Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework

by

Richard Wicentowski

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2002

Abstract

Computational morphology is an important component of most natural language processing tasks including machine translation, information retrieval, word-sense disambiguation, parsing, and text generation. Morphological analysis, the process of finding a root form and part-of-speech of an inflected word form, and its inverse, morphological generation, can provide fine-grained part of speech information and help resolve necessary syntactic agreements. In addition, morphological analysis can reduce the problem of data sparseness through dimensionality reduction.

This thesis presents a successful original paradigm for both morphological analysis and generation by treating both tasks in a competitive linkage model based on a combination of diverse inflection-root similarity measures. Previous approaches to the machine learning of morphology have been essentially limited to string-based transduction models. In contrast, the work presented here integrates both several new noise-robust, trie-based supervised methods for learning these transductions, and also a suite of unsupervised alignment models based on weighted Levenshtein distance, position-weighted contextual similarity, and several models of distributional similarity including expected relative frequency. Via iterative bootstrapping the combination of these models yields a full lemmatization analysis competitive with fully supervised approaches but without any direct supervision. In addition, this thesis also presents an original translingual projection model for morphology induction, where previously learned morphological analyses in a second language can be robustly projected via bilingual corpora to yield successful analyses in the new target language without any monolingual supervision.

Collectively these methods outperform previously published algorithms for

the machine learning of morphology in several languages, and have been applied to a large representative subset of the world's language's families, demonstrating the effectiveness of this new paradigm for both supervised and unsupervised multilingual computational morphology.

Advisor: David Yarowsky

Readers: David Yarowsky
Jason Eisner

Dedicated to my family and friends

who helped make this possible

Acknowledgements

Though my name is the only author on this work, many people have contributed to its completion: those who provided insight and comments, those who provided ideas and suggestions, those who provided entertainment and distractions, and those who provided love and support.

My advisor, David Yarowsky, is naturally at the top of this list. He has always been an enthusiastic supporter of my work, providing a nearly unending supply of ideas. He gave me the independence to pursue my own interests, and in the end, gave me the guidance needed to clear the final hurdles.

Jason Eisner has supplied a tremendous amount of feedback in constructing this thesis. His clear thinking (and acute skill at rewriting probability models) provided practical and insightful comments at key moments.

Both David and Jason also deserve special awards for their willingness to give me comments at all hours of night (and early morning), at short notice, and without complaint. Their selflessness made the completion of this thesis much easier.

In addition to being a friend, coffee buddy, and research partner, Hans Florian is due a huge debt of gratitude for his assistance in nearly everything that I needed. Whether it was installing Linux on my laptop (twice), running to grab me an overhead projector during my thesis defense, or just being around to answer my never-ending stream of questions, Hans was always there, ready to help.

Many thanks are also due to the incredibly talented group of colleagues who I had the privilege of working with while I was at Hopkins: Grace Ngai, Charles Schafer, Gideon Mann, Silviu Cucerzan, Noah Smith, John Henderson, Jun Wu, and Paola Virga. Special mention goes to Charles, who helped collect many of the corpora

used in this thesis, and to Gideon, who, along with Charles, made every day at school a lot more liveable.

In what seems like a long time ago, Scott Weiss was there at the right time to help me get my teaching career going. The operating systems class we taught together in the Spring of 1997 was probably the most important thing that happened to me at Hopkins. For helping me get to where I am now, I will always be thankful.

I will be forever thankful to my closest friends: Andrew Beiderman, Paul Sack, and Matthew Sydes. While they were always there to help when the going was rough, I'm most thankful for the morning bagels (everything with cream cheese, not toasted), movies, long road trips, endless Scrabble marathons, frisbee golf, and Age of Empires.

I can never thank my family enough, especially Mom and Dad, for always providing encouragement, the occasional "Are you finished yet?", and a place to turn to when the going was rough. Dad was making sure the light was still on at the end of the tunnel, and Mom was making sure that I kept my head straight. Both provided nothing short of unconditional love and support.

And last, but absolutely not least, Naomi has been incredibly patient with me throughout this process. Her love and understanding have been a constant source of inspiration for me. ILY.

Richard Wicentowski

October 2002

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Morphology in Language

In every language in the world, whether it be written, spoken or signed, morphology is fundamentally involved in both the production of language, as well as its understanding. Morphology is what makes a *painter* someone who *paints*, what makes *inedible* something that is not *edible*, what makes *dogs* more than a single *dog*, and why *he jumps* but *they jump*.

But it's not always that easy. Morphology is also what makes a *cellist* someone who plays the *cello*, what makes *inedible* something that cannot be *eaten*, makes *geese* more than a single *goose*, and why *they are* but *he is*.

Morphology plays two central roles in language. In its first role, *derivational morphology* allows existing words to be used as the base for forming new words with different meanings and different functionality. From the above examples, the noun *cellist* is formed from the noun *cello*, and the adjective *inedible* has a different semantic meaning its related

verb from *eat*.

In its second role, *inflectional morphology* deals with syntactic features of the languages such as **person** (I *am*, you *are*, he *is*), **number** (one *child*, two *children*), **gender** (*actor*, *actress*), **tense** (*eat*, *eats*, *eating*, *eaten*, *ate*), **case** (*he*, *him*, *his*), and **degree** (*cold*, *colder*, *coldest*). These syntactic features, required to varying degrees by different languages, do not change the part of speech of the word (as the verb *eat* becomes the adjective *inedible*) and do not change the underlying meaning of the word (as *cellist* from *cello*).

Speakers, writers and signers of language form these syntactic agreements, called *inflections*, from base words, called *roots*. In doing so, these producers of language start with a root word (for example, the verb *go*) and, governed by a set of syntactic features (for example, *third person, singular, present*), form the appropriate inflection (*goes*). This process is called **morphological generation**.

In order for this process to be effective, the listeners, readers and observers of language must be able to take the inflected word (*actresses*) and find the underlying root (*actor*) as well as the set of conveyed syntactic features (*feminine, plural*). This decoding process is called **morphological analysis**.

While both morphological generation and morphological analysis will be addressed in this thesis, the primary focus of the work presented here will be morphological analysis. More specifically, the focus task will be *lemmatization*, a sub-task of morphological analysis concerned with finding the underlying root of an inflection (e.g. geese $\rightarrow$ goose) as a distinct problem from fully analyzing the syntactic features encoded in the inflection (e.g. third person singular).

Additionally, the work presented here will deal exclusively with orthography, the way in which words are written, not phonology, the way words are spoken. In many languages, this distinction is largely meaningless. In Italian, for example, words are spoken in a very systematic relation to how they are written, and vice versa. On the other hand, the association between the way words in English are spoken and the way they are written is often haphazard.

1.2 Computational Morphology

Mathematically, the process of lemmatization in inflectional morphology can be described as the binary relation over the set of roots in the language (W), the set of parts of speech in the language (π), and the set of inflections in the language (W') as shown in (1.1):

$$\begin{gathered}\textsc{Infl} : W \times \Pi \rightarrow W' \\ \text{where } \textsc{Infl}(w, \pi) = w' \text{ such that } w' \text{ is an inflection of } w \text{ with part-of-speech } \pi\end{gathered} \tag{1.1}$$

Lemmatization in computational morphology can be viewed as a machine learning task whose goal is to learn this relation. The Infl relation effectively defines a string transduction from w to w' for a given part-of-speech π. This string transduction defines the process by which the root is rewritten as the inflection. One way to define such a transduction is shown in Table 1.1.

