I, n_D)}{n_S^{[n_I+n_D]}} \cdot \prod_{j=i}^{m} f(e_j) \right] \end{aligned}$$

Switching to the equivalent arc-view in the WFSA, we get:

$$\begin{aligned} p(a_i : s_k \to s_\ell \mid a_1, \ldots, a_{i-1}) &\propto \sum_{a_{i+1} \ldots a_m \,:\, a_1 \ldots a_m \in A_j^{stm}(\mathbf{u}^{stm})} \left[\prod_{j=i}^{m} w(a_j) \right] \\ &= w(a_i) \cdot \beta_\ell, \end{aligned}$$

giving us the proposed sampling formula.

4.7.2 Sampling Cognate Indicators

For each word $\mathbf{u}_i$, we sample a corresponding cognate indicator variable c_i. Recall that $c_i = 1$ indicates that $\mathbf{u}_i$ was generated along with a latent known-language cognate $\mathbf{h}_i$ (via edit-sequences $\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}$). Value $c_i = 0$ indicates that lost-language word $\mathbf{u}_i$ was generated alone according to a lost-language language model. The posterior for c_i is given by:

$$P(c_i \mid \mathbf{u}_i, \mathcal{H}) \;\propto\; P(\mathbf{u}_i \mid c_i, \mathcal{H}) \cdot P(c_i)$$

Thus we wish to to simple a Bernoulli random variable in proportion to the two values:

$$\begin{gathered} P(\mathbf{u}_i \mid c_i = 0, \mathcal{H}) \cdot P(c_i = 0) \\ P(\mathbf{u}_i \mid c_i = 1, \mathcal{H}) \cdot P(c_i = 1) \end{gathered}$$

In our model we treat $P(c_i = 1)$ as a fixed parameter (set to 0.5 in our experiments). The value $P(\mathbf{u}_i \mid c_i = 0, \mathcal{H}) = P(\mathbf{u}_i \mid c_i = 0)$ is given by the lost-language character language model of equation 4.11. The language model parameters are fixed using the observed character frequencies in the lost-language corpus. Finally, we calculate $P(\mathbf{u}_i \mid c_i = 1, \mathcal{H})$ by marginalizing over all segmentations and parts-of-speech of $\mathbf{u}_i$:

$$P(\mathbf{u}_i \mid c_i = 1, \mathcal{H}) = \sum_{\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, j} P(\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, j \mid \mathbf{u}_i, \mathcal{H})$$

The terms in this sum are identical to equation 4.12 and are calculated in the same manner. The sum itself is calculated through explicit enumeration of all segmentations $\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf} = \mathbf{u}_i$ and parts-of-speech $j \in 1 \ldots M$.

4.7.3 Resampling Chinese Restaurant Tables

Our sampling of individual word analyses can be viewed as inducing a three-part *clustering* of words, based on prefix, stem, and suffix edit-sequences. Consider the set of currently sampled stem edit-sequences: $E = \left\{\vec{e} \mid \exists i : (\vec{e}_{stm})_i = \vec{e}\right\}$. For each such edit-sequence $\vec{e} \in E$, we define its *cluster* as $c(\vec{e}) = \left\{i \mid (\vec{e}_{stm})_i = \vec{e}\right\}$. Thus $c[E]$ (the image of set E under function c) defines a partition over instances based on their sampled stem edit-sequences. In the Chinese restaurant metaphor, $c[E]$ gives us the set of sampled tables.

In the standard Gibbs sampling scenario, it can be difficult for these clusters to mix properly. To see this, consider a stem edit-sequence $\vec{e}$ and its cluster of instances $c(\vec{e})$. Single instances $i \in c(\vec{e})$ may shift in and out of the cluster either due to sampling a different edit-sequence $\vec{e'}$ for $\mathbf{u}_i^{stm}$ or due to segmenting $\mathbf{u}_i$ in a new way. However, changing the edit-sequence of the entire cluster would require individually sampling a new value *separately* for each instance $i \in c(\vec{e})$. In essence, this would require the temporary fragmentation of the cluster and its eventual rebuilding. So even if some other value $\vec{e'}$ is a much more likely candidate for $c(\vec{e})$, reaching it may involve passing through a very low probability sample path.

In order to avoid this problem, we introduce an additional procedure in which we explicitly resample each cluster's edit-sequence from the base distribution, conditioned on the clustering itself. Thus, for each cluster $c \in c[E]$ with a stem morpheme $\mathbf{u}^{stm}$ and part-of-speech j, we resample according to:

$$P(\vec{e}_{stm}|\mathbf{u}^{stm}, \mathcal{H}) \propto \begin{cases} F(\vec{e}) & \text{if for } y(\vec{e}) = (\mathbf{u}, \mathbf{h}): \ \mathbf{u} = \mathbf{u}^{stm} \ \wedge \ \mathbf{h} \in \mathcal{H}_j^{stm}, \\ 0 & \text{otherwise,} \end{cases}$$

where $F(\vec{e})$ is the marginalized posterior base distribution previously given in equation 4.14.[9] As in section 4.7.1, we sample a path through the WFSA $A_j^{stm}(\mathbf{u}^{stm})$ using the techniques discussed after equation 4.17. Once a new edit-sequence $\vec{e'}$ has been sampled, it is assigned to all the instances in cluster c. Similar operations are performed for the clusters induced by prefix and suffix assignments.

4.7.4 Sampling Sparsity Indicators

Recall from section 4.6.2 that for each pair of characters (u, h) in the lost and known languages, we posit a sparsity indicator variable $\lambda_{(u,h)}$. Intuitively, $\lambda_{(u,h)} = 1$ indicates that u and h represent historically related phonemes which often substitute for one another in cognate pairs. Formally, $\lambda_{(u,h)}$ determines the value of $v_{(u,h)}$, the Dirichlet hyperparameter corresponding to base distribution probability $\rho_{(u,h)}$ (see equation 4.9 and figure 4-2 for the resulting effect on $\rho_{(u,h)}$). We start by deriving the posterior probability for a given joint setting of $\vec{\lambda}$. We indicate the other sampled variables (including all word analyses and cognate indicators) with an ellipsis "...":

$$\begin{aligned} P(\vec{\lambda} \mid \ldots) &\propto P(\vec{\lambda}) \cdot P(\ldots \mid \vec{\lambda}) \\ &\propto g(\vec{\lambda}) \cdot \frac{\prod_{\text{sub } e} v_e^{[N(e)]}}{\left(\sum_{\text{sub } e} v_e\right)^{[N_S]}} \end{aligned}$$

[9] With the caveat that the cached counts now exclude all instances in the cluster under consideration.

The first factor $g(\vec{\lambda})$ is the unnormalized structural sparsity prior given in equation 4.1. The second factor is the predictive probability of the base-distribution (with Dirichlet prior hyperparameters v_e) when the multinomial parameters (ρ_e) have been marginalized out. As before in equation 4.14, $N(e)$ denotes the number of times edit-operation e has occurred in the *unique* edit-sequences obtained from each morpheme-pair distribution ($G^{stm}, G_1^{pre}, \ldots$); or more precisely, the number of times e appears among the sets:

$$\begin{aligned}
E^{stm} &= \left\{ (\vec{e}_{stm})_{k \neq i} \right\} \\
E_j^{pre} &= \left\{ (\vec{e}_{pre})_{k \neq i} \mid \text{pos}_k = j \right\}, \ \forall j \in 1 \ldots M \\
E_j^{suf} &= \left\{ (\vec{e}_{suf})_{k \neq i} \mid \text{pos}_k = j \right\}, \ \forall j \in 1 \ldots M
\end{aligned}$$

and N_S give the *total* number of substitutions among those sets. The notation $x^{[n]}$ represents the rising factorial: $x(x+1)\cdots(x+n-1)$.

In order to speed mixing of our sampler, we jointly sample blocks of sparsity indicator variables. In particular, for each lost-language letter u, we jointly sample all variables involving u: $\left\{ \lambda_{(u,h)} \mid \forall h \right\}$ (in the binary matrix view, a *row* of $\vec{\lambda}$), and for each known-language letter h, we jointly sample: $\left\{ \lambda_{(u,h)} \mid \forall u \right\}$ (a *column*). To do so, we enumerate all possible values $\vec{\lambda}, \vec{\lambda}', \vec{\lambda}'', \ldots$ (keeping fixed the values for λ_e's not being sampled), compute their posteriors, and sample from the resulting discrete distribution. Note that we can avoid enumerating settings of $\vec{\lambda}$ which are assigned probability zero by the structural sparsity prior (e.g. where $\sum_h \lambda_{(u,h)} > 3$). As a result, we very rarely need consider more than $\binom{30}{2} = 435$ possible values, and often many fewer suffice.

4.7.5 Prediction

The output of our sampler (algorithm 5) is a set of latent variable values sampled from the posterior distribution of our model. In this section we describe how we make our final model predictions on the basis of these samples. As a reminder, sampled variables include:

- Word analyses for each instance i:

$$\Big\{\big[\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}, \mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}, \mathbf{h}^{pre}, \mathbf{h}^{stm}, \mathbf{h}^{suf}, \mathrm{pos}\big]_i^{(r)}\Big\}_{r=1}^{1000}$$

- Cognate indicator variables for each instance i:

$$\Big\{c_i^{(r)}\Big\}_{r=1}^{1000}$$

- Sparsity indicator variables:

$$\Big\{\big\{\lambda_{(u,h)} \mid \forall(u,h)\big\}^{(r)}\Big\}_{r=1}^{1000}$$

We use the following procedure to induce our model's final predictions:

1. Predict the set of sparsity indicator variables $\big\{\lambda_{(u,h)} \mid \forall(u,h)\big\}^{(*)}$ which occurs most frequently among sampled values.

2. Collect sets of *unique* edit-sequences over all sampling rounds:

$$\begin{aligned} E^{stm} &= \big\{\vec{e} \mid \exists(i,r) : (\vec{e}_{stm})_i^{(r)} = \vec{e}\big\} \\ E_j^{pre} &= \big\{\vec{e} \mid \exists(i,r) : (\vec{e}_{pre})_i^{(r)} = \vec{e} \,\wedge\, \mathrm{pos}_i = j\big\},\ \forall j \in 1 \ldots M \\ E_j^{suf} &= \big\{\vec{e} \mid \exists(i,r) : (\vec{e}_{suf})_i^{(r)} = \vec{e} \,\wedge\, \mathrm{pos}_i = j\big\},\ \forall j \in 1 \ldots M \end{aligned}$$

3. Define a base distribution $F^*(\vec{e})$ as in equation 4.14 except with counts now based on the preceding collection of sets (i.e. over all sampling rounds).

4. For each instance i:

   (a) Predict cognate indicator value $c^{(*)}$ which occurs most frequently among sampled values for instance i.

   (b) Predict segmentation $\big[\mathbf{u}^{pre}, \mathbf{u}^{stm}, \mathbf{u}^{suf}\big]^{(*)}$ which occurs most frequently among sampled values for instance i.

(c) Define constraint sets based on the predicted segmentation and the known-language lexicon:

$$\begin{aligned}
C_j^{pre} &= \Big\{\vec{e} \mid y_1(\vec{e}) = [\mathbf{u}^{pre}]^{(*)} \ \wedge\ y_2(\vec{e}) \in \mathcal{H}_j^{pre}\Big\},\ \forall j \in 1 \ldots M \\
C_j^{stm} &= \Big\{\vec{e} \mid y_1(\vec{e}) = [\mathbf{u}^{stm}]^{(*)} \ \wedge\ y_2(\vec{e}) \in \mathcal{H}_j^{stm}\Big\},\ \forall j \in 1 \ldots M \\
C_j^{suf} &= \Big\{\vec{e} \mid y_1(\vec{e}) = [\mathbf{u}^{suf}]^{(*)} \ \wedge\ y_2(\vec{e}) \in \mathcal{H}_j^{suf}\Big\},\ \forall j \in 1 \ldots M
\end{aligned}$$

(d) Predict edit-sequences $[\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}]^{(*)} =$

$$\begin{aligned}
&\operatorname*{argmax}_{\vec{e}_{pre},\vec{e}_{stm},\vec{e}_{suf}} \ F^*(\vec{e}_{pre}) \cdot F^*(\vec{e}_{stm}) \cdot F^*(\vec{e}_{suf}) \\
&\text{s.t. for some } j\text{:} \\
&\quad \vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf} \in C_j^{pre} \times C_j^{stm} \times C_j^{suf}
\end{aligned} \tag{4.18}$$

(e) Predict cognate:

$$\big[\mathbf{h}^{pre}\,\mathbf{h}^{stm}\,\mathbf{h}^{suf}\big]^{(*)} = y_2\big([\vec{e}_{pre}]^{(*)}\big)\ y_2\big([\vec{e}_{stm}]^{(*)}\big)\ y_2\big([\vec{e}_{suf}]^{(*)}\big)$$

We compute the constrained maximum in equation 4.18 using the same finite-state machines used during sampling. Recall that we construct $A_j^{stm}(\mathbf{u})$ to be a WFSA for which every path $\vec{a}$ corresponds to an edit-sequence $\vec{e}$ with $y_1(\vec{e}) = \mathbf{u}$ and $y_2(\vec{e}) \in \mathcal{H}_j^{stm}$. We now weight the arcs so that $\prod_{a \in \vec{a}} w(a) = F^*(\vec{e})$. Thus computing the constrained maximum for sets $C_j^{pre}, C_j^{stm}, C_j^{suf}$ requires constructing corresponding finite-state acceptors A_1, A_2, A_3 and computing:

$$\max_{\vec{a} \in A_1} \prod_{a \in \vec{e}} w(a) \ \cdot\ \max_{\vec{a} \in A_2} \prod_{a \in \vec{e}} w(a) \ \cdot\ \max_{\vec{a} \in A_3} \prod_{a \in \vec{e}} w(a)$$

To find the maximum weight path through each WFSA, we use the same dynamic program given in equation 4.16, merely replacing the summation with a maximum:

$$\beta_k = \begin{cases} 1 & \text{if } s_k \text{ is an end state,} \\ \max_{a\,:\,s_k \to s_\ell} w(a) \cdot \beta_\ell & \text{otherwise.} \end{cases} \tag{4.19}$$

Intuitively, β_k gives the maximum path-weight of paths starting at state s_k and ending at an end-state. Values can again be computed by starting with the end-states and progressing backwards along arcs. The resulting value β_0, corresponding to the start-state s_0, gives the desired maximum. To retrieve the actual maximizing path, backpointers are stored during each step of the dynamic program.

4.7.6 Implementation Details

This section describes implementation details that are necessary to reproduce our experiments.

Computational Details

Most steps in our sampling algorithm simply require us to compute counts over our currently drawn sample of variables. Instead of recomputing these values on the fly, we keep a persistent cache of counts which is incrementally updated after each sampling step. Thus, most sampling steps simply require a constant-time lookup in a hashtable and a quick computation of probabilities.

When sampling lost-language word segmentations, we do explicitly enumerate all possible segmentations and parts-of-speech. However, we use an extremely coarse notion of part-of-speech (only 4, see 4.8.1 for details) and we cap the length of prefixes and suffixes to three characters (in line with what we observe for the known-language prefix and suffix length). Thus, even for very long words, we need only consider $4^3 = 64$ possibilities.

Many steps during sampling require the construction of weighted finite-state automata and the computation of dynamic programs over these automata. Each WFSA is of polynomial-size in the length of the corresponding lost-language morpheme $\mathbf{u}$. The required dynamic programs are all linear-time in the size of the corresponding WFSA. The construction of each WFSA, however, can be expensive, as it requires intersecting a lost-language morpheme WFSA $A(\mathbf{u})$ with a much larger WFSA representing the known-language lexicon. In order to avoid unnecessary computation, we cache the resulting WFSA and store it for future use (simply reweighting the arcs as necessary).

For example, the first time we compute probabilities for lost-language prefix $\mathbf{u}^{pre}$ with part-of-speech j, we intersect $A(\mathbf{u}^{pre})$ with A_j^{pre}. We use the resulting WFSA and store it in our cache. The next time we encounter prefix $\mathbf{u}^{pre}$ with part-of-speech j, we retrieve the WFSA and reweight its arcs according to our current cache of counts.

Initialization and Pruning

We initialize our latent variables with results from the HMM baseline (see 4.9). In particular, the baseline provides us with letter substitution probabilities $P(h|u)$. First, we prune our search space by ruling out all substitutions (u, h) which are given probability < 0.05. We then initialize insertion probabilities $P(u)$ based on the frequencies of known-language letters. As another pruning step, we rule out insertions of all but the two most frequent letters. The result is a string-edit distribution $S(\vec{e})$. We then use this distribution to initialize our word analyses. In particular, for each word $\mathbf{u}_i$, we consider all segmentations and parts-of-speech, and for each compute the constrained maximum given in equation 4.18 (replacing $F^*(\vec{e})$ with our initializing distribution $S(\vec{e})$). We initialize $\mathbf{u}_i$ with the resulting values. As a final pruning step, the character substitutions for each letter u are restricted to a single letter h, if after the initialization round (u, h) is found to occur more than five times more than any (u, h').

We initialize all cognate indicator variables c_i to 1, and we initialize the sparsity indicator variables $\lambda_{(u,h)}$ to be the character-mapping predictions made by the baseline (in particular, Baseline 1 in section 4.9).

Hyperparameter Values

Finally, we list all the values used for fixed hyperparameters:

- $\vec{b} = (b_i = 0, \forall i)$ // Structural sparsity parameters (equation 4.2)
- $\vec{b'} = (b'_0 = 0, b'_1 = 0, b'_2 = 7, b'_3 = 1, b'_{>3} = 0)$
- $\vec{w} = (w_0 = -\infty, w_1 = 0, w_2 = -50, w_{>2} = -\infty)$
- $K = 50$ // Spike-and-slab parameter (equation 4.9)
- $q(N_I, N_D) = \begin{cases} 0.5 & \text{if } N_I, N_D = (0,0) \\ 0.5 & \text{if } N_I, N_D = (1,0) \\ 0 & \text{otherwise} \end{cases}$ // Base distribution (equation 4.4)
- $\alpha_0 = 1000$ // Concentration parameter (section 4.6.2)
- $P(c_i = 1) = 0.5$ // Cognate prior (section 4.6.2)

4.8 Experiments

In this section we describe experiments applying our model to the ancient Ugaritic language (see section 4.4 for background) with biblical Hebrew as the observed known language. In section 4.8.1 we describe the Ugaritic corpus and gold-standard annotations; in section 4.9 we describe our evaluation tasks and baseline, and in the remaining sections we describe our various experiments and results.

4.8.1 Corpus and Annotations

Our undeciphered corpus consists of an electronic transcription of the Ugaritic tablets [28]. This corpus contains 7,386 unique word-forms. As our known language corpus, we use the Hebrew Bible, which is both geographically and temporally close to Ugaritic. To extract a Hebrew morphological lexicon we assume the existence of manual morphological and part-of-speech annotations [45]. We divide Hebrew

	Number of Hebrew Cognates						
	0	1	2	3	4	5	6
Number of Ugaritic Word-forms	5172	1677	359	160	14	1	3

Table 4.1: Number of Ugaritic word-forms with various numbers of identified Hebrew cognates.

stems into four main part-of-speech categories each with a distinct affix profile: Noun, Verb, Pronoun, and Particle. For each part-of-speech category, we automatically determine the set of allowable affixes using the annotated Bible corpus.

To evaluate the output of our model, we annotated the words in the Ugaritic lexicon with the corresponding Hebrew cognates found in the standard reference dictionary [30]. In addition, manual morphological segmentation was carried out with the guidance of a standard Ugaritic grammar [101]. Although Ugaritic is an inflectional rather than agglutinative language, in its written form (which lacks vowels) words can easily be segmented (e.g. *wypltٖn* becomes *wy-plṭ-n*). Note that our analysis allows only a single prefix or suffix string, and as a result multiple prefixes or suffixes are combined into a single string.

Overall, we identified Hebrew cognates for 2,214 word forms, covering almost one-third of the Ugaritic vocabulary. We are confident that a majority of Ugaritic words with known Hebrew cognates were thus identified. The remaining Ugaritic words include many personal and geographic names, words with cognates in other Semitic languages, and words whose etymology is uncertain.

Since our annotation was performed at the vocabulary-level rather than the text-level, we faced the problem of word ambiguity. A single Ugaritic word-form can often be identified with several potential Hebrew cognates, depending on actual context. For example, the Ugaritic word-form *bth* could be identified with at least four Hebrew cognates: *bth* (her daughter), *btw* (his daughter), *byth* (her house), or *bytw* (his house). In such cases, we annotated the Ugaritic word form with all Hebrew cognate possibilities. Table 4.1 gives the number of Ugaritic word-forms with various numbers of identified Hebrew cognates.

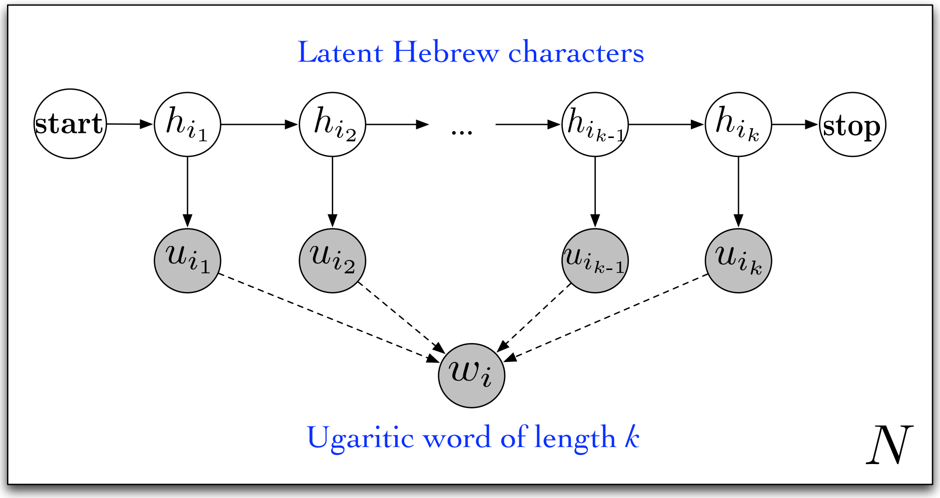


Figure 4-4: **Plate diagram of the baseline HMM model** (shown here as a first-order rather than second-order HMM for simplicity). Each of N observed Ugaritic word-forms w_i is determined by its observed character sequence. Each such character is generated by a latent Hebrew letter. Hebrew character transition distributions are estimated directly from transition counts in the Hebrew Bible. Emission distributions and the latent Hebrew characters are estimated using EM.

4.9 Results

In the following section we evaluate our model on three separate decipherment tasks: (i) Learning alphabetic mappings, (ii) Cognate decipherment, and (iii) Cognate identification.

As a baseline for these tasks, we use the HMM-based method of [66] for learning letter substitution ciphers. In its original setting, this model was used to automatically map the written form of a language to its spoken form, under the assumption that each written character was emitted from a hidden phonemic state. In our adaptation, we assume instead that each Ugaritic character was generated by a hidden Hebrew letter, and that each hidden Hebrew letter was generated by the previous two Hebrew letters (a second-order character-level HMM). See figure 4-4 for a graphical depiction of this baseline. Hebrew character trigram transition probabilities are estimated using counts from the Hebrew Bible, and Hebrew to Ugaritic character emission probabilities are learned by applying EM to the Ugaritic vocabulary (see [66] for details). Finally, the highest probability sequence of latent Hebrew letters is predicted for each Ugaritic word-form, using the Viterbi algorithm.

4.9.1 Alphabetic Mapping

The first essential step towards successful decipherment is recovering the mapping between the symbols of the lost language and the alphabet of a known language. Although the exact phonetic values of letters in ancient scripts can never be known with complete certainty, it is possible to recover the historical relationships between phonemes (and thus alphabets) of related languages using the comparative method [19]. As a gold standard for our comparison, we use the well-established historical relationship between the sounds of the Ugaritic and Hebrew alphabets [55]. In particular, we wish to automatically recover pairs of *reflexes* in two languages – that is, pairs of letters whose corresponding phonemes descend from a common phoneme in an ancestral language (in this case Proto-Semitic).

This mapping is not one-to-one but is generally quite sparse. Of the 30 Ugaritic symbols, 27 map almost exclusively to a single Hebrew letter, and the remaining three map to two Hebrew letters. The Hebrew alphabet contains 23 letters, of which three map to three Ugaritic letters, four map to two Ugaritic letters, and the remainder map to a single Ugaritic letter. See table F.2 in appendix F for the gold standard alphabetic mapping used in evaluation.

We recover our model's predicted alphabetic mappings by simply examining the predicted values of the binary indicator variables $\lambda_{u,h}$ for each Ugaritic-Hebrew letter pair (u, h). Due to our structural sparsity prior $P(\vec{\lambda})$, the predicted mappings are quite sparse: all Ugaritic letters maps to a single Hebrew letter, 17 Hebrew letters map to a single Ugaritic letter, five Hebrew letters map to two Ugaritic letters, and one Hebrew letter maps to three Ugaritic letters. See table F.3 for the predicted alphabetic mappings.

To recover alphabetic mappings from the HMM substitution cipher baseline, we consider several possibilities.

Baseline 1: For each Hebrew letter h, we can simply choose the single Ugaritic letter u with highest emission probability: $u = \text{argmax}_{u'} P(u'|h)$. Table F.5 gives the predictions under this baseline. Notice that this procedure results in many Ugaritic letters that are not mapped to any Hebrew letter at all.

Baseline 2: For each Ugaritic letter u, we can simply choose the single Hebrew letter h such that: $h = \text{argmax}_{h'} P(h'|u) \propto P(u|h')P(h')$. Table F.4 gives the predictions under this baseline. This procedure guarantees that all Ugaritic letters are mapped to a single Hebrew letter. However, three Hebrew letters remain unmapped.

Baseline 3: This procedure simply combines the procedures of Baselines 1 and 2. A mapping between (u, h) is predicted if (u, h) are mapped under either (or both) of the first two baselines. This procedure results in a many-to-many mapping where every letter in each alphabet is guaranteed to map to at least one letter in the other. Table F.6 gives the predictions under this baseline.