The Infl relation in (1.1) describes the process of morphological generation. Since much of this work deals with morphological analysis, the relation that will be modeled is

	ROOT	PART OF SPEECH	STRING TRANSDUCTION	INFLECTION
English:	**take**	**VBG**	e → ing	**taking**
	take	**VBZ**	ϵ → s	**takes**
	take	**VBN**	ϵ → n	**taken**
	take	**VBD**	ake → ook	**took**
	skip	**VBD**	ϵ → ped	**skipped**
	defy	**VBG**	ϵ → ing	**defying**
	defy	**VBZ**	y → ies	**defies**
	defy	**VBD**	y → ied	**defied**
Spanish:	**jugar**	**VPI1P**	r → mos	**jugamos**
	jugar	**VPI3S**	gar → ega	**juega**
	jugar	**VPI3P**	gar → egan	**juegan**
	tener	**VPI3P**	ener → ienen	**tienen**

Table 1.1: The target output of morphological generation is an alignment between a (root, part of speech) pair, and the inflection of that root appropriate for the part of speech, while analysis is its inverse. The hypothesized string transductions shown above are just one of many possible ways that this string transduction process can be modeled. The labels VBD, VBG, VBZ, and VBN in English refer to the past tense, present participle, third person singular, and past participle, respectively. VPI3S, VPI3P, and VPI1P refer to the Spanish third person present indicative singular and plural, and first person present indicative plural, respectively.

the inverse of the INFL relation defined as in (1.2).

$$\text{INFL}^{-1} : W' \rightarrow W \times \pi$$

$$\text{where } \text{INFL}^{-1}(w') = (w, \pi) \text{ such that } w \text{ is the root of } w', \text{and } \pi \text{ is its part-of-speech} \tag{1.2}$$

Any string transduction which transforms w into w' in INFL should ideally be reversible to the appropriate string transduction transforming w' to w in INFL^{-1}. In this way, the transduction $e \rightarrow ing$ which transforms *take* into *taking*, can be reversed ($ing \rightarrow e$) to transform *taking* into *take*.

1.3 Morphological Phenomena

In linguistics, morphology is the study of the internal structure and transformational processes of words. In this way, it is analagous to biological morphology which studies the internal structures of animals. The internal structure of animals are its individual organs. The internal structure of words are its morphemes. Just as every animal is a structured combination of organs, every word in every language of the world is a structured combination of morphemes.

Each morpheme is an individual unit of meaning. Words are formed from a combination of one or more *free morphemes* and zero or more *bound morphemes*. Free morphemes are units of meaning which can stand on their own as words. Bound morphemes are also units of meaning; however, can not occur as words on their own: they can only occur in combination with free morphemes. From this definition, it follows that a word is either a single free morpheme, or a combination of a single free morpheme with other free and bound morphemes.

The English word *jumped*, for example, is comprised of two morphemes, *jump+ed*. Since *jump* is an individual unit of meaning which cannot be broken down further into smaller units of meaning, it is a morpheme. And, since *jump* can occur on its own as a word in the language, it is a free morpheme. The unit *+ed* can be added to a large number of English verbs to create the past tense. Since *+ed* has meaning, and since it can not be segmented into smaller units, it is a morpheme. However, *+ed* can only occur as a part of another word, not as a word on its own; therefore, it is a bound morpheme.

The process by which bound morphemes are added to free morphemes can often

be described using a *word formation rule*. For example, "Add *+ed* to the end of an English verb to form the past tense of that verb" is an orthographic word formation rule. This process, since it is true for a large number of English verbs, is said to be *regular*. When a rule can only can be used to explain only a small number of word forms in the language, the word formation rule is said to be *irregular*. For example, the rule that says "Change the final letter of the root from *o* into the string *id* to form the past tense" which is applicable only to the root-inflection pair *do-did*, is irregular. There are other processes that deviate from the regular pattern in partially or fully systematic ways for certain subsets of the vocabulary. Such processes include the doubling of consonants which occurs for some verbs (e.g. *thin* $\leftrightarrow$ *thinned*), but not others (e.g. *train* $\leftrightarrow$ *trained*). Such processes are often referred to as being *semi-regular*.

Each of the regular word formation rules can be classified as realizing one (or more) of a set of morphological phenomena which are found in the world's languages. These phenomena include:

1. **Simple affixation**: adding a single morpheme (an *affix*) to the beginning (*prefix*), end (*suffix*), middle (*infix*), or to both the beginning and the end (*circumfix*) of a root form. Affixation may involve phonological (or orthographical) changes at the location where the affix is added. These are called *point-of-affixation changes*.

2. **Vowel harmony**: affixation (usually suffixation) where the phonological content of the resulting inflection may be altered from affixation to obey systematic preferences for vowel agreement in the root and affix.

3. **Internal vowel shifts**: systematic changes in the vowel(s) between the inflection and

the root often, but not always, associated with the addition of an affix.

4. **Agglutination**: multiple affixes are "glued" together (concatentated) in constrained sequences to form inflections.

5. **Reduplication**: affixes are derived from partial or complete copies of the stem.

6. **Template filling**: inflections are formed from roots using an applied pattern of affixation, vowel insertion, and other phonological changes.

Table 1.2 illustrates each of these phenomena and Table 1.3 shows the distribution of these phenomena across the space of languages investigated in this thesis.

1.4 Applications of Inflectional Morphological Analysis

1.4.1 Dimensionality Reduction

For many applications, such as information retrieval (IR), inflectional morphological variants (such as *swim*, *swam*, *swims*, *swimming*, and *swum*) typically carry the same core semantic meaning. The differences between them may capture temporal information (such as past, present, future), or syntactic information (such as nominative or objective case). But they all essentially have the same meaning of "directed self-propelled human motion through the water", and the tense itself is largely irrelvant for many IR queries.

Google, the popular internet search engine, currently does not perform automatic morphological clustering when searching for related information on queries. Thus, trying to find people who've swam across the English Channel using the query "swim English Chan-

affixation				
prefixation:	geuza	↔	**mli**geuza	(*Swahili*)
suffixation:	adhair	↔	adhair**im**	(*Irish*)
	sleep	↔	sleep**ing**	(*English*)
circumfixation:	mischen	↔	**ge**misch**t**	(*German*)
infixation:	palit	↔	p**um**alit	(*Tagalog*)
point-of-affixation changes				
	pla**c**er	→	pla**ç**a	(*French*)
	zwer**f**t	→	zwer**v**en	(*Dutch*)
elision:	clos**e**	→	closing	(*English*)
gemination:	stir	→	stir**r**ed	(*English*)
vowel harmony and internal vowel shifts				
internal vowel shift:	afbr**y**de	→	afbr**ø**d	(*Danish*)
	skr**i**ke	→	skr**ei**k	(*Norwegian*)
	sw**ee**p	→	sw**e**pt	(*English*)
vowel harmony:	abartmak	→	ab**a**rtm**a**s**a**nız	(*Turkish*)
	addetmek	→	add**e**tm**e**s**e**niz	(*Turkish*)
agglutination and reduplication				
reduplication:	habol	→	**ha**habol	(*Tagalog*)
agglutination:	habol	→	**ma**hahabol	
agglutination:	habol	→	ma**ka**hahabol	
agglutination:	ev	→	ev**de**	(*Turkish*)
agglutination:	evde	→	evde**ki**	
agglutination:	evdeki	→	evdeki**ler**	
reduplication:	rumah	→	**rumah**rumah	(*Malay*)
reduplication:	ibu	→	**ibu**ibu	
root-and-pattern (templatic morphology)				
	ktb	→	k**a**t**e**b	(*Arabic*)
	ktb	→	k**a**tt**a**b	
highly irregular forms				
	fi	→	erai	(*Romanian*)
	jānā	→	gayā	(*Hindi*)
	eiga	→	áttum	(*Icelandic*)
	go	→	went	(*English*)

Table 1.2: Examples of cross-lingual morphological phenomenon

Language	pre-fix	suf-fix	in-fix	circum-fix	agglut-inative	redup-licative	vow. harm.	word order
Spanish	-	✓	-	-	-	-	-	SVO
Portuguese	-	✓	-	-	-	-	-	SVO
Catalan	-	✓	-	-	-	-	-	SVO
Occitan	-	✓	-	-	-	-	-	SVO
French	-	✓	-	-	-	-	-	SVO
Italian	-	✓	-	-	-	-	-	SVO
Romanian	-	✓	-	-	-	-	-	SVO
Latin	-	✓	-	-	-	-	-	SOV/free
English	-	✓	-	-	-	-	-	SVO
German	✓	✓	✓	✓	-	-	-	V2
Dutch	✓	✓	✓	✓	-	-	-	V2
Danish	-	✓	-	-	-	-	-	SVO
Norwegian	-	✓	-	-	-	-	-	SVO
Swedish	-	✓	-	-	-	-	-	SVO
Icelandic	-	✓	-	-	-	-	-	SVO
Czech	✓	✓	-	-	-	-	-	SVO/free
Polish	✓	✓	-	-	-	-	-	SVO/free
Russian	✓	✓	-	-	-	-	-	SVO/free
Irish	✓	✓	✓	-	-	-	✓	VSO
Welsh	-	✓	-	-	-	-	-	VSO
Greek	✓	✓	-	-	-	-	-	SVO
Hindi	-	✓	-	-	-	-	-	SOV
Sanskrit	-	✓	-	-	-	-	-	SOV/free
Estonian	-	✓	-	-	✓	-	-	SVO
Finnish	-	✓	-	-	✓	-	✓	SVO
Turkish	-	✓	-	-	✓	-	✓	SOV
Uzbek	-	✓	-	-	✓	-	✓	SOV
Tamil	-	✓	-	-	✓	-	-	SOV
Basque	-	✓	-	-	✓	-	✓	SOV
Tagalog	✓	✓	✓	-	✓	✓	-	SVO
Swahili	✓	✓	-	-	✓	-	-	SVO
Klingon	✓	-	-	-	✓	-	-	OVS

Table 1.3: A cross-section of verbal inflectional phenomenon and word ordering for languages presented in this thesis. Excluded are languages such as Malay with whole-word reduplication, and Semitic languages such as Hebrew and Arabic with templatic morphologies. Word order refers to the syntax of the language. An "SVO" language means that the Subject, Verb and Object of the sentence generally appear in the order "Subject-Verb-Object.

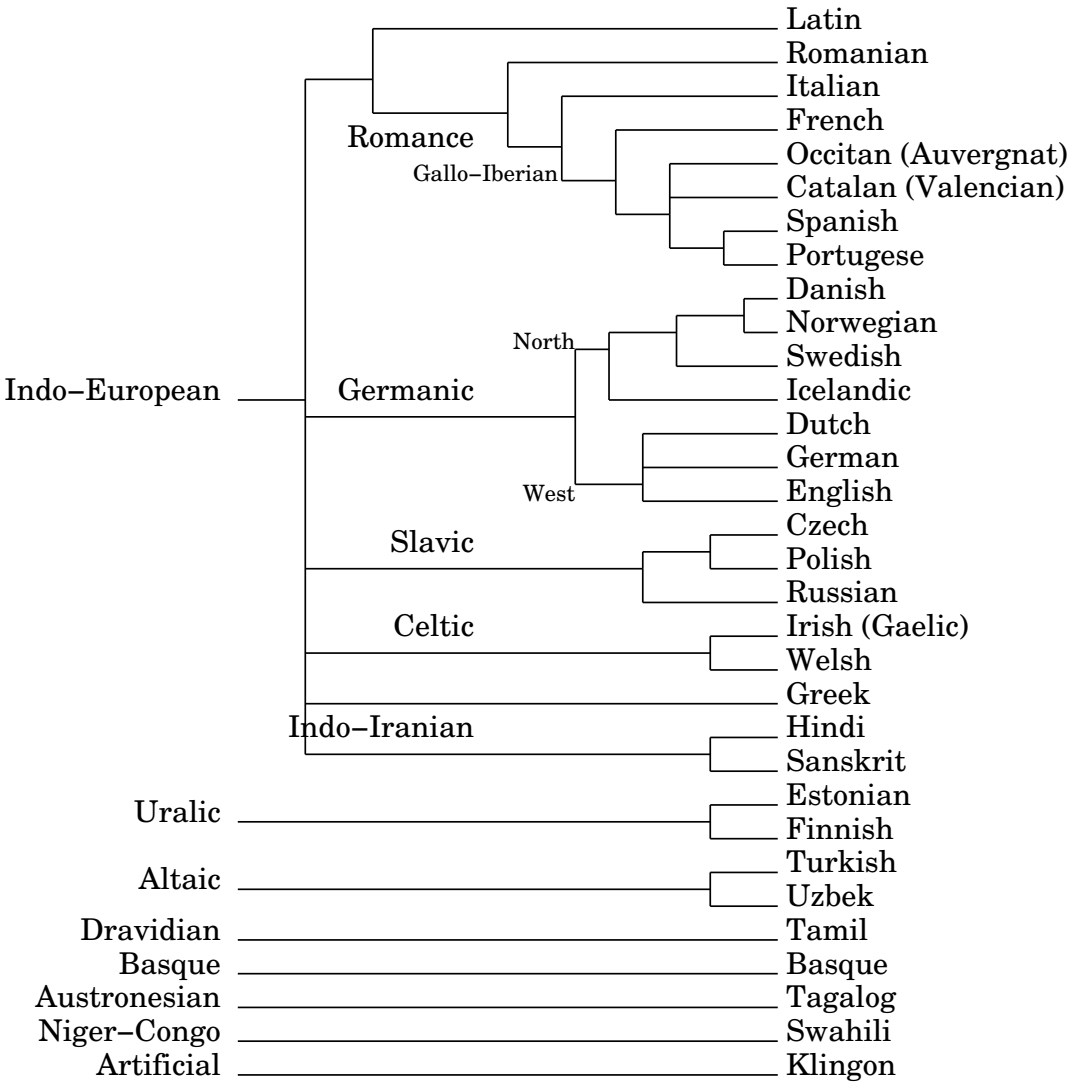


Figure 1.1: Language families represented in this thesis

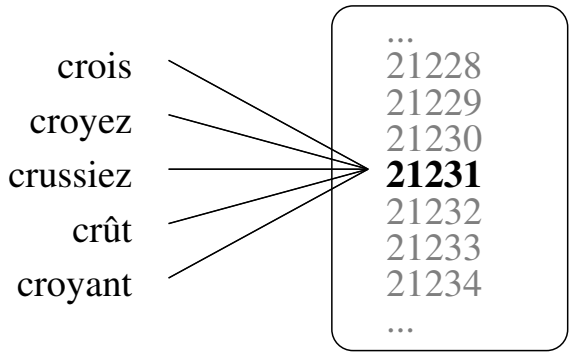


Figure 1.2: For dimensionality reduction, the important property is that the inflectional variants are clustered together; the actual label of the cluster is less meaningful.

nel" turns up a different data set than searching for "swam English Channel".[1] This means that searching for a particular document about swimming the English Channel has been split into five potentially non-overlapping sets of query results, each requiring a separate search.[2]. For highly inflected languages such as Turkish (where each verb in the test used here has an average of nearly 335 inflections) the problem is much more severe, and can cause substantial sparse data problems.

For the purposes of dimensionality reduction, the most important property is that the inflectional variants are clustered together. The actual label of the cluster is less meaningful, and fully unsupervised clustering of terms often achieves the desired functionality.

Dimensionality reduction is also appropriate for feature space simplification for other classification tasks such as word sense disambiguation (WSD). In WSD, the meaning of a word such as *church* (a place of worship vs. an institution vs. a religion) can be partially

[1]Searching for the "swim English Channel" finds 45100 documents, "swam English Channel" finds 10800, "swum English Channel" finds 1730, "swimming English Channel" finds 79700, and "swims English Channel" finds 5570".

[2]Alternatively, all of these inflections can be combined into a single query with all the inflections separated by OR

distinguished using words in context or in specific syntactic relationships. For example, the collocation "*build/builds/building/built a church*" typically indicates that the church has the sense of "a place of worship". Ideally, if there is evidence that "*builds a church*" is an indication of this first sense, this should automatically be extended to "*built a church*" without the need to observe both inflections separately. This is dictinct from the model simplification that results by merging the sense models for *church* and *churches* together using morphological clustering of the polysemous keywords as well as their features.