To evaluate these predicted mappings, we consider several metrics. Our first, somewhat crude, measure is to simply count the number of Ugaritic letters that are correctly mapped to at least one of their Hebrew reflexes. Under this metric our model recovers correct mappings for 28 of 30 Ugaritic letters (yielding 93.3% accuracy), while the best baseline predictions (Baselines 2 and 3) yield correct mappings for 23 of 30 Ugaritic letters (76.7% accuracy).

Note that this first evaluation metric ignores the fact that the gold-standard mappings are many-to-many (though quite sparse). We can evaluate performance with greater sensitivity by instead treating each mapped character pair (u, h) as a positive prediction in a binary classification problem. Under this scenario, the gold-standard contains 33 positive examples out of 690 possible letter pairings. Results under both this and the first metric are given in table 4.2. Our model yields performance superior to the baselines on all measures, achieving F1-measure of .89, compared to .73 for the best baseline variant.

	ACCURACY	PRECISION	RECALL	FI-MEASURE
Baseline 1	.57	.74	.52	.61
Baseline 2	.77	.77	.70	.73
Baseline 3	.77	.69	.73	.71
Our Model	.93	.93	.85	.89

Table 4.2: **Evaluation of alphabetic mappings** predicted by HMM baseline variations and our model. Column one (ACCURACY) simply counts the number of Ugaritic letters that are correctly mapped to at least one of their Hebrew reflexes. Columns 2-4 treats each possible character pair as an example in a binary classification problem.

4.9.2 Cognate Decipherment

One of the primary goals of lost language decipherment is to accurately translate and understand ancient texts. An important step in this process is the recovery of cognate pairs between the lost language and a known related language. Cognate are words in sister languages that descend from a common word in a shared ancestral language. As such, cognates are often accurate translations of one another, or at least share common semantic features. As detailed in section 4.8.1, we manually identified gold-standard Hebrew cognates for about one-third of the Ugaritic word-forms. When multiple gold-standard cognates exist for an Ugaritic word-form (see table 4.1) we evaluate predictions using the Hebrew cognate which yields the best performance.

Cognate predictions for our model are produced by examining the predicted values of the latent Hebrew morphemes associated with each Ugaritic word-form. Even for words where we predict a cognate indicator variable $c_i = 0$ (indicating that the word does not have a Hebrew cognate), we still predict the most likely latent Hebrew word $\mathbf{h}_i^*$ by conditioning on $c_i = 1$. For the HMM baseline, we simply predict the most likely latent sequence of Hebrew characters for each Ugaritic word-form using the Viterbi algorithm.

We evaluate these cognate predictions using several measures. The simplest measure simply counts how many predicted Hebrew cognates exactly match one of the gold-standard cognates. As seen in table 4.3 (WORDS → ACCURACY → TYPE), our

	WORDS				MORPHEMES			
	ACCURACY		EDIT-DISTANCE		ACCURACY		EDIT-DISTANCE	
	TYPE	TOKEN	TYPE	TOKEN	TYPE	TOKEN	TYPE	TOKEN
Baseline	.288	.460	1.261	.878	n/a		n/a	
Our Model	.630	.697	.501	.400	.763	.838	.337	.239
Prev Version	.604	.683	.552	.450	.740	.813	.369	.270
Only Sub	.599	.683	.529	.414	.745	.828	.323	.216
Only Mapped	.567	.648	.555	.436	.731	.799	.366	.276
No Cogs	.447	.560	.802	.660	.677	.760	.448	.341
No Spike	.471	.577	.722	.587	.710	.787	.400	.302
No Morph	.363	.554	1.175	.779	n/a		n/a	
Know Cogs	.710	.787	.422	.314	.834	.892	.231	.152

Table 4.3: **Evaluation of cognate decipherments** for the HMM baseline as well as our model. Two evaluation measures are used: 0-1 accuracy (higher is better) and Levenshtein edit-distance (lower is better). The unit of prediction can be either complete words or each of their three morphemic parts (prefix, stem, and suffix). Evaluation is carried out both at the word-form (TYPE) level, ignoring word frequency, as well as at the token-level, taking frequency into account. The first two rows show results for the baseline and our full model. The final seven rows show results for variants of our model (see section 4.9.4 for details).

model achieves 63% accuracy on this measure, while the baseline achieves accuracy of 28.8%.

To gain a finer sense of how close our predictions are to the true Hebrew cognates, we also measure the Levenshtein edit-distance [72] between predicted and gold-standard cognates. This distance metric gives the minimal number of character edit-operations (substitutions, insertions, and deletions) needed to make an input string identical to a reference string. As seen in table 4.3 (WORDS → EDIT-DISTANCE → TYPE), our model's predictions are on average .5 edit-operations away from a gold-standard cognate, whereas the baseline predictions require, on average, 1.26 edit-operations to match a true cognate.

Since our model also predicts prefix-stem-suffix morpheme boundaries, we can evaluate its decipherment performance on a per-morpheme basis as well. This metric can be both stricter and more lenient than per-word accuracy. For example, if *b-bt-w* were predicted in place of the correct *b-byt-w*, two out of three morpheme predictions would be judged as correct, whereas the per-word measure would judge this prediction as fully incorrect. On the other hand, we also now require that the morphemic segmentation boundaries be correct. For example, a prediction of *bbytw* (no prefix or suffix) would be judged as correctly predicting *none* of the morphemes of the correct *b-byt-w*, whereas the per-word evaluation would count this as a fully correct prediction.

As table 4.3 shows, per-morpheme performance (both in terms of simple accuracy and edit-distance) is consistently better than per-word performance. In fact, our model correctly deciphers over 3/4 of the morphemes on all Ugaritic word-forms with Hebrew cognates (MORPHEMES → EDIT-DISTANCE → TYPE). This metric is not available for the baseline, as it does not predict morpheme boundaries.

Besides carrying out these evaluations at the word-form-level (TYPE in table 4.3), we also investigated whether predictions were are more or less accurate for frequent words by evaluating at the token-level as well (TOKEN) in table 4.3). As shown, performance improves across the board at the token-level, indicating that more frequent words are easier to decipher than their infrequent counterparts.

4.9.3 Cognate Identification

The previous section evaluated our model's ability to decipher Ugaritic words which are, in fact, cognate to one or more Hebrew words. In this section we consider the problem of identifying which Ugaritic words have such cognates.

As before, we use our annotated Ugaritic corpus as a gold-standard (see section 4.8.1 and in particular table 4.1). About one-third of Ugaritic word-forms were identified as having known Hebrew cognates. We note that this is a conservative gold-standard. Our knowledge of ancient Hebrew comes almost exclusively via the texts of the Hebrew Bible (about 300,000 tokens) and there are certainly many ancient Hebrew words which have been lost to history. In addition, it is possible that some Hebrew cognates were missed during the the annotation process (though we are confident that the great majority were included).

We evaluate our model's ability to identify cognates using the predicted values of the indicator variables c_i. Note also that for Ugaritic word-form $\mathbf{u}_i$: $P(c_i|\mathbf{u}_i) \propto$

$P(\mathbf{u}_i|c_i) \cdot P(c_i)$. Since $P(c_i)$ (the cognate prior) is a fixed Bernoulli parameter of our model, we can vary its value to achieve different cognate prediction thresholds, allowing a trade-off between precision and recall.

As before, we compare our performance against the HMM substitution cipher baseline. To produce baseline cognate identification predictions, we calculate the probability (using the learned emission and transition parameters) of each Ugaritic word $\mathbf{u}_i$ given the latent Hebrew letter sequence predicted by the HMM. This probability can be regarded as measuring the likelihood that the given Ugaritic word-form $\mathbf{u}_i$ was generated by a latent Hebrew cognate: $P(\mathbf{u}_i|c_i = 1)$.[10] The probability that $\mathbf{u}_i$ was generated as a lone Ugaritic word, $P(\mathbf{u}_i|c_i = 0)$, is simply given by $(\frac{1}{31})^{length(\mathbf{u}_i)+1}$ (assuming a uniform distribution over the 30 Ugaritic letters and a special end symbol). Finally, as in our model, we assume a fixed cognate prior $P(c_i)$ and predict that $\mathbf{u}_i$ has a Hebrew cognate if:

$$\frac{P(\mathbf{u}_i|c_i = 1)}{P(\mathbf{u}_i|c_i = 0)} > \frac{P(c_i = 0)}{P(c_i = 1)}$$

As for our model, when the prior $P(c_i = 1)$ is set higher, we will detect more true cognates, but the number of false positives increases as well.

Finally, we compare our model's performance to that of our previously published version [109]. The primary difference between the model presented in this thesis and the previously published version is how we model the generation of non-cognate Ugaritic words: $P(\mathbf{u}_i|c_i = 0)$. In the previous publication we used a simple uniform distribution Ugaritic language model:

$$P(\mathbf{u}_i|c_i = 0) = \left(\frac{1}{31}\right)^{length(\mathbf{u}_i)+1} \tag{4.20}$$

Since our cognate generation sub-model $P(\mathbf{u}_i|c_i = 1)$ (detailed in section 4.6.2) leads to an exponential distribution over the length of Ugaritic words with latent Hebrew cognates, it is important that $P(\mathbf{u}_i|c_i = 0)$ display the same exponential decay on

[10] Or more precisely, this quantity should be regarded as a Viterbi approximation to $P(\mathbf{u}_i|c_i = 1)$ as it only accounts for the highest probability latent Hebrew letter sequence.

word length. Otherwise, the dominating factor in predicting c_i would be the length of $\mathbf{u}_i$ rather than the intrinsic plausibility of its joint generation with a Hebrew counterpart.

While definition 4.20 helps us avoid a length bias, it ignores the frequency of the Ugaritic characters in word $\mathbf{u}_i$. In contrast, the posterior of the cognate generation sub-model $P(\mathbf{u}_i|c_i = 1)$ is quite sensitive to the observed frequencies of the characters composing $\mathbf{u}_i$, generally assigning far lower probability to Ugaritic words with infrequent characters.[11] As a result, the previous version of our model displayed an unfortunate bias towards predicting that Ugaritic words with infrequent characters had no Hebrew cognates. To correct this bias, the model described in this thesis defines the distribution over non-cognate Ugaritic words using a unigram (rather than uniform) Ugaritic language model:

$$P(\mathbf{u}_i|c_i = 0) = P(END) \cdot \prod_{0<j\leq length(\mathbf{u}_i)} P\left(u_i[j]\right) \tag{4.21}$$

This definition allows a trade-off between $P(\mathbf{u}_i|c_i = 0)$ and $P(\mathbf{u}_i|c_i = 1)$ which focuses on the inherent posterior plausibility of $\mathbf{u}_i$ having been generated along with a Hebrew word $\mathbf{h}_i$ (i.e. in terms of consistent of character substitutions and morpheme matchings) rather than the length or character frequency of $\mathbf{u}_i$.

As cognate identification is a binary classification problem, we evaluate performance by plotting Receiver Operator Characteristic (ROC) and Precision-Recall curves. An ROC curve shows the achievable trade-offs between the False Positive Rate, defined as the fraction of all positive predictions which are actually negative

$$FPR = \frac{FP}{TP + FP},$$

and the True Positive Rate, defined as the fraction of all positive instances which

[11] This effect stems from the string-edit base distribution, which can be thought of as a character-level language model over bilingual character pairs.

are correctly predicted as positive:

$$TPR = \frac{TP}{TP + FN}.$$

A Precision-Recall curve shows the achievable trade-offs between Precision, defined as the fraction of all positive predictions which are in fact positive

$$Precision = \frac{TP}{TP + FP},$$

and Recall, which is identical to the True Positive Rate.

Figure 4-5 shows the ROC curve for our model, the ACL 2010 version of our model, and the HMM baseline. All three models achieve better than random performance for all possible operating points. The ROC curve of our previous model dominates the the curve of the HMM baseline, and likewise our current model dominates the performance of our previous model.

Figure 4-6 shows the Precision-Recall curve for the same three models. All three models show fluctuations in Precision when Recall is set very low. This may be due to the fact that the number of positive predictions in this setting is very small so the addition of a single false positive or true positive can have a great effect on Precision. For values of Recall above 0.1, the trend we observed in the ROC setting reemerges: Our previous model dominates the HMM baseline, and our current model dominates our previous model.

To gain some further insight we can graph F1-Measure as a function of Recall. F1-Measure is defined as the harmonic mean of Precision and Recall and is often used as a unified measure of binary classification performance:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Figure 4-7 shows the various achievable F1 scores for the three models under consideration. The respective maximum achievable F1 scores are .473, .522, and .563 for the HMM baseline, our previous model, and our current model.