1.4.2 Lexicon Access

For other applications, the important problem is identifying, for a particular inflected word, its standardized form in a lexicon or dictionary. A typical need for this is in machine translation (MT), where one needs to first know that the root form of *swum* is *swim* in order to discover its translation into Spanish as *nadar*, into German as *schwimmen* or into Basque as *igeri*. It's not sufficient simply to recognize that *swim, swam, swimming, swims,* and *swum* refer to the same concept or cluster, but to assign a name to that cluster that corresponds to the name used in another resource such as a translation dictionary.

While crude stemming (truncating the endings of *computes, computed,* and *computing,* to obtain *comput*, as done by a standardized IR tool such as the Porter Stemmer [Porter, 1980]) may be sufficient for clustering of terms, it does not match the conventional dictionary citation form. Correctly identifying the name of the lemma as *compute* rather than *comput* is essential for successful lookup in a standard dictionary.

Alternatively, the dictionary can also be stemmed and then these stems can be looked up in the altered dictionary. However, stemming often conflates two distinct words

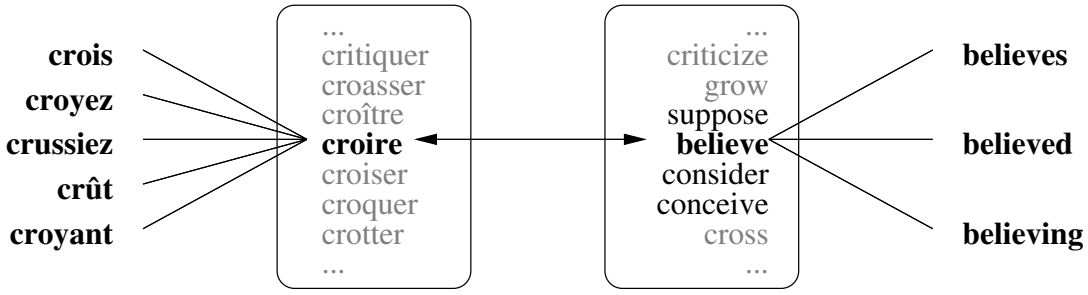


Figure 1.3: For machine translation, the important problem is one of lexicon access: identifying the inflection's standardized form in a lexicon so that its translation can be discovered.

by truncating the endings of words. For example, *sing* and *singe* may both be conflated to *sing*. Stemming *sings* would now involve a dictionary lookup which defined *sing* as both the meanings of *sing* and *singe*.

1.4.3 Part-of-Speech Tagging

For other applications, the tense, person, number, and/or case of a word are also important, and a morphological analyzer must not only identify the lemma of an inflection, but also its syntactic features. To perform this fully requires context, as there is often ambiguity based on context. For example, the word *hit* can be the present or past tense of the verb *hit*, the singular noun *hit*, or the adjective *hit*[3]. But the process of lemmatization often yields information that can be useful in part-of-speech analysis, such as identifying the canonical affix of an inflection, which can be used as an information source in the further mapping to tense. In addition, the affix and stem change processes may be highly informative in terms of predicting the core part of speech of a word (noun or verb). These issues will be only covered briefly in this thesis, but the analyses performed here do have

[3]As in, "The hit batter walked to first base."

value for the larger goal of part-of-speech tagging and inflectional feature extraction.

1.5 Thesis overview

1.5.1 Target tasks

Throughout this thesis, the primary focus will be on the task of lemmatization. Results covering thirty-two languages will presented. While the majority of the languages are Indo-European, the range of morphological phenomena that will be tested is extensive and is limited only by the availability of evaluation data.[4]

The task of morphological analysis will first be presented in a fully supervised framework (Chapter 3). Four supervised models will be introduced and each will have separate evaluation along with a discussion on the strengths and weaknesses of the model. Next, morphological analysis will be presented in an unsupervised framework (Chapter 4). Four unsupervised similarity measures will be used to align inflections with potential roots. While none of these models will be sufficient on their own to serve as a stand-alone morphological analyzer, their output will serve to bootstrap the supervised models (Chapter 5). The accuracy of the models that result from this unsupervised bootstrapping will often approach (and even exceed) the accuracy achieved by the fully supervised models.

1.5.2 Supervised Methods

Four supervised methods are presented in Chapter 3. Each uses a trie-based model to store context-sensitive smoothed probabilities for point-of-affixation changes.

[4]With the exception of templatic languages such as Arabic, and fully reduplicative languages such as Malay, which were intentionally omitted.

The first model, the *Base* model, will treat all string transductions necessary to transform an inflection into a root as a word-final rewrite rule. In this model, only end-of-string changes are stored in the trie. Although unable to handle any prefixation, and limited in its ability to capture point-of-suffixation changes which are consistent across multiple inflections of the same root, this model is remarkably accurate across a broad range of the evaluated languages.

The second model, the *Affix* model, will handle prefixation and suffixation as separable processes from the single point-of-suffixation change. In this model, only purely concatenative prefixation can be handled (e.g. it results in no point-of-prefixation change). Additionally, the Affix model relies on user-supplied lists of prefixes, suffixes, and root endings.

The third and fourth models, the *Wordframe* models, are able to model point-of-prefixation changes, point-of-suffixation changes, and also a single internal vowel change. The third model, *WFBase*, is the Wordframe model built upon the Base model. WFBase treats the merged affix and point-of-affixation change as a single string transduction which is stored in the trie. The fourth model, *WFAffix*, is the Wordframe model built upon the Affix model. WFAffix separates the representation of the affix from the point-of-affixation change by using user-supplied lists of prefixes, suffixes, and endings.

Finally, these models will be combined to achieve accuracies which are higher than accuracies from any of the individual models.

1.5.3 Unsupervised Models

Although the supervised models perform quite well, direct application of the supervised models to new languages is limited by the availability of training data. The unsupervised models introduced in Chapter 4 will present four diverse similarity measures which will perform an alignment between a list of potential inflections and a list of potential roots.

The first unsupervised model is based on the frequency distributions of the words in an unannotated corpus. The intuition behind this model is that inflections which occur with high frequency in a corpus should align with roots which also occur with high frequency in the corpus, and vice-versa, and, in general, should exhibit "compatible" frequency ratios. Since many inflections and roots occur with similar frequencies in a corpus, the primary purpose of this model is not to create initial alignments between roots and inflections, but rather to serve as a filter for the alignments proposed by the remaining similarity measures.

The second unsupervised model uses an unannotated corpus to find contextual similarities between inflections and roots. This contextual similarity is able to identify the a small set of potential roots in which the correct root is found between 25-50% of the time. The contextual models have a number of parameters which have a large impact on the final performance of the model and this will be extensively evaluated.

The third unsupervised model uses a weighted variant of Levenshtein distance to model the orthographic distance between an inflection and its potential root. This model is able to perform substitutions on vowel clusters and consonant clusters, as well as being able to perform these substitutions on individual characters. In addition, a position-sensitive weighting factor is employed to better model the affixation patterns of a particular language.

The final unsupervised model uses a word-aligned, bilingual corpus (bitext) between a language for which a morphological analyzer already exists and a language for which one does not exist. This model uses the analyzer to find the lemmas of all the words in one language. The alignments are then projected across the aligned words to form inflection-root alignments in the second language. On its own, this model is quite precise; however, it is limited by the availability and coverage of the bitext.

1.5.4 Model Combination and Bootstrapping

While none of the unsupervised models is capable of serving effectively as a stand-alone analyzer, the output of these models can be used as noisy training data for the supervised models. In addition, the output of the unsupervised models can be used to estimate an effective weighting for the combination of the four supervised models. The output of the supervised models trained on the unsupervised alignments can then be used estimate an improved parameterization of the unsupervised models. Iteratively retrained, the combination of the unsupervised and supervised models is capable of approaching (and even exceeding) the performance of the fully supervised models.