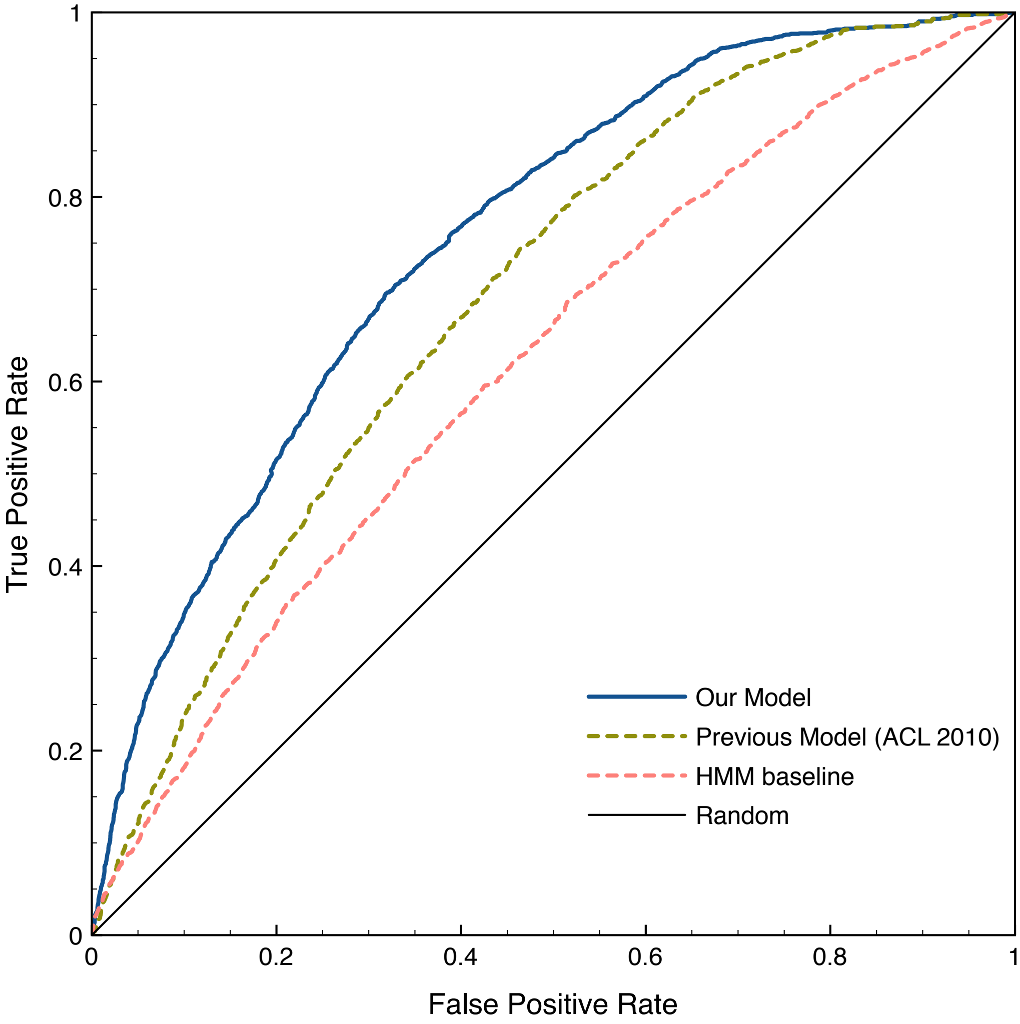


Figure 4-5: **Cognate identification ROC curves** for the model presented in this thesis (Our Model), the version of our model presented at ACL 2010 (using a uniform character model for non-cognate probabilities), and the HMM baseline.

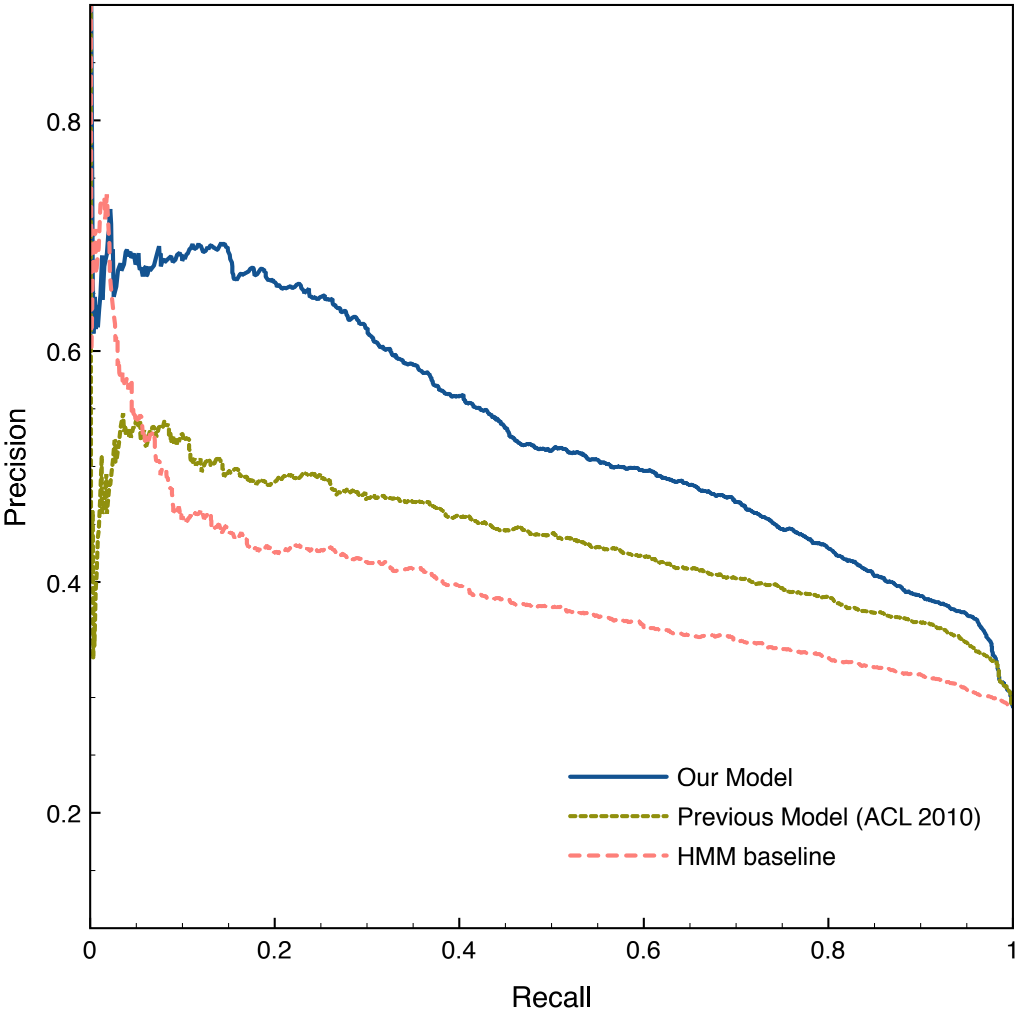


Figure 4-6: **Cognate identification Precision-Recall curves** for the model presented in this thesis (Our Model), the version of our model presented at ACL 2010 (using a uniform character model for non-cognate probabilities), and the HMM baseline.

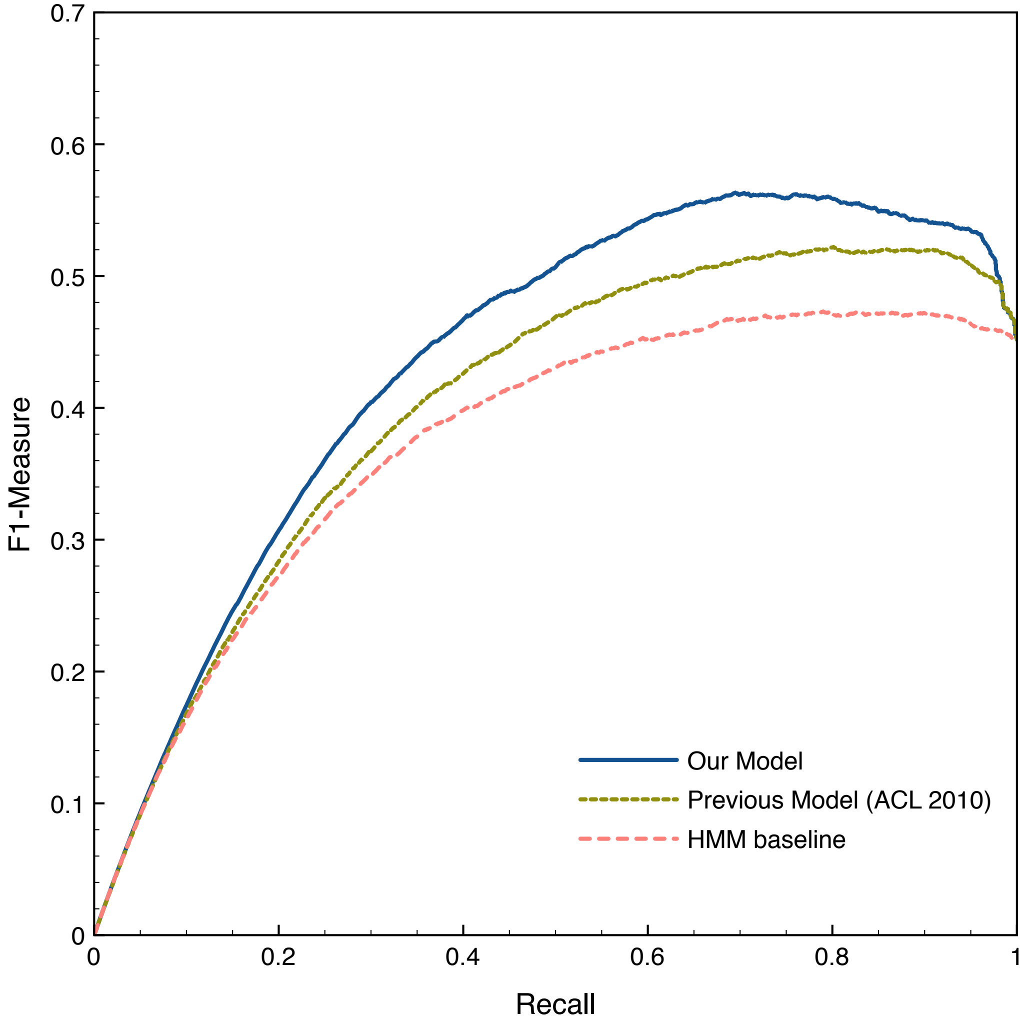


Figure 4-7: **Cognate identification F1 curves** for the model presented in this thesis (Our Model), the version of our model presented at ACL 2010 (using a uniform character model for non-cognate probabilities), and the HMM baseline.

4.9.4 Comparison to Model Variants

In this section we describe experiments comparing the performance of our full model to different variants. We focus on the performance of these variants on the task of cognate decipherment (section 4.9.2 above). Below we list each model variant and discuss its performance. Results are shown in the final rows of table 4.3.

Previous Version: This is the variant of our model originally published [109]. The key difference is that this older version models non-cognate word generation, $P(\mathbf{u}_i|c_i = 0)$, using a uniform distribution over Ugaritic letters. The model presented in this thesis, however, uses a unigram character distribution, taking into account letter frequency. As discussed in section 4.9.3, this leads to more accurate predictions of the cognate indicators c_i. In fact, as table 4.3 shows ("Prev Version"), this difference also leads to better performance on the task of cognate decipherment. While the previous version achieved 60.4% accuracy (WORDS → ACCURACY → TYPE), the current model yields 63% accuracy.

Only Substitutions: In this variant of our model, the only string-edit operation allowed is character substitution. All insertions and deletions are given zero probability. This allows a more precise comparison to the HMM baseline, which also only allows a one-to-one mapping between characters.

The results of this variant are reported in table 4.3 (**Only Sub**). Perhaps surprisingly, strictly requiring one-to-one character substitutions has little effect on performance. Word type prediction accuracy falls from 63% (when insertions are permitted) down to to 59.5%. Even smaller differences are seen across the other performance metrics. This small difference is partially due to the fact that substitutions are by far the dominant edit operation. Of equal importance, though, is the fact that our full model simply doesn't model insertions very effectively. As we discuss below, our model's biggest source of error involves missing a character insertion for a very common suffix.

Only Mapped: Recall that our model predicts alphabetic mappings via the structural sparsity indicator variables $\{\lambda_{(u,h)}\}$. These variables lead to higher prior probability being assigned to substitution (u, h) when the corresponding $\lambda_{(u,h)} = 1$. When predicting latent cognates, however, other substitutions do sometimes still occur, even when the corresponding $\lambda_{(u,h)} = 0$.

In this model variant, we assess the impact of allowing these substitutions. We run our sampling algorithm as usual, but at prediction time (section 4.7.5), we *only* allow substitution (u, h) to occur if $\lambda_{(u,h)} = 1$. As table 4.3 shows (**Only Mapped**), this restriction leads to lower performance on all our evaluation metrics. This result indicates that although our model is effective at predicting alphabetic mappings, it still benefits from allowing "unauthorized" substitutions to occur occasionally. In fact, an analysis shows that the most common of these substitutions is for the letter pair (š, š) (i.e. Hebrew שׁ substituting for Ugaritic 𐎌), which is actually a false-negative letter mapping prediction.

In the next three experiments, we explore the relative contribution of various components of our model. In each case, we remove one model component and report the resulting performance.

No Cognate Indicators: In this experiment, we test the performance of our model when *all* lost-language words are assumed to have cognates. In other words, the model is not allowed to "explain away" difficult word-forms by setting the corresponding cognate indicator to 0. This is achieved experimentally by simply setting the cognate prior $P(c_i = 1)$ to 1. As table 4.3 indicates (**No Cogs**), this leads to a serious degradation of performance across all measures. Thus, it seems crucial to allow the model to ignore word-forms which don't allow consistent mappings, even though the cognate identifications themselves are imperfect.

No Spike-and-slab: We test the performance of our model when the structural sparsity prior (section 4.6.2) is removed. Recall that the purpose of this prior was to ensure that predicted alphabetic mappings obey the following intuition: that each letter map to a very limited number of letters in the other language. In this experiment we test the importance of this intuition. In particular, the value of K in equation 4.9 is set to 1 in this experiment. The result is that the structural sparsity indicator variables $\vec{\lambda}$ are essentially ignored. As table 4.3 shows (**No Spike**), performance in the absence of these variables degrades quite seriously. This finding confirms that incorporating the intuition of alphabetic-mapping sparsity is quite crucial for achieving high performance.

No Morphology: We test the performance of our model when no morphological segmentation is performed. Instead of segmenting Ugaritic words and matching the resulting morphemes to latent Hebrew morphemes, we instead match entire Ugaritic word-forms to entire Hebrew word-forms. This is achieved experimentally by setting the Hebrew prefix and suffix lexicons to the empty set, and setting the stem lexicon to the set of *entire* Hebrew words. As table 4.3 indicates (**No Morph**), this variant achieves only 36% accuracy on word types, far below that of all the other variants. This finding confirms that, just as for humans, morphological awareness is one of the key ingredients of success for computational decipherment.

Knowing Hebrew Cognates: Finally, we test the performance of our model when the Hebrew vocabulary is restricted to those morphemes which *actually* occur as cognates with Ugaritic words. The identity of these morphemes would not be known in a realistic decipherment scenario. Nevertheless, one could imagine a separate model which first predicts which Hebrew morphemes and words are likely to have Ugaritic cognates. This model could exploit the general tendencies of languages to preserve certain words and could also examine the cross-linguistic evidence given by cognates among known Semitic languages (Aramaic, Arabic, Akkadian, etc). The experiment presented here is intended to show our model's performance in the most ideal of situations, where the set of cognate morphemes is known *exactly.* The results given in table 4.3 (**Know Cogs**) indicate, perhaps unsurprisingly, that our model yields significantly improved performance in this scenario. Accuracy on word types reaches 71%.

4.9.5 Combining Model Variants

To further tease out the contributions of each component of our model, we consider various combinations of the above variants. First we consider a model very similar to the HMM baseline: All Ugaritic words are assumed to be cognates, no spike-and-slab prior is used, and only character substitutions are allowed (i.e. **Only Substitutions + No cognate Indicators + No Spike-and-slab**). The chief difference between this variant and the HMM baseline itself is that the latter attempts to match Ugaritic and Hebrew and the character trigram level, whereas our model matches the languages at the morphemic level. The results for this model are given in table 4.4 (**Combo 1**). The cognate decipherment accuracy at the word-type level drops to 35%, from 63% for the full model. However, we note that performance is still above that of the HMM baseline (29%). This result indicates that matching the lost and known languages at the morphemic level can indeed be more powerful than character-level matching alone.

Next, we drop the use of morphology as well (i.e. **Only Substitutions + No Cognate Indicators + No Spike-and-slab + No Morphology**). Now our model simply tries to find entire Hebrew words which match entire Ugaritic words, with a consistent character-level mapping. The results here are drastically worse. As table 4.4 indicates (**Combo 2**), the word-type decipherment accuracy for this model is only 21%, far below that of the baseline. This finding confirms once again the importance of morphological-level analysis for decipherment. Finally, **Combo 3** and **Combo 4** mirror the first two combinations, except that character insertions are now allowed. In both cases, allowing such insertions leads to about 2 percentage points of improved accuracy.

	WORDS				MORPHEMES			
	ACCURACY		EDIT-DISTANCE		ACCURACY		EDIT-DISTANCE	
	TYPE	TOKEN	TYPE	TOKEN	TYPE	TOKEN	TYPE	TOKEN
Combo 1	.345	.491	.976	.744	.620	.737	.480	.333
Combo 2	.210	.270	1.61	1.20	.418	.595	1.01	.632
Combo 3	.361	.483	.984	.785	.635	.727	.468	.348
Combo 4	.230	.283	1.55	1.20	.414	.592	1.04	.657
Baseline	.288	.460	1.261	.878	n/a		n/a	
Our Model	.630	.697	.501	.400	.763	.838	.337	.239

Table 4.4: Combinations of the variants given in table 4.3. **Combo 1** only allows substitutions, assumes all words have cognates, and does not employ the structural sparsity prior (**Only Substitutions + No Cognate Indicators + No Spike-and-slab**). **Combo 2**, in addition, removes morphological analysis from the model (**Only Substitutions + No Cognate Indicators + No Spike-and-slab + No Morphology**). **Combo 3** allows all string edits, but assumes that all words have cognates, and does not use the structural sparsity prior (**No Cognate Indicators + No Spike-and-slab**). **Combo 4**, in addition, removes morphological analysis (**No Cognate Indicators + No Spike-and-slab + No Morphology**). The baseline and full model performance are repeated here for easy comparison.

4.9.6 Related Language Discovery

One of the key assumptions we have made throughout this chapter is that the lost language is related to a known language, and that the known language has been identified. In the case of Ugaritic at least, human decipherers immediately surmised that Ugaritic was likely to be a Semitic language, due to the dating and geographical location of the discovered clay tablets. However, in many other cases, the identity of a known, related language is far from certain. The Linear B script, for example, was discovered to encode an early form of Greek only after 50 years of decipherment efforts. The currently undeciphered Indus Valley symbol system remains even more mysterious. Although some scholars believe that it represents an early Dravidian language, others have argued that it is not likely to encode any spoken language.

For future statistical decipherment efforts we clearly need to move beyond the assumption that a known, related language has been clearly identified. This line of thought is largely beyond the scope of this thesis. However, we did run one final experiment to test whether our model can at least *distinguish* between a related

	*A*verage entropy of...		
	$P_u(h)$	$P_h(u)$	$P_u(h)\&P_h(u)$
Ugaritic-Hebrew	0.43	0.56	0.48
Ugaritic-English	1.63	1.51	1.58

Table 4.5: **Cross-character entropy** when the known language is Hebrew versus English. Column one gives the average entropy of known-language letters for all Ugaritic letters. Column two gives the average entropy of Ugaritic letters for all known-language letters. Column three averages the entropies over letters from both alphabets.

and an unrelated known language.

In particular, we have applied our model to the decipherment of Ugaritic using *English* as the known language. In fact, of course, no known cognate pairs exist between Ugaritic (a Semitic language) and English (a Germanic language with significant Romance influence). The question we pose is the following: Can we automatically distinguish between our system output when using an actual related language (Hebrew) and our system output when using a non-related language (English).

To make our English lexicon as comparable as possible to the Hebrew lexicon, we base it on an English translation of the Hebrew Bible [94]. We use the Stanford Tagger [117], version 1.6[12] to part-of-speech tag the English corpus. We map the predicted parts-of-speech to five inflectional categories: adjective, adverb, noun, verb, and particle (non-inflectional). We use the Porter2 stemming algorithm[13] to induce an inventory of stems and suffixes for each of these categories.

We tested two criteria for automatically distinguishing between Hebrew and English output. In both cases, we consider predictions for *all* Ugaritic word-forms (not just those that actually have Hebrew cognates).

Cross-character Entropy

For both English and Hebrew the predicted indicator variables $\{\lambda_{(u,h)}\}$ will themselves be sparse (simply due to the structural sparsity prior). However, as noted

[12] `http://nlp.stanford.edu/software/tagger.shtml`

[13] `http://snowball.tartarus.org/algorithms/english/stemmer.html`

in section 4.9.4, even when $\lambda_{(u,h)} = 0$, the corresponding substitution (u, h) will sometimes be used in cognate predictions. We hypothesize that if a large number of cognates truly exist between the lost and known languages, then character substitutions will display much greater regularity than would otherwise be possible. In other words, the actual substitutions used in cognate predictions will be far more sparse if the languages are truly related.

We measure this sparsity by computing the *cross-character entropy* of the predicted cognates. For each lost-language character u, we compute an empirical distribution over known-language characters h:

$$P_u(h) = \frac{N(u,h)}{\sum_{h'} N(u,h')},$$

where $N(u, h)$ denotes the number of times the substitution (u, h) appears in the final cognate decipherment predictions. Likewise, for each known-language character h we compute:

$$P_h(u) = \frac{N(u,h)}{\sum_{u'} N(u',h)}$$

We then compute the Shannon entropy (log base 2) for each distribution. As table 4.5 shows, the average cross-character entropy is indeed over three times higher when English is used as the known language, clearly distinguishing it from Hebrew, an actual related language.

Decipherment Count

Another method for distinguishing between related and unrelated languages uses the predicted alphabetic mapping $\{\lambda_{(u,h)}\}$. At prediction time we can force our model to *only* use character substitutions with $\lambda_{(u,h)} = 1$ (as in the "Only Mapped" experiment in section 4.9.4). One consequence of this constraint will be that we simply *won't be able* to find candidate cognates for some number of lost-language words. However, if the languages are truly related and our alphabetic mapping is mostly correct, then we should still be able to find candidate cognates for a good portion of the lost language vocabulary.

We thus hypothesize that, under this constraint, we will find a much larger number of impossible-to-map words when an unrelated language is used. Our experiments bear out this hypothesis in the cases of English and Hebrew. When using Hebrew, we are still able to find candidate mappings for over 67% of all Ugaritic words (4,797 / 7,386). However, when using English as the known language, we can only propose cognates for 7% of Ugaritic words (551 / 7,386). As before, this difference would allow us to easily distinguish a related language (Hebrew) from an unrelated language (English) even before the decipherment predictions themselves have even been viewed by a human, let alone authenticated.

4.9.7 Error Analysis

In this section we analyze some of the errors made by our model in the task of cognate decipherment. We separately examine prefix errors, stem errors, and suffix errors. Table 4.6 gives the top errors in each category (see appendix F for a mapping from original graphemes to the characters in our transcription). We divide the errors into three major categories.

Segmentation Errors: The most obvious category of errors consists of segmentation mistakes. All but one of the top prefix errors falls into this category. For example, the most common prefix error is predicting a prefix *m*- when in reality no prefix occurs at all (for either the Ugaritic word or its Hebrew counterpart). Sometimes a prefix occurs but is not predicted. For example, the fourth most common prefix error consists of predicting the empty prefix instead of the actual prefix *b*-. In addition, segmentation errors explain some decipherment errors of stems as well. For example, our model deciphers the Ugaritic word *aṯt* (woman) as *ašm* (guilt), rather than correctly segmenting the feminine suffix and predicting the Hebrew word *aš-h*.

Substitution Errors: Another category of errors consists of incorrect letter substitutions. For example, the Ugaritic letter š maps to both š and ś in Hebrew.

However, our model only predicts the mapping (š,ś). In many instances, replacing Ugaritic š with either š *or* ś results in valid Hebrew words (with entirely different meanings). Six of the top stem prediction errors involve ambiguities of this sort. For instance, the Ugaritic word *šbʕ*, meaning "seven" is incorrectly mapped to the Hebrew word *śbʕ*, meaning "satisfied" rather than the correct Hebrew word for seven, *šbʕ*. Even if our model were to correctly predict both character mappings ((š,š) and (š,ś)), it is not obvious if it would automatically pick out the correct Hebrew word.

Insertion Errors: Finally, another category of errors consists of missing Hebrew character *insertions*. As discussed in the previous section, our model does almost equally well if we disallow insertions altogether. The most common error, by far, is a suffix error involving a missing insertion. The Ugaritic masculine plural suffix, -*m*, corresponds to the Hebrew masculine plural suffix -*ym*. However, -*m* is *also* a suffix in Hebrew, indicating the third person masculine plural possessive. As before, it is not obvious how such errors can be corrected. In the next and final section of the chapter we discuss some possible directions for enriching the model to account for these shortcomings.

4.10 Conclusion and Future Work

In this chapter we proposed a method for the automatic decipherment of lost languages. The key strength of our model lies in its ability to incorporate a range of linguistic intuitions in a statistical framework.

First among these intuitions is that both character and lexical correspondences across related languages should be consistent. In addition, morphological analysis played a crucial role in our model, as the correspondences between highly frequent prefixes and suffixes can be particularly revealing (and easy to find). Finally, we developed a novel prior that encodes a crucial intuition: that the mapping between alphabets should be *structurally sparse*. Each character in the lost language should map to a very limited number of characters in the related language, and vice versa.

We applied our decipherment model to a corpus of Ugaritic, an ancient Semitic language discovered in 1928 and manually deciphered four years later, using knowledge of Hebrew, a related language. As input to our model, we use the corpus of Ugaritic texts along with a Hebrew lexicon extracted from the Hebrew Bible.

Our main experiments show that by modeling the interplay between morphology, character correspondences, and lexical correspondences, our model was able to predict a largely correct decipherment of Ugaritic. 28 of 30 letters were correctly mapped to their Hebrew counterparts, and over 63% of words with Hebrew cognates were correctly deciphered. Further experiments indicated that several factors were crucial to this success. In the absence of morphological modeling and the prior constraint on character fertility, prediction accuracy degrades significantly.

Finally, we examined the issue of related language identification. For many currently undeciphered lost languages, the key challenge lies in finding a related living language (if one exists). While our model is not designed to *find* related languages, our experiments show that it can at least distinguish between related and unrelated pairs.