1.5.5 Evaluation

Throughout the thesis, there will be three statistics used to measure the performance of a model. The most frequently used measure is that of accuracy, which is the number of inflections for which the correct root was found, divided by the total number of inflections in the test set. Less often, precision will be used to measure the performance of a model. Precision is the number of inflections for which the root was correctly identified

divided by the total number of inflections for which a root was identified at all. This is important in the supervised models where a threshold is used to filter out low-confidence alignments. Finally, the coverage of the models will be cited. The coverage is simply the number of inflections that were aligned with a (correct or incorrect) root divided by the total number of inflections in the test set. This measure is used throughout the section describing the contextual model (Section 4.4) since the performance of this model is limited by the extent to which the roots and inflections in the test set actually appear in the corpus.

In addition, the notion of "correct," as used throughout the thesis, is, unless otherwise specified[5], whether or not the correct root was identified for a particular inflection. Whether or not a learned string transduction represented a linguistically plausible explanation for the inflection is not used in any way in this thesis. No claims are made about the models presented here having the ability to find linguistically plausible analyses. Often words will be analyzed as having the correct root using extremely implausible word formation rules. In these cases, the since the root is correctly identified, the example is deemed correct. Likewise, there is no inductive bias favoring models with cognitively plausible behavior and error, and unlike in much previous work, the similarity between errors made by the model and errors made by typical adults or child learners is not considered in evaluating alternative models.

Without the use of evaluation data, it is difficult to measure the effectiveness of a morphological analyzer. One way to do this, not presented in this thesis, is to measure its success in performing a second task. Florian and Wicentowski [2002] present results for

[5]Such as in the section on morphological generation

the task of word sense disambiguation, both with and without the analysis tools presented here, and achieved statistically significant performance gain with the inclusion of the morphological analyzer. Such downstream tasks or applications can be used to measure the relative success of diverse morphological models directly on their intended applications.

1.5.6 Data Sources

To obtain the evaluation pairs for this thesis, a random sample of root forms was taken from mono- and bilingual dictionaries. These forms were then inflected using a variety of existing systems[6] These systems were also the source of the classifications *Regular*, *Semi-Regular*, *Irregular*, and *Obsolete*, used throughout the evaluation sections of this thesis.

The accuracy of these classifications was somewhat inconsistent across languages, with the major fault being that irregular conjugations were listed as being regular. Therefore, the actual performances on each of these sets is, at best, an approximation of the actual performance achieved for each classification.

In addition, since the output of this system was judged against the evaluation data generated by other systems, there will always be cases where the models presented here have not learned the correct inflectional analysis, but rather have mimicked the analysis presented in the training data. However, since a variety of sources have been used, and, in many cases, the number of inflections tested on is quite large, this effect should be somewhat mitigated.

Appendix A shows the number of roots and inflections available for evaluation and training. In the supervised models, all forms were tested by using 10-fold cross-validation[7]

[6] Including many available on the internet.

[7] Trained on $\frac{9}{10}$ of the data, and evaluated on $\frac{1}{10}$

on the all the available evaluation data

1.5.7 Stand-alone morphological analyzers

For many applications, once the number of analyzed inflections achieves sufficiently broad coverage, these inflection-root pairs effectively become a fast stand-alone morphological analyzer by simple table lookup. Of course, this will be independent of any necessary resolution between competing "correct" forms which may only be resolved by observing the form in context.[8] While the full analyzer (or generator) that was used to create such an alignment will remain useful for unseen words, such words are typically quite regular, so most of the difficult substance of the lemmatization problem can often be captured by a large **(inflection, part-of-speech)** $\rightarrow$ **root** mapping table and a simple transducer to handle residual forms. This is, of course, not the case for agglutinative languages such as Turkish or Finnish, or for very highly inflected languages such as Czech, where sparse data becomes an issue.

[8]For example, *axes* may be the plural of *axis* in a math textbook whereas it may be the plural of *axe* in a lumberjack's reference.

Chapter 2

Literature Review

2.1 Introduction

This chapter will provide an overview of the most closely related prior work in both supervised and unsupervised computational morphology.

2.2 Hand-crafted morphological processing systems

The hand-crafted two-level model of morphology developed by Koskenniemi [1983], also referred to as KIMMO, has been a very popular and successful framework for manually expressing the morphological processes of a large number of the world's languages. The KIMMO approach uses individual hand-crafted finite-state models to represent context-sensitive stem-changes. Each finite-state machine models one particular affixation or point-of-affixation stem change.

The notation that is used in this thesis to describe affixation and associated stem

changes have been partially inspired by the KIMMO framework. For example, a two-level equivalent capturing *happy + er = happier* is *y:i* $\Leftrightarrow$ *p:p* _, is quite similar in spirit and function to the basic probabilistic model $P(y{\rightarrow}i|...app, +er)$ presented in Section 3.3.1.

While there has been recent work in learning two-level morphologies (see Section 2.3.3), under its standard usage, a set of two-level rules need to be hand-crafted for all observed phenomena in a language. For the breadth of languages presented in this thesis, such an achievement would be extremely difficult for any one individual in a reasonable time frame. And, once trained, such rule-based systems are not typically robest in handling unseen irregular forms.

2.3 Supervised morphology learning

2.3.1 Connectionist approaches

Historically, computational models of morphology have been motivated by one of two major goals: psycholinguistic modeling and support of natural language processing tasks. The connectionist frameworks of Rumelhart and McClelland [1986], Pinker and Prince [1988], and Sproat and Egedi [1988], all tried to model the psycholinguistic phenomenon of child language learning as it was manifested in the acquistion of the past tense of English verbs. Inflection-root paired data was used to train a neural network which yielded some behavior that appeared to mimc some learning patterns observed in children. These approaches were designed explicitly to handle phonologically represented string transductions, handled only English past tense verb morphology, and were not effective at predicting analyses of irregular forms that were unseen in training data.

2.3.2 Morphology as parsing

Morphology induction in agglutinative languages such as Turkish, Finnish, Estonian, and Hungarian, presents a problem similar to parsing or segmenting a sentence, given the long strings of affixations allowed and the relatively free affix order. Karlsson et al. [1995], have approached this problem in a finite-state framework, and Hakkani-Tür et al. [2000] have done so using a trigram tagger, with the assumption of a concatenative affixation model.

This research has focused on the agglutinative languages Finnish and Turkish. The models presented in this thesis do not attempt to directly handle unrestricted agglutination beyond standard inflectional morphology. Agglutinative morphologies generally have very free "word" (morpheme) order; yet, these morphologies are often extremely regular in that there are few point-of-affixation or internal stem changes. Although agglutinative languages tend to have free morpheme order, much like many free word order grammars, their morphologies generally have a 'default' affix ordering which can be learned effectively, as shown in Chapter 3.

2.3.3 Rule-based learning

Mooney and Califf [1995] used both positive (correct) and negative (randomly generated incorrect) paired training data to build a morphological analyzer for English past tense using their FOIDL system (based on inductive logic programming and decision lists).

Theron and Cloete [1997] sought to learn a 2-level rule set for English, Xhosa and Afrikaans by supervision from approximately 4000 aligned inflection-root pairs extracted

from dictionaries. Single character insertion and deletions were allowed, and the learned rules supported both prefixation and suffixation. Their supervised learning approach could be applied directly to the aligned pairs induced in this paper.

The 2-level rules of Theron and Cloete [1997] are modeled with the framework of KIMMO's 2-level morphology. Since the rules are written for the KIMMO system, their representational framework has the same weaknesses as KIMMO regarding their inefficiency in generalizing to previously unseen irregular forms.

Oflazer and Nirenburg [1999] and Oflazer et al. [2000] have developed a framework to learn a two-level morphological analyzer from interactive supervision in a Elicit-Build-Test loop under the Boas project. Language specialists provide as-needed feedback, correcting errors and omissions. Recently applied to Polish, the model also assumes concatenative morphology and treats non-concatenative irregular forms through table lookup.

These active learning methods could be used as a way of training the supervised methods presented in this thesis.

Recently, Clark [2002] has built a memory-based supervised phonological morphology system to handle English past tense, Arabic broken plurals as well as German and Slovene plural nouns. This model performs well on regular morphology but does quite poorly on irregular morphology. In addition, these models were trained and tested on single part of speech training data and applied to very small test sets, making it difficult to directly compare this work. Previously, Clark [2001a] devised a completely supervised method for training stochastic finite state transducers (Pair Hidden Markov models). Again, this work is completely supervised, is tested only on a single part of speech at a time, and does

poorly on irregulars. However, the FST framework he uses is quite capable of handling the internal changes handled by the supervised model presented in Chapter 3. Indeed, Clark [2001b] builds an unsupervised version of his FST work which, "is closely related to that of Yarowsky and Wicentowski, 2000" by recasting some of the work presented here in terms of finite state transducers.

2.4 Unsupervised morphology induction

2.4.1 Segmental approaches

Kazakov [1997], Brent et al. [1995], Brent [1999], de Marcken [1995], Goldsmith [2001], and Snover and Brent [2001], have each focused on the problem of unsupervised learning of morphological systems as essentially a segmentation task, yielding a morphologically plausible and statistically motivated partition of stems and affixes. De Marcken [1995] approaches this task from a psycholinguistic perspective; the others primarily from a machine learning or natural language processing perspective. Each used a variant of the minimum description length framework, with the primary goal of inducing a segmentation between stems and affixes.

Goldsmith [2001] specifically sought to induce suffix paradigm classes (for example, {*NULL.ed.ing*}, {*e.ed.ing*}, {*e.ed.es.ing*} and {*ted.tion*}) from distributional observations over raw text. Irregular morphology was largely excluded from these models, and a strictly concatenative morphology without stem changes was assumed.

These works have largely been applied only to English, though Kazakov has presented some limited results in French and Latin morphology. All of these works are focused

on the task of segmenting words into their constituent morphemes, which is not the same task as finding the root form. In segmentation, precision is measured by whether or not a line can be correctly drawn between the constituent morphemes, while generally ignoring stem changes or point-of-affixation changes. This crucial distinction means that these segmentalist approaches may find the stem of "closing" to be "clos", not "close". Yielding such non-standard roots is deficient, as previously noted in tasks where standardized lexicon access after analysis is important. Such segmentation does yield some dimensionality reduction needed for information retrieval and word sense disambiguation. However, since different infections of the same root are often reduced to different stems (e.g. "closed" and "closing" are segmented to "clos" but "closed" and "close" are segmented to "close"), fully compatible clustering is not achieved.

2.5 Non-segmental approaches

Schone and Jurafsky [2000] initially developed a supervised method for discovering English suffixes using trie-based models. Their training pairs were derived from unsupervised methods including latent semantic analysis and distributional information. Similar to the work done in segmentation (Section 2.4.1), this work did not attempt to do morphological analysis, such as lemmatization, but rather tried to discover the suffixes for a given language. However, unlike this work, or the work of the many of the segmentalists, and similar to the work of Goldsmith [2001], Schone and Jurafsky [2000] attempted to actually identify the morphemes, not just the morpheme boundaries.

Schone and Jurafsky [2001] later extended this work from finding only suffixes to

finding prefixes and circumfixes. However, similar to Schone and Jurafsky [2000], this work is designed purely to identify the regular affixes in the language, not to do lemmatization. The resulting lists of prefixes and suffixes can be beneficial to the supervised morphology algorithms presented in Chapter 3. They "expect improvements could be derived [from their work], which focuses primary on inducing regular morphology, with that of Yarowsky and Wicentowski [2000] ... to induce some irregular morphology".

In work somewhat derivative of that in Yarowsky and Wicentowski [2000] and Schone and Jurafsky [2001], Baroni et al. [2002] use string edit distance and semantic similarity to find inflection-root pairs which are then used to train a finite-state transducer.

2.6 Learning Irregular Morphology

There is a notable gap in the research literature for the induction of analyzers for irregular morphological processes, including substantial stem changing. The set of algorithms presented in this thesis directly addresses this gap, while successfully inducing regular analyses without supervision as well.

2.7 Final notes

The work presented in this thesis is an original paradigm of morphological analysis and generation based on three original methods:

1. Completely unsupervised morphological alignment based on frequency similarity, context similarity, and Levenshtein similarity

2. Supervised trie-based morphology capable of identifying prefixes, suffixes, point-of-affixation changes, and internal vowel shifts

3. Projection of morphological analyses onto a second language through translingual projection over parallel corpora by leveraging existing morphological analyzers in one (or more) source languages

While originally reported in Yarowsky and Wicentowski [2000] and a section of Yarowsky et al. [2001], the approaches and algorithms presented exclusively in this text also constitute a substantial original contribution to the complete body of morphology induction research, comprehensively reported in this thesis.

Chapter 3

Trie-based Supervised Morphology

3.1 Introduction

One approach to the problem of morphological analysis is to learn a set of string transductions from inflection-root pairs in a supervised machine learning framework. These string transductions can then be applied to transform unseen inflections to their corresponding root forms.

Since this work is applied to a broad range of the world's languages[1], the string transductions must be able to describe the inflectional phenomena found in these languages, as presented in Table 1.2. To this end, the supervised algorithms presented here use a set of linguistically motivated patterns to constrain the set of potential string transductions.[2] These patterns directly model prefixation and suffixation, the associated point-of-affixation

[1] Languages with templatic morphologies (such as Arabic) and languages with whole word reduplication (such as Malay) have been excluded from this work.

[2] While these string transductions may potentially resemble linguistic word formation rules, this work makes no claims about its ability to actually model the underlying linguistic processes involved with inflectional morphology.

changes, and internal vowel shifts.

Circumfixation is modeled as disjoint prefixation and suffixation patterns; however, infixation is not yet modeled. Vowel harmony and agglutination, insofar as they are manifested through prefixation, suffixation, and stem changes, are modeled as single string transduction patterns.

Partial word reduplication, such as is found in Tagalog, is not explicitly modeled, but patterns generated from large amounts of training data can be reasonably effective for finding root forms. No attempt has been made either to model whole word reduplication, such as is found in Malay, or the templatic roots often found in Semitic languages such as Arabic, Hebrew and Amharic.

Highly irregular forms can only be handled through memorization of specific irregular pairings; however, many inflections described as "irregular" are actually examples of unproductive and infrequently occuring morphological phenomenon which, when observed in training data, are capable of providing productive supervision to other inflections with similar behavior.

3.1.1 Resource requirements

Training data of the form $< inflection, root, POS >$, or simply $< inflection, root >$, is required for supervised morphological analysis. For many of the world's major languages, and for all of the languages for which results are presented here, morphological training data of this type can be obtained either from on-line grammars or extracted from printed materials which have been hand-entered or scanned. Unfortunately, morphological training data is not available for many languages with non-Roman character sets, low-density languages,

languages which are largely oral, or most extinct languages. For some of these languages, there may be only a few language experts who can be used to fill this void. When experts or native language speakers are available, using them to hand-enter training data can be prohibitively expensive. To address this issue, Chapter 4 will present four independent similarity measures which can provide noisy inflection-root seed pairs without the use of any morphological training pairs for supervision. Chapter 5 will then provide evaluation of the supervised models when bootstrapped from this noisy training data.

3.1.2 Terminology

This section will serve to further clarify the terminology used in describing the models presented and phenomena observed in this thesis. Section 1.3, which presented an introduction to this terminology, will serve as a foundation here and a familiarity with that material will be assumed here.

The following definitions should be used as a reference to help understand the model framework presented in Section 3.2. In addition, refer to Table 3.1 for additional examples of the terms below.

- A *prefix* is a bound morpheme which attaches to the beginning of an inflection. In English, there are no examples of prefixes being used in inflectional morphology, the focus of this work. There are examples of English derivational prefixes which include *un+*, *non+*, and *dis+*, used generally to form the negative of the words to which they attach.