We hope to address several issues in future work. One deficiency of our model is that it fails to take into account the known *frequency* of Hebrew words and morphemes. The existence of a word or morpheme in the Hebrew lexicon is simply treated as a hard constraint. If the word exists, it may be matched to an Ugaritic counterpart, and otherwise it may not. What we see, in fact, is that the most common error of our model can be attributed to this feature: Our model incorrectly deciphers the Ugaritic masculine plural suffix (-*m*) as the Hebrew third person plural possessive suffix (-*m*), rather than the correct and much more common plural suffix (-*ym*). One way to achieve this frequency matching would be to simultaneously model the vocabularies of Ugaritic *and* Hebrew. Our current model treats the Hebrew lexicon as a wholly observed conditioning variable. Instead, we could assume that, just as Ugaritic words have latent Hebrew counterparts, so too do Hebrew words have latent Ugaritic counterparts. In this way, frequently occurring

Hebrew morphemes will have to be accounted for by frequently occurring Ugaritic morphemes.

Another direction for future work is to add *contextual cues* to our model. Currently, our model operates purely at the vocabulary level. As we saw in the previous section, many of the errors our model makes are due to ambiguity. A single Ugaritic word could be legitimately translated into several Hebrew counterparts based solely on the historical character mappings. Scholars, of course, use the *literary context* of words to help uncover their meanings. While the Hebrew words for "seven" and "satisfied" both fit the characters of the Ugaritic word *šb*ʕ, it is unlikely that both words would fit the context in which this word appears.

Prefix errors				*Stem errors*				*Suffix errors*			
Ug.	Predict	Hb.		Ug.	Predict	Hb.		Ug.	Predict	Hb.	
m-	m-		68	šlm	ślm	šlm	12	-m	-m	-ym	190
y-	y-		26	mit	mat	ma-h	8	-n	-n	-nw	43
a-	a-		19	ʕn	ʕwn	ʕyn	8			-h	29
		b-	18	šbʕ	śbʕ	šbʕ	8	-t	-t		28
b-	b-		14	aṯt	ašm	aš-h	7	-m	-m		28
		l-	13	ʕd	ʕd	bʕd	7	-y	-y		21
w-	w-	wy-	11	ib	ab	awyb	7	-n	-n		20
		k-	10	špt	śpt	śp-h	6	-n	-nw		16
		y-	8	dd	dd	dwd	6	-h	-h	-th	11
		t-	8	aḫt	aḥt	aḥ-wt	6	-k	-k	-tk	10
t-	m-	t-	7	šrš	śry	šrš	5	-k	-k		10
n-	n-		7	śś	nwn	śwś	5			-ym	9
wy-	wy-	w-	7	šm	śm	šm	5	-k	-k	-yk	7
h-	h-		6	qn	qn	qn-h	5			-t	7
k-	k-		6	šir	śar	šar	5	-w	-w		6
i-	a-		5	ḥrš	ḥrś	ḥrš	4	-m	-m	-wt	6

Table 4.6: Top cognate decipherment errors for prefixes, stems, and suffixes. For each morpheme category, the first column gives the Ugaritic morpheme (as segmented by our model), the second column gives the predicted Hebrew morpheme, the third column gives the correct Hebrew morpheme, and the fourth column gives the number of times this particular error was made.

Chapter 5

Conclusions and Future Work

In this thesis, we introduced the framework of multilingual learning. The core idea underlying this framework is that the systematic variations that we observe across languages correspond to variations in ambiguity. In other words, what one language leaves implicit, and thus ambiguous for computers (or even humans), another will express directly through overt linguistic forms. In the framework of multilingual learning, we treat these variations in ambiguity as a form of *naturally occurring supervision*. By jointly modeling multiple languages, the idiosyncratic ambiguities of each can be resolved through information explicit in the others.

We have applied this idea to several fundamental tasks of linguistic analysis, including part-of-speech tagging [111, 112, 85] (chapter 2), grammar induction [112] (chapter 3), and morphological analysis [108, 107] (not detailed in this thesis). In all three cases, we assumed access to multilingual parallel text corpora at training time, without any human annotations. We treated these corpora as a *computational Rosetta stone*, in which each language helps expose the latent structure of the others. We tested our approach by extracting language-specific models and then applying them to purely monolingual test data. In all cases, the models originally trained on multilingual data provided performance superior to monolingually trained counterparts, sometimes by very large margins. These results validate our core hypothesis.

One of the key challenges we faced throughout these tasks is that even for parallel sentences, the latent structure used by each language can vary significantly. Thus, one of our goals throughout this thesis was to discover shared cross-lingual structure while still allowing significant language-specific idiosyncrasies. To achieve this balancing act, we posited hierarchical Bayesian models which explain parallel sentences through a combination of multilingual and language-specific latent random variables.

Even so, the *scope* of the shared explanatory mechanism is often unknown: some sets of languages exhibit a much larger degree of shared structure than other. For example, parallel phrases in related language pairs like Hebrew and Arabic tend to mirror each other in morphological structure much more than unrelated language pairs (such as English and Hebrew). To account for this variability in shared structure, we employed non-parametric statistical methods which allow for a flexible number of shared variables, as dictated by the languages and data at hand.

5.1 Discussion

The key conclusion we have reached throughout this thesis is that multilingual modeling can yield significant gains in accuracy even without the presence of human annotation. In the introduction to this thesis (chapter 1), we posed a series of additional questions. We discuss each questions here in light of the results presented throughout the thesis.

Question 1: *Will multilingual learning provide more or less benefit when the languages in question are from the same family (e.g. Hebrew and Arabic, Italian and French, German and Dutch)? One might argue either way. One the one hand, related languages are likely to have a greater degree of shared latent structure. On the other hand, if their patterns of ambiguity are almost identical then little benefit would be gained.*

In chapter 2 we concluded that language relatedness by itself was neither positively nor negatively correlated with the success of multilingual learning. This result is not altogether surprising. It is well known in linguistics that even when languages descend from a common ancestor, they can quite quickly diverge in their basic structure when exposed to different neighboring languages. In fact, we found that common structural properties, regardless of language origin, correlated positively with multilingual success. For example, languages with higher word alignment density, and lower cross-lingual entropy tended to help one another.

Our work on unsupervised morphology induction [108, 107], though not discussed extensively in this thesis, provides another clue. There we initially found that Hebrew prediction accuracy was boosted by English, an unrelated but morphologically simple language, more so than by the related languages of Arabic and Aramaic. However, after encoding the *phonetic* relationship between these Semitic languages as a prior in our model, we found that the related languages indeed provided superior benefit. Indeed, for certain tasks, such as lost language decipherment (discussed below), the *only* benefit will come through the assumption of language-relatedness.

We can synthesize these findings in the following way: If historical language-relatedness is *not* explicitly modeled (as in chapter 2), then more abstract structural properties of the language pairing will prove decisive. However, the greatest benefits of multilingual learning may only be seen when we explicitly model language-relatedness as a background factor in our models.

Question 2: *Can multilingual learning be made to scale-up beyond pairs of languages? It seems that the arguments in favor of multilingual learning would only be strengthened as additional languages are modeled. Each language may provide some unique disambiguation cues lacking in the others. As a practical matter, massively multilingual data-sets do exist (e.g. the Bible, which has been translated into* *over 1,000 languages) and an ideal multilingual learning technique would thus scale gracefully in the number of languages.*

Our results in chapter 2 provide some initial answers to this question. There we formulated a part-of-speech tagging model that learns jointly from multilingual parallel text in any number of languages. We tested our model on up to eight languages and found that performance consistently improves as more languages are added (even when going from seven to eight). When we assumed a full tagging dictionary, jointly modeling eight languages cut the performance gap between supervised and unsupervised learning by *two-thirds.* It seems likely that performance would continue to improve with larger multilingual corpora.

Question 3: *Can multilingual learning account for complex latent structure where cross-lingual shared elements are minimal and difficult to discern? To do so effectively and efficiently will require an unobtrusive* representation *of whatever shared structure exists.*

Chapter 3 dealt with the difficult problem of unsupervised grammar induction. The greatest challenge in applying a multilingual framework to this task was in developing the right *representation.* For a simpler task like part-of-speech tagging, the sentence themselves (and their word alignments) determine the *structure* of the latent variables. The main learning task there is simply *labeling* those variables. In contrast, for grammar induction our main objective is one of structure induction itself. However, languages can use very different syntactic structures to express the same meaning. To account for this variability, we developed a probabilistic version of the tree alignment formalism [60]. This allowed us to represent the commonalities, or at least the systematic regularities, between the languages' trees. Experimentally, we showed that using this formalism on bilingual corpora yielded significant performance benefits over a state-of-the-art baseline. Grammar induction remains a difficult task and more work remains to be done. Nevertheless, these initial findings show that multilingual learning can indeed account for complex latent structure when the right formalism is deployed.

Question 4: *Can multilingual learning be effective without parallel data? Throughout this section our arguments have depended on the existence of parallel data as a computational Rosetta stone. However, if the languages in question come from the same family, it may be possible to use language-wide structural correspondences rather than the correspondences delivered by parallel text.*

To answer this question, chapter 4 turned to an inherently multilingual task: lost language decipherment. When a lost script or language is discovered we very rarely have the luxury of parallel data. Our only hope of recovering the language comes from cross-lingual structural analysis that links the lost writing system to a known language. Such analysis can take humans decades to perform. Our results on the Ugaritic language show that it is indeed possible to effectively capture shared language structure in the absence of parallel texts.

The key to this result lies in designing a model with the appropriate inductive biases. In particular, we know that the correct mapping between related languages will conform to certain rules and intuitions. For example, the mapping between alphabets should be *structurally sparse* (no letter should map to an inordinate number of others), and regular morphemic and lexical patterns should obtain. We designed a model that enforces these regularities. We believe that this kind of modeling can be transferred to more traditional NLP tasks to allow multilingual benefits even when parallel data is absent or scarce.

5.2 Future Work: Multilingual Semantics

To conclude, we briefly turn to a major direction for future multilingual work. Throughout this thesis, we have discussed the various layers of latent structure that undergird natural language sentences. These range from the morphemes involved in word production to the syntax trees which determines word order. However, we have never discussed the *meaning* of sentences. In fact, the ability to extract meaning from text is one of the paramount goals of natural language processing. Bringing this goal to fruition has been difficult for several reasons. Perhaps most fundamentally, it is not known conclusively whether consistent meaning representations underly language production at all. Even assuming they do, it is unlikely that a study of language alone will yield their structure without further gains from cognitive psychology and neuroscience.

One step around this dilemma is to posit a difference between “deep” and “shallow” semantics. The latter, rather than claiming to represent cognitively significant mental structures, instead seeks a representation of predicate-argument structure which hews loosely to the form of the sentence. For example, consider the following two sentences:

(1) I love fish.

(2) Fish are loved by me.

Although these sentences differ in emphasis (and would be used in very different discourse contexts), it is reasonable to assume that they convey the same basic information. In fact, a shallow semantic analysis of these sentences would likely yield a single predicate-argument structure, which we might simply represent as: $loves(I, fish)$. The key benefit of shallow semantic analysis is *precisely* that it allows us to capture the factual equivalence of sentences (1) and (2), despite their surface dissimilarity.

As mentioned throughout this thesis, languages differ in their latent structure, even when expressing the same meaning. This is likely to hold true for shallow

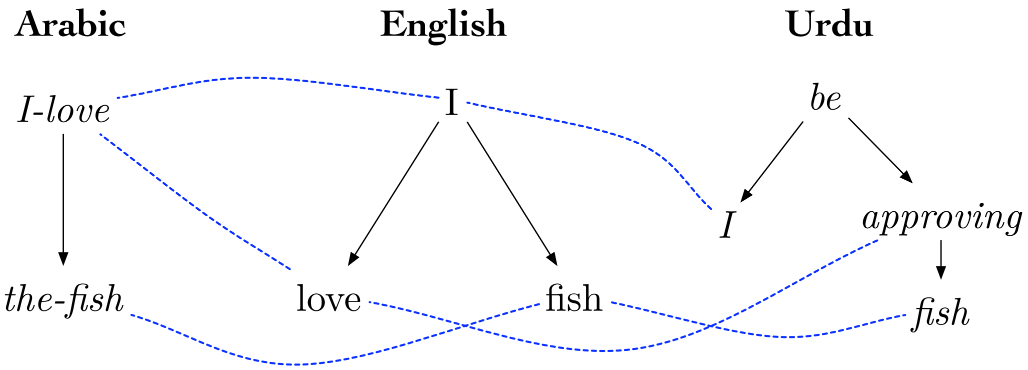


Figure 5-1: Word-aligned dependency parses for a sentence in English, Arabic, and Urdu.

semantics (and perhaps even for "deep" semantics). Nonetheless, the same sort of multilingual triangulation that we've applied to other areas of linguistic structure should succeed here as well. If the patterns of semantic ambiguity vary by language, then joint multilingual modeling should help pinpoint the correct analyses.

To get a sense of what this might look like, we can consider a multilingual example:

English:	I love fish.
Arabic:	*I-love the-fish.*
Urdu:	*I fish approving be.*

To see the underlying shallow semantics of these sentences, we can display them as word-aligned dependency trees, as in figure 5-1. What this analysis would hopefully reveal is the set of cross-lingual semantic correspondences:

$$\textit{love(I,fish)} = \textit{I-love(the-fish)} = \textit{be(I,approve(fish))}$$

Perhaps this example only serves to illustrate how language-specific this notion of shallow semantics can be. Be that as it may, it is certain that we cannot progress to any deeper level of understanding without considering the wide variety of languages and all the manners in which they express thought.