- A *suffix* is a bound morpheme which attaches to the end of an inflection. Examples of

English suffixes includes the suffix *+ed*, which indicates past tense of verbs, the suffix *+s* which indicates the third-person present tense of verbs (*he jumps*) and also the canonical plural morpheme for nouns.

- A *canonical prefix* or *canonical suffix* is a prefix or suffix which is found as part of a regular word formation rule. So, while *+s* would be a canonical suffix for English nouns, *+en*, which forms the plural of *children* and *oxen* is not considered a canonical suffix. Throughout this thesis, the terms prefix and suffix are meant always to refer to these canonical prefixes and canonical suffixes.

- A *canonical ending* is a bound morpheme which is attached to the end of a root. English does not make use of canonical endings, but is widely used in other languages. For example, all French verbs must end *+er, +ir,* or *+re*[3], all Spanish verbs must end *+ir, +ar,* or *+er*, and all Estonian verbs must end *+ma*. Canonical endings, when present for a particular part-of-speech, are usually required for all roots of this part-of-speech in the language. In addition, when forming inflections of roots with canonical endings, these endings often must be removed before adding other morphemes.

- A *point-of-affixation change* is a change in the orthographic representation of the word which occurs at the location of an affixation. For example, forming the past tense of *cry* involves adding the suffix *+ed*. When this suffix is added, the final *y* of *cry* is changed to *i*. Hence, the point-of-suffixation change here is $y \rightarrow i$.

Further details on how each of these is used in the models will be presented in

[3]With 2 exceptions which end *+ïr*.

Section 3.4.2.

3.2 Supervised Model Framework

Four supervised learning algorithms are presented here: two primary algorithms, each with two separate components. The two main algorithms are the point-of-suffixation model, presented in Section 3.3, and the Wordframe-based models, described in Section 3.5. Both models generate morphological patterns from training data and analyze test data using these patterns.

The components are distinguished by the set of constraining templates used to generate the morphological patterns. If a list of canonical prefixes, canonical suffixes and canonical endings has been provided by the user, one component is used; otherwise, the other component is used. Table 3.1 presents an example list of these affixes and endings for Spanish.

This defines four models: two point-of-suffixation models called the **Base** model (which does not use user-supplied affix lists) and the **Affix** model (which does use these lists), and two Wordframe models called the **WFBase** model (which is the Wordframe model built without user-supplied lists) and the **WFAffix** model (which does use these lists).

As will be shown in Table 3.32, no one system performs best across all languages. Each model outperforms the other models for some of the languages, and for some of the examples in each language. For this reason, as will be shown in Table 3.35, a linear combination of the models outperforms each of the individual models for nearly every language.

Prefixes	ϵ (none)
Suffixes	-o, -a, -as, -amos, -ais, -áis, -an, -es, -e, -emos, -éis, -en, ...
Root Endings	-ar, -er -ir

Table 3.1: Example of prefix, suffix, and root ending lists for Spanish verbs

The outline of the algorithm is the same for all four models. First, training data is analyzed to generate a set of patterns which explain all of the example pairs, and the raw counts of each of the patterns are stored in a trie. Figure 3.1 presents an example of a trie used to store such patterns. Then, for each inflection in the test data, all applicable patterns are applied. The result of each pattern application is a proposed root form which is assigned a confidence score.

The confidence score of the proposed root form depends on how often the patterns that derived it were seen in training data. It may also be affected by the root form's presence or absence in a provided dictionary or corpus-derived wordlist. For many languages, a dictionary or clean list of root forms for the language is available. If this list is reasonably complete[4], only proposed roots which exist in this root list will be considered. For many resource-poor languages, a broad coverage root list will not be available; however, even a small list of root forms from a grammar book or hand-entered by native speakers can be helpful.

The training data used in these models consists of an *unannotated* list of inflection-root pairs, optionally including part of speech tags[5]. Table 3.2 provides example lists of

[4]It is unlikely that a *complete* root list is available given the breadth and continual change of language; however, for many tasks, a broad coverage dictionary is sufficient to get high accuracies.

[5]For a task such as machine translation, fine-grained part-of-speech tags such as "verb, 1st person,

German Verbs		English Verbs		Tagalog Verbs	
inflection	root	inflection	root	inflection	root
...	...	...	...	...	...
machend	machen	builds	build	palitan	palit
gemacht	machen	built	build	pinalitan	palit
freimachend	freimachen	replaced	replace	papalitan	palit
freigemacht	freimachen	replacing	replace	pinapalitan	palit
...	...	...	...	...	...

Table 3.2: Example training data for verbs in German, English, and Tagalog

German, English, and Tagalog training pairs These training pairs *do not* contain an analysis of the morphological processes used to derive the inflection from the root.

The algorithms described here will automatically generate patterns based on the unannotated training data. The automatically generated patterns need not be linguistically plausible, but they must explain the data sufficiently to be able to analyze new forms.

3.2.1 The seven-way split

The foundation for all of the supervised models presented in this thesis is based on a seven-way split of both inflected words and roots, designed to capture prefixation and point-of-prefixation changes, suffixation and point-of-suffixation changes, as well as internal vowel shifts. The following notation will be used when referring to this split:

	canonical prefix/ beginning	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	canonical suffix/ ending
inflection	ψ'_p	δ'_p	γ_p	δ'_v	γ_s	δ'_s	ψ'_s
root	ψ_p	δ_p		δ_v		δ_s	ψ_s

singular, present, indicative" are necessary. For many other tasks, such as IR or word sense disambiguation, coarse-grained POS tags such as "verb" or "noun" are sufficient.

The subscripts p, s and v are mnemonics for *prefix*, *suffix* and *vowel*. δ is used for changed material, and ψ is used for added or subtracted affixes. Examples of inflection-root pairs analyzed according to this seven-way split are found in Table 3.3. Segments with the prime notation (e.g. δ'_s vs. δ_s) indicate segments in the inflection. Segments without this prime represent segments in the root. In this way, it is intentionally the same presentation as given in (1.1).

As will be shown in further detail, the changes represented by ψ are changes which must be derived from the user-supplied lists of canonical prefixes, suffixes and root endings. The changes represented by δ are the residual changes.

This framework was not designed to handle whole-word reduplication or templatic morphologies, hence languages such as Arabic, Hebrew and Malay are excluded from this presentation. Nor was this model designed to handle partial-word reduplication, but as shown in the Tagalog example from Table 3.3, the framework is able to construct a plausible explanation for this phenomenon which, as will be seen in Section 3.5, is reasonably productive.

The key distinctions between each of the models are the extent to which the full power of this seven-way split is utilized. The Base model, to be presented in Section 3.3, handicaps this seven-way split by forcing everything except γ_s, δ'_s, and δ_s to ϵ, the empty string, with the result that all morphological analyses must be modeled as word-final string-rewrite rules.

	$\psi'_p \to \psi_p$	$\delta'_p \to \delta_p$	γ_p	$\delta'_v \to \delta_v$	γ_s	$\delta'_s \to \delta_s$	$\psi'_s \to \psi_s$	
	prefix/ beginning	point-of-prefix. change		vowel change		point-of-suffix. change	suffix/ ending	
warming	ϵ	ϵ	ϵ	ϵ	warm	ϵ	ing	English
warm	ϵ	ϵ		ϵ		ϵ	ϵ	
hopping	ϵ	ϵ	ϵ	ϵ	hop	p	ing	English
hop	ϵ	ϵ		ϵ		ϵ	ϵ	
cries	ϵ	ϵ	ϵ	ϵ	cr	ie	s	English
cry	ϵ	ϵ		ϵ		y	ϵ	
kept	ϵ	ϵ	k	ee	p	ϵ	t	English
keep	ϵ	ϵ		e		ϵ	ϵ	
applaudissons	ϵ	ϵ	ϵ	ϵ	applaud	ϵ	issons	French
applaudir	ϵ	ϵ		ϵ		ϵ	ir	
abrège	ϵ	ϵ	abr	è	g	ϵ	e	French
abréger	ϵ	ϵ		é		ϵ	er	
gefallen	ge	ϵ	ϵ	ϵ	fall	ϵ	en	German
fallen	ϵ	ϵ		ϵ		ϵ	en	
gefielt	ϵ	ϵ	gef	ie	l	ϵ	t	German
gefallen	ϵ	ϵ		a		l	en	
geacteerd	ge	ϵ	act	ee	r	ϵ	d	Dutch
acteren	ϵ	ϵ		e		ϵ	en	
bIHutlh	bI	ϵ	ϵ	ϵ	Hutlh	ϵ	ϵ	Klingon
Hutlh	ϵ	ϵ		ϵ		ϵ	ϵ	
pinutulan	ϵ	pin	ut	u	l	ϵ	an	Tagalog
putol	ϵ	p		o		ϵ	ϵ	

Table 3.3: Examples of inflection-root pairs in the 7-way split framework.