Appendix A

Tag Repository

	BG	CS	EN	ET	HU	RO	SL	SR
Adjective	x	x	x	x	x	x	x	x
Conjunction	x	x	x	x	x	x	x	x
Determiner	-	-	x	-	-	x	-	-
Interjection	x	x	x	x	x	x	x	x
Numeral	x	x	x	x	x	x	x	x
Noun	x	x	x	x	x	x	x	x
Pronoun	x	x	x	x	x	x	x	x
Particle	x	x	-	-	-	x	x	x
Adverb	x	x	x	x	x	x	x	x
Adposition	x	x	x	x	x	x	x	x
Article	-	-	-	-	x	x	-	-
Verb	x	x	x	x	x	x	x	x
Residual	x	x	x	x	x	x	x	x
Abbreviation	x	x	x	x	x	x	x	x

Table A.1: Tag repository for each language

Appendix B

Alignment Statistics

	BG	CS	EN	ET	HU	RO	SL	SR
BG		42163	51098	33849	31673	42017	45969	46434
CS	42163		43067	40207	31537	32559	57789	49740
EN	51098	43067		40746	39012	50289	52869	48394
ET	33849	40207	40746		32056	27709	42499	37681
HU	31673	31537	39012	32056		26455	34072	29797
RO	42017	32559	50289	27709	26455		36442	38004
SL	45969	57789	52869	42499	34072	36442		59865
SR	46434	49740	48394	37681	29797	38004	59865	

Table B.1: Number of alignments per language pair

	BG	CS	EN	ET	HU	RO	SL	SR	Avg.
BG		2.77	6.13	3.36	4.04	4.52	2.95	3.48	3.89
CS	2.77		3.67	1.92	2.73	3.61	2.59	2.64	2.85
EN	6.13	3.67		4.35	6.12	5.59	3.54	3.86	4.75
ET	3.36	1.92	4.35		2.88	3.88	2.44	2.21	3.01
HU	4.04	2.73	6.12	2.88		4.13	3.09	3.06	3.72
RO	4.52	3.61	5.59	3.88	4.13		3.78	3.92	4.20
SL	2.95	2.59	3.54	2.44	3.09	3.78		4.11	3.22
SR	3.48	2.64	3.86	2.21	3.06	3.92	4.11		3.33

Table B.2: Percentage of alignments removed per language pair

	All	BG	CS	EN	ET	HU	RO	SL	SR
Adjective	80.52	84.39	85.14	86.09	77.55	67.04	70.72	88.56	87.05
Conjunction	84.51	84.93	84.44	95.09	88.61	73.41	78.49	88.18	83.82
Determiner	54.32	-	-	56.82	-	-	41.07	-	-
Interjection	87.01	87.85	100.00	93.94	90.00	91.01	83.17	85.11	68.57
Numeral	82.56	79.31	86.78	93.66	74.51	72.97	85.94	91.50	80.27
Noun	85.39	88.01	88.63	91.31	80.43	77.90	78.31	91.84	87.52
Pronoun	61.86	69.53	61.61	73.73	57.75	39.55	52.29	68.93	65.13
Particle	69.71	66.71	84.39	-	-	-	68.79	71.92	73.03
Adverb	68.09	77.77	74.35	82.19	60.18	53.45	57.42	78.96	75.57
Adposition	62.48	66.58	65.17	65.54	35.10	33.88	46.62	74.77	72.58
Article	48.56	-	-	-	-	50.81	43.68	-	-
Verb	72.72	78.93	79.43	71.98	68.14	62.87	63.49	75.22	78.51
Residual	84.16	95.00	86.32	84.62	37.50	88.00	60.81	100.00	77.46
Abbreviation	87.46	66.67	90.00	-	90.74	88.61	69.23	90.00	91.18

Table B.3: For each part-of-speech, percentage of occurrences with an edge from a superlingual tag (in the latent variable model). A dash ("-") indicates that the part-of-speech does not occur in the given language.

Appendix C

Stanford Tagger Performance

Language	Accuracy
BG	96.1
CS	97.2
EN	97.6
ET	97.1
HU	96.3
RO	97.6
SL	96.6
SR	95.5
Avg.	96.7

Table C.1: Performance of the (supervised) Stanford tagger for the full lexicon scenario

Appendix D

Rank Correlation

Performance correlates for MergedNode *model*

Language	Cross-lingual entropy	Alignment density	LatentVariable performance
BG	-0.29	0.09	-0.09
CS	0.39	0.34	0.24
EN	0.28	0.77	0.42
ET	0.46	0.56	0.56
HU	0.31	-0.02	0.29
RO	0.34	0.83	0.89
SL	0.59	0.66	0.95
SR	0.21	0.13	0.63
Avg.	0.29	0.42	0.49

Performance correlates for LatentVariable *model*

Language	Cross-lingual entropy	Alignment density	MergedNode performance
BG	0.58	0.44	-0.09
CS	-0.40	-0.44	0.24
EN	0.67	0.41	0.42
ET	0.14	0.32	0.56
HU	-0.14	-0.72	0.29
RO	0.04	0.68	0.89
SL	0.57	0.54	0.95
SR	0.18	0.10	0.68
Avg.	0.21	0.17	0.49

Table D.1: Pearson correlation coefficients between bilingual performance on the target language and various rankings of the supplementary language. For both models and for each target language, we obtain a ranking over all supplementary languages based on bilingual performance in the target language. These rankings are then correlated with other characteristics of the bilingual pairing: **cross-lingual entropy** (the entropy of tag distributions in the target language given aligned tags in the supplementary language); **alignment density** (the percentage of words in the target language aligned to words in the auxiliary language); and performance in the alternative model (target language performance when paired with the same supplementary language in the alternative model).

Appendix E

Universal Helpfulness

MergedNode model		LatentVariable model	
ET	2.43	BG	1.86
EN	2.57	SR	3.00
SL	3.14	ET	3.14
BG	3.43	CS	3.71
SR	3.43	EN	3.71
RO	4.71	SL	3.71
CS	5.00	RO	4.14
HU	5.71	HU	6.00

Table E.1: Average helpfulness rank for each language under the two models

Appendix F

Ugaritic-Hebrew letter mappings

א	a
ב	b
ג	g
ד	d
ה	h
ו	w
ז	z
ח	ḥ
ט	ṭ
י	y
כ	k
ל	l
מ	m
נ	n
ס	s
ע	ʕ
פ	p
צ	ṣ
ק	q
ר	r
שׁ	š
שׂ	ś
ת	t

𐎀	a
𐎁	b
𐎂	g
𐎃	ḫ
𐎄	d
𐎅	h
𐎆	w
𐎇	z
𐎈	ḥ
𐎉	ṭ
𐎊	y
𐎋	k
𐎌	š
𐎍	l
𐎎	m
𐎏	ḏ
𐎐	n
𐎑	ẓ
𐎒	s
𐎓	ʕ
𐎔	p
𐎕	ṣ
𐎖	q
𐎗	r
𐎘	ṯ
𐎙	ġ
𐎝	ś
𐎚	t
𐎛	i
𐎜	u

Table F.1: Hebrew letters (left) and Ugaritic letters (right) with phonetic transcription.

		Hebrew																						
		a	b	g	d	h	w	z	ḥ	ṭ	y	k	l	m	n	s	ʕ	p	ṣ	q	r	š	ś	t
Ugaritic	a	X																						
	b		X																					
	g			X																				
	ḫ								X															
	d				X			X																
	h					X																		
	w						X																	
	z							X																
	ḥ								X															
	ṭ									X														
	y										X													
	k											X												
	š																					X	X	
	l												X											
	m													X										
	ḏ							X																
	n														X									
	ẓ																		X					
	s															X								
	ʕ																X							
	p																	X						
	ṣ																		X					
	q																			X				
	r																				X			
	ṯ																					X		
	ġ																X		X					
	ś															X								
	t																							X
	i	X																						
	u	X																						

Table F.2: **Gold Standard:** Mappings between Ugaritic and Hebrew letters, reflecting the historical relationship between the corresponding phonemes.

	Hebrew																						
Ugaritic	a	b	g	d	h	w	z	ḥ	ṭ	y	k	l	m	n	s	ʕ	p	ṣ	q	r	š	ś	t
a	X																						
b		X																					
g			X																				
ḫ								X															
d				X			*																
h					X																		
w						X																	
z		X*					*																
ḥ								X															
ṭ									X														
y										X													
k											X												
š																					*	X	
l												X											
m													X										
ḏ							X																
n														X									
ẓ																		X					
s															X								
ʕ																X							
p																	X						
ṣ																		X					
q																			X				
r																				X			
ṯ																					X		
ġ																X		*					
ś														X*	*								
t																							X
i	X																						
u	X																						

Table F.3: **Model Predictions:** Mappings between Ugaritic and Hebrew letters, as predicted by the matrix of indicator variables $\{\lambda_{(u,h)}\}$ in our model. Entries where predictions differ from the gold-standard mapping are indicated with an asterisk (*).

Hebrew

Ugaritic	a	b	g	d	h	w	z	ḥ	ṭ	y	k	l	m	n	s	ʕ	p	ṣ	q	r	š	ś	t
a	*																						
b		X																					
g			X																				
ḫ								*															
d				X			*																
h					*																		
w						*																	
z							X																
ḥ								X															
ṭ									X														
y						X*				*													
k											X												
š																					*	X	
l												X											X*
m										X*			X										
ḏ							*																
n														X									
ẓ																		*					
s															X								
ʕ	X*															X							
p																	X						
ṣ																		X					
q																			X				
r																				X	X*		
ṯ																					*		
ġ																*		*					
ś															*								
t					X*																		*
i	*																						
u	*																						

Table F.4: **Baseline Predictions 1:** Mappings between Ugaritic and Hebrew letters, as predicted by the HMM baseline (where (u, h) is predicted iff $u = \mathrm{argmax}_{u'} P(u'|h)$). Entries where predictions differ from the gold-standard mapping are indicated with an asterisk (*).

	Hebrew																						
Ugaritic	a	b	g	d	h	w	z	ḥ	ṭ	y	k	l	m	n	s	ʕ	p	ṣ	q	r	š	ś	t
a	X																						
b		X																					
g			X																				
ḫ								X															
d				X			*																
h					*	X*																	
w						*				X*													
z							X																
ḥ								X															
ṭ									X														
y						X*				*													
k											X												
š																					X	*	
l												X											
m													X										
ḏ							X																
n														X									
ẓ			X*															*					
s															X								
ʕ																X							
p																	X						
ṣ																		X					
q																			X				
r																				X			
ṯ																					X		
ġ																*		*	X*				
ś															*					X*			
t					X*																		*
i	X																						
u	X																						

Table F.5: **Baseline Predictions 2:** Mappings between Ugaritic and Hebrew letters, as predicted by the HMM baseline (where (u, h) is predicted iff $h = \text{argmax}_{h'} P(h'|u) \propto P(u|h')P(h')$). Entries where predictions differ from the gold-standard mapping are indicated with an asterisk (*).

Hebrew

Ugaritic	a	b	g	d	h	w	z	ḥ	ṭ	y	k	l	m	n	s	ʕ	p	ṣ	q	r	š	ś	t
a	X																						
b		X																					
g			X																				
ḫ								X															
d				X			*																
h					*	X*																	
w						*				X*													
z							X																
ḥ								X															
ṭ									X														
y						X*				*													
k											X												
š																					X	X	
l												X											X*
m										X*			X										
ḏ							X																
n														X									
ẓ			X*															*					
s															X								
ʕ	X*															X							
p																	X						
ṣ																		X					
q																			X				
r																				X	X*		
ṯ																					X		
ġ																*		*	X*				
ś															*					X*			
t					X*																		*
i	X																						
u	X																						

Table F.6: **Baseline Predictions 3:** Mappings between Ugaritic and Hebrew letters, as predicted by the HMM baseline (where (u, h) is predicted iff **either** $u = \text{argmax}_{u'} P(u'|h)$ **or** $h = \text{argmax}_{h'} P(h'|u) \propto P(u|h')P(h')$). Entries where predictions differ from the gold-standard mapping are indicated with an asterisk (*).

Bibliography

[1] Steven Abney and Steven Bird. The human language project: Building a universal corpus of the world's languages. In *Proceedings of the ACL*, pages 88–97, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[2] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174, November 1974.

[3] James Baker. Trainable grammars for speech recognition. In *Proceedings of the Acoustical Society of America*, 1979.

[4] Michele Banko and Robert C. Moore. Part-of-speech tagging in context. In *Proceedings of the COLING*, pages 556–561, 2004.

[5] Emily M. Bender. Linguistically naïve != language independent: why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics*, pages 26–32, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[6] Taylor Berg-Kirkpatrick and Dan Klein. Phylogenetic grammar induction. In *Proceedings of the ACL*, pages 1288–1297, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[7] Shane Bergsma and Greg Kondrak. Multilingual cognate identification using integer linear programming. In *RANLP Workshop on Acquisition and Management of Multilingual Lexicons*, Borovets, Bulgaria, September 2007.

[8] Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. Phrase-based statistical machine translation with pivot languages. In *International Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation (IWSLT)*, pages 143–149, 2008.

[9] Indrajit Bhattacharya, Lise Getoor, and Yoshua Bengio. Unsupervised sense disambiguation using bilingual probabilistic models. In *Proceedings of ACL*, page 287, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[10] Ann Bies, Martha Palmer, Justin Mott, and Colin Warner. *English Chinese translation treebank v 1.0.* LDC2007T02, 2007.

[11] Phil Blunsom, Trevor Cohn, and Miles Osborne. Bayesian synchronous grammar induction. In *Proceedings of NIPS*, 2008.

[12] Alexandre Bouchard, Percy Liang, Thomas Griffiths, and Dan Klein. A probabilistic approach to diachronic phonology. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 887–896, 2007.

[13] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[14] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. Word-sense disambiguation using statistical methods. In *Proceedings of the ACL*, pages 264–270, 1991.

[15] 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)*, pages 149–164, New York City, June 2006. Association for Computational Linguistics.

[16] David Burkett, John Blitzer, and Dan Klein. Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of NAACL*, 2010.

[17] David Burkett and Dan Klein. Two languages are better than one (for syntactic parsing). In *Proceedings of EMNLP*, pages 877–886, 2008.

[18] David Burkett, Slav Petrov, John Blitzer, and Dan Klein. Learning better monolingual models with unannotated bilingual text. In *Proceedings of CoNLL*, 2010.

[19] Lyle Campbell. *Historical Linguistics: An Introduction.* Cambridge: MIT Press, 2004.

[20] Eugene Charniak and Glen Carroll. Two experiments on learning probabilistic dependency grammars from corpora. In *Proceedings of the AAAI Workshop on Statistically-Based NLP Techniques*, pages 1–13, 1992.

[21] Yu Chen, Andreas Eisele, and Martin Kay. Improving statistical machine translation efficiency by triangulation. In *Proceedings of LREC*, 2008.

[22] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the ACL*, pages 263–270, 2005.

[23] Shay B. Cohen and Noah A. Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of the NAACL/HLT*, 2009.

[24] Trevor Cohn and Mirella Lapata. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of ACL*, 2007.

[25] M. Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.

[26] Bernard Comrie. *Language universals and linguistic typology: Syntax and morphology*. Oxford: Blackwell, 1989.

[27] CTIA-The Wireless Association. CTIA's wireless industry indices: 1985 - 2009.

[28] Jesus-Luis Cunchillos, Juan-Pablo Vita, and Jose-Ángel Zamora. Ugaritic data bank. CD-ROM, 2002.

[29] Ido Dagan, Alon Itai, and Ulrike Schwall. Two languages are more informative than one. In *Proceedings of the ACL*, pages 130–137, 1991.

[30] Gregoria del Olo Lete and Joaquín Sanmartín. *A Dictionary of the Ugaritic Language in the Alphabetic Tradition*. Number 67 in Handbook of Oriental Studies. Section 1 The Near and Middle East. Brill, 2004.

[31] Mona Diab and Philip Resnik. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the ACL*, pages 255–262, 2002.

[32] Jason Eisner. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of the ACL*, pages 205–208, 2003.

[33] T. Erjavec. MULTEXT-East version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Fourth International Conference on Language Resources and Evaluation, LREC*, volume 4, pages 1535–1538, 2004.

[34] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(230):577–588, 1995.

[35] Anna Feldman, Jirka Hana, and Chris Brew. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In *Proceedings of LREC*, pages 549–554, 2006.

[36] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, 1:209–230, 1973.

[37] Pascale Fung and Kathleen McKeown. Finding terminology translations from non-parallel corpora. In *Proceedings of the Annual Workshop on Very Large Corpora*, pages 192–202, 1997.

[38] Kuzman Ganchev, Joao Graca, and Ben Taskar. Better alignments = better translations? In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, 2008.

[39] Andrew Gelman, John B. Carlin, Hal .S. Stern, and Donald .B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2004.

[40] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[41] Dmitriy Genzel. Inducing a multilingual dictionary from a parallel multitext in related languages. In *Proceedings of HLT/EMNLP*, pages 875–882, 2005.

[42] W.R. Gilks, S. Richardson, and DJ Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

[43] Sharon Goldwater and Thomas L. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the ACL*, pages 744–751, 2007.

[44] Joshua Goodman. *Parsing inside-out*. PhD thesis, Harvard University, 1998.

[45] Alan Groves and Kirk Lowery, editors. *The Westminster Hebrew Bible Morphology Database.* Westminster Hebrew Institute, Philadelphia, PA, USA, 2006.

[46] Jacques B. M. Guy. An algorithm for identifying cognates in bilingual wordlists and its applicability to machine translation. *Journal of Quantitative Linguistics*, 1(1):35–42, 1994.

[47] Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Proceedings of HLT-NAACL*, pages 320–327, 2006.

[48] Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the ACL/HLT*, pages 771–779, 2008.

[49] Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[50] Kenneth Hale, Michael Krauss, Lucille J. Watahomigie, Akira Y. Yamamoto, Colette Craig, L. M. Jeanne, and N. C. England. Endangered languages. *Language*, 68(1):1–42, 1992.

[51] David Hall and Dan Klein. Finding cognates using phylogenies. In *Proceedings of ACL*, 2010.

[52] C. Han, N.R. Han, E.S. Ko, H. Yi, and M. Palmer. Penn Korean Treebank: Development and evaluation. In *Proc. Pacific Asian Conf. Language and Comp*, 2002.

[53] Jiri Hana, Anna Feldman, and Chris Brew. A resource-light approach to russian morphology: Tagging russian using czech resources. In *Proceedings of EMNLP*, pages 222–229, 2004.

[54] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[55] Robert Hetzron, editor. *The Semitic Languages.* Routledge, 1997.

[56] Geoffrey E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

[57] John E. Hopcroft. An n log n algorithm for minimizing states in a finite automaton. Technical report, Stanford, CA, USA, 1971.

[58] R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Journal of Natural Language Engineering*, 11(3):311–325, 2005.

[59] H. Ishwaran and J.S. Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[60] T. Jiang, L. Wang, and K. Zhang. Alignment of trees – an alternative to tree edit. *Theoretical Computer Science*, 143(1):137–148, 1995.

[61] Mark Johnson. Why doesn't EM find good HMM POS-taggers? In *Proceedings of EMNLP/CoNLL*, pages 296–305, 2007.

[62] Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York, April 2007. Association for Computational Linguistics.

[63] D. Klein. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University, 2005.

[64] Dan Klein and Christopher D. Manning. A generative constituent-context model for improved grammar induction. In *Proceedings of the ACL*, pages 128–135, 2002.

[65] K. Knight and K. Yamada. A computational approach to deciphering unknown scripts. In *ACL Workshop on Unsupervised Learning in Natural Language Processing*, 1999.

[66] Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL*, pages 499–506, 2006.

[67] Kevin Knight and Richard Sproat. Writing systems, transliteration and decipherment. NAACL Tutorial, 2009.

[68] Philipp Koehn and Kevin Knight. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 9–16, 2002.

[69] Grzegorz Kondrak. Identifying cognates by phonetic and semantic similarity. In *Proceeding of NAACL*, pages 1–8, 2001.

[70] Jonas Kuhn. Experiments in parallel-text based grammar induction. In *Proceedings of the ACL*, page 470, 2004.

[71] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[72] Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[73] Cong Li and Hang Li. Word translation disambiguation using bilingual bootstrapping. In *Proceedings of the ACL*, pages 343–351, 2002.

[74] Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[75] John B. Lowe and Martine Mazaudon. The reconstruction engine: a computer implementation of the comparative method. *Computational Linguistics*, 20(3):381–417, 1994.

[76] Gideon S. Mann and David Yarowsky. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[77] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1994.

[78] I. Dan Melamed. Multitext grammars and synchronous parsers. In *Proceedings of the NAACL/HLT*, pages 79–86, 2003.

[79] Bernard Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–171, 1994.

[80] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[81] Rada Mihalcea. *Current Issues in Linguistic Theory: Recent Advances in Natural Language Processing*, chapter Unsupervised Natural Language Disambiguation Using Non-Ambiguous Words. John Benjamins Publisher, 2004.

[82] Iain Murray, Zoubin Ghahramani, and David MacKay. MCMC for doubly-intractable distributions. In *UAI*, 2006.

[83] Jerome L. Myers and Arnold D. Well. *Research Design and Statistical Analysis.* Lawrence Erlbaum, 2nd edition, 2002.

[84] Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of EMNLP*, 2010.

[85] Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. Multilingual part-of-speech tagging: two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36(1):341–385, 2009.

[86] Hwee Tou Ng, Bin Wang, and Yee Seng Chan. Exploiting parallel texts for word sense disambiguation: an empirical study. In *Proceedings of the ACL*, pages 455–462, 2003.

[87] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[88] Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL*, pages 950–958, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[89] Franz Josef Och and Hermann Ney. Statistical multi-source translation. In *MT Summit 2001*, pages 253–258, 2001.

[90] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[91] Sebastian Padó and Mirella Lapata. Cross-linguistic projection of role-semantic information. In *Proceedings of the HLT*, pages 859–866, 2005.

[92] Sebastian Padó and Mirella Lapata. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of ACL*, pages 1161 – 1168, 2006.

[93] Gerald Penn and Travis Choma. Quantitative methods for classifying writing systems. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 117–120, New York City, USA, June 2006. Association for Computational Linguistics.

[94] Oxford University Press. *The Revised Standard Version Bible.* Oxford University Press, 2002.

[95] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[96] Reinhard Rapp. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the ACL*, pages 519–526, 1999.

[97] Philip Resnik and David Yarowsky. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 79–86, 1997.

[98] Eric Sven Ristad and Peter N. Yianilos. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532, 1998.

[99] C.P. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation.* Springer Verlag, 2007.

[100] Andrew Robinson. *Lost Languages: The Enigma of the World's Undeciphered Scripts.* Thames & Hudson, 2009.

[101] William M. Schniedewind and Joel H. Hunt. *A Primer on Ugaritic: Language, Culture and Literature.* Cambridge University Press, 2007.

[102] Yoav Seginer. Fast unsupervised incremental parsing. In *Proceedings of the ACL*, pages 384–391, 2007.

[103] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.

[104] David A. Smith and Noah A. Smith. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceeding of EMNLP*, pages 49–56, 2004.

[105] Mark S. Smith, editor. *Untold Stories: The Bible and Ugaritic Studies in the Twentieth Century.* Hendrickson Publishers, 1955.

[106] Noah A. Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the ACL*, pages 354–362, 2005.

[107] Benjamin Snyder and Regina Barzilay. Cross-lingual propagation for morphological analysis. In *Proceedings of the AAAI*, pages 848–854, 2008.

[108] Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of the ACL/HLT*, pages 737–745, 2008.

[109] Benjamin Snyder, Regina Barzilay, and Kevin Knight. A statistical model for lost language decipherment. In *Proceedings of the ACL*, pages 1048–1057, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[110] Benjamin Snyder, Tahira Naseem, and Regina Barzilay. Unsupervised multilingual grammar induction. In *Proceedings of the ACL*, pages 73–81, 2009.

[111] Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Unsupervised multilingual learning for POS tagging. In *Proceedings of EMNLP*, pages 1041–1050, 2008.

[112] Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Adding more languages improves unsupervised multilingual part-of-speech

tagging: a Bayesian non-parametric approach. In *Proceedings of the NAACL*, pages 83–91, 2009.

[113] Andreas Stolcke and Stephen M. Omohundro. Inducing probabilistic grammars by Bayesian model merging. In *Proceedings of ICGI*, pages 106–118, 1994.

[114] Kuo-Chung Tai. The tree-to-tree correction problem. *J. ACM*, 26(3):422–433, 1979.

[115] The Radicati Group, Inc. Email statistics report, 2009-2013.

[116] 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, *NIPS*, pages 1521–1528, Cambridge, MA, 2008. MIT Press.

[117] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP*, pages 63–70, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[118] Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of NAACL/HLT*, pages 484–491, 2006.

[119] Chong Wang and David Blei. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS*, pages 1982–1989. MIT Press, 2009.

[120] Wilfred Watson and Nicolas Wyatt, editors. *Handbook of Ugaritic Studies*. Brill, 1999.

[121] Dekai Wu. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *IJCAI*, pages 1328–1337, 1995.

[122] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

[123] Dekai Wu and Hongsing Wong. Machine translation with a stochastic grammatical channel. In *Proceedings of the ACL/COLING*, pages 1408–1415, 1998.

[124] Chenhai Xi and Rebecca Hwa. A backoff model for bootstrapping resources for non-english languages. In *Proceedings of HLT/EMNLP*, pages 851 – 858, 2005.

[125] JL Xu. Multilingual search on the world wide web. In *Proceedings of the Hawaii International Conference on System Sciences HICSS*, volume 33, 2000.

[126] David Yarowsky and Grace Ngai. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the NAACL*, pages 1–8, 2001.

[127] David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT*, pages 161–168, 2000.

[128] David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[129] Hao Zhang and Daniel Gildea. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the ACL*, pages 475–482, 2005.

[130] Kaizhong Zhang, Rick Statman, and Dennis Shasha. On the editing distance between unordered labeled trees. *Inf. Process. Lett.*, 42(3):133–139, 1992.