BASE MODEL

	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection					γ_s	δ'_s	
root						δ_s	

For example:

swept					swe	pt	
sweep						ep	
cried					cr	ied	
cry						y	

The Affix model, to be presented in Section 3.4, allows ψ'_p, ψ'_s, and ψ_s to be taken from a user-supplied list of prefixes, suffixes and endings, which allows for basic modeling of point-of-suffixation changes and simple concatenative prefixation. Examples of inflection-root pairs analyzed under the Base and Affix models are presented in Table 3.4.

AFFIX MODEL

	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection	ψ'_p				γ_s	δ'_s	ψ'_s
root						δ_s	ψ_s
swept					swe	p	t
sweep						ep	
cried					cr	i	ed
cry						y	

The WFBase model, to be presented in Section 3.5, adds the point-of-prefixation change and the internal vowel change, but does not use the canonical affixes/endings. Examples of inflection-root pairs analyzed under the WFBase model are presented in Table 3.5.

WFBASE MODEL

	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection		δ'_p	γ_p	δ'_v	γ_s	δ'_s	
root		δ_p		δ_v		δ_s	
swept			sw	e	p	t	
sweep				ee			

			BASE MODEL			AFFIX MODEL				
inflection	→	root	δ'_s	→	δ_s	δ'_s	ψ'_s	→	δ_s	ψ_s
warming	→	warm	ing	→	ϵ	ϵ	+ing	→	ϵ	+ϵ
hopping	→	hop	ping	→	ϵ	p	+ing	→	ϵ	+ϵ
kept	→	keep	pt	→	ep	p	+t	→	ep	+ϵ
chantons	→	chanter	ons	→	er	ϵ	+ons	→	ϵ	+er
chante	→	chanter	e	→	er	ϵ	+e	→	ϵ	+er
abrège	→	abréger	ége	→	èger	ég	+e	→	èg	+er
gefielt	→	gefallen	fielt	→	fallen	fiel	+t	→	fall	+en
gefallen	→	fallen	gefallen	→	fallen	gefall	+en	→	fall	+en

Table 3.4: Example rules formed when analyzing inflection-root pairs using the Base model and the Affix model. With the Affix model, point-of-suffixation changes can be modeled separately from the suffix, and simple prefixation can be handled.

The WFAffix model, to be presented in Section 3.5, utilizes all of the components of the seven-way split with the exception of ψ_p.[6] Further examples of inflection-root pairs analyzed under this model are presented in Table 3.3.

	WFAFFIX MODEL						
	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection	ψ'_p	δ'_p	γ_p	δ'_v	γ_s	δ'_s	ψ'_s
root		δ_p		δ_v		δ_s	ψ_s
swept			sw	e	p		t
sweep				ee			

As will be seen in both Chapter 3 and Chapter 4, the simpler Base and Affix models outperform the more complex WFBase and WFAffix models for languages which do not have many complex examples that need such increased power of generalization. In addition, the simpler models are trained faster and execute faster, though run-time performance of each of the systems is not formally presented here . On the other hand, the simple Base and

[6] ψ_p is omitted in all of the models presented here because, for the languages studied, there is no linguistically plausible "canonical beginning" of a root form.

			WFBASE MODEL					
			inflection				**root**	
			prefixal change	stem + vowel change	suffixal change		prefixal change	stem + vowel change	suffixal change
inflection	→	root	δ'_p	$\gamma_p[\delta'_v]\gamma_s$	δ'_s	→	δ_p	$\gamma_p[\delta_v]\gamma_s$	δ_s
warming	→	warm	ϵ	warm	+ing	→	ϵ	warm	ϵ
hopping	→	hop	ϵ	hop	+ping	→	ϵ	hop	ϵ
kept	→	keep	ϵ	k[e]p	+t	→	ϵ	k[ee]p	ϵ
chantons	→	chanter	ϵ	chant	ons	→	ϵ	chant	er
chante	→	chanter	ϵ	chant	e	→	ϵ	chant	er
abrège	→	abréger	ϵ	abr[é]g	e	→	ϵ	abr[è]g	er
gefielt	→	gefallen	ϵ	gef[ie]l	t	→	ϵ	gef[a]l	len
gefallen	→	fallen	ge	fallen	ϵ	→	ϵ	fallen	ϵ

Table 3.5: Sample of the rules learned when analyzing inflection-root pairs when using the WFBase model. With the WFBase model, internal vowel changes and point-of-prefixation changes can be modeled. Details on using the WFBase and WFAffix models are presented in Section 3.5. Examples of the WFAffix model are presented in Table 3.3.

Affix models are not able capture the more complex point-of-prefixation changes or internal vowel shifts that must be modeled to perform accurate morphological analysis for some languages.

3.3 The Base model

The Base model of morphological analysis models morphological transformations from inflection to root as word-final string-rewrite rules. While this model is not sufficient for languages with prefixal, infixal and reduplicative morphologies, it is remarkably productive across Indo-European languages (Figure 1.1, Table 1.3). This coverage, in terms of the number of speakers of these languages, the representation of these languages in on-line texts, and, more practically, the availability of training data, all provide motivation for this simple model.

The Base model presented here does not assume the presence of an affix inventory (Table 3.1). This model, the simplest model of morphological similarity, models suffixation, and any potential point-of-suffixation changes, as a single combined end-of-string stem change which converts an inflection into a root (and optional part of speech).

3.3.1 Model Formulation

The Base model presented here is a handicapped version of the seven-way split found in Section 3.2.1 where all substrings except γ_s, δ'_s, and δ_s are set to ϵ, the empty string. The Base model then represents an inflection-root pair as $\gamma_s\delta'_s$ and $\gamma_s\delta_s$, respectively, where γ_s is the maximal common initial string, and δ'_s and δ_s represent the dissimilar final strings of the inflection and root. The Base model, therefore, models the single end-of-string stem change converting $\gamma_s\delta_s$ into $\gamma_s\delta'_s$ as $\delta'_s \rightarrow \delta_s$. Because γ_s is the longest common initial substring, δ'_s and δ_s will never start with the same initial letter.

It is important to note that while $\delta'_s \rightarrow \delta_s$ may describe a particular suffix, it can also describe a combined suffixation and stem-change, a stem-change with no suffixation, or be completely null (Table 3.6).

As it pertains to all the supervised models presented in this thesis, it is often unclear where the "linguistically correct" separation between stem change and suffix occurs, or what the "linguistically correct" morphological analysis should be. For this reason, an algorithm's correctness is not measured by the "correctness" of the analysis, but rather whether or not the inflection's analysis yielded the correct root for which there is generally universal agreement by educated speakers of the language.[7] For example, if the inflection

[7] Or whether or not the root's analysis yielded the correct inflection, in the case of generation.

	inflection	→	root	γ_s	δ'_s	γ_s	δ_s	δ'_s	→	δ_s
Suffix Only	played	→	play	play	ed	play	ϵ	ed	→	ϵ
	jumping	→	jump	jump	ing	jump	ϵ	ing	→	ϵ
Suffix	hurried	→	hurry	hurr	ied	hurr	y	ied	→	y
and	closed	→	close	close	d	close	ϵ	d	→	ϵ
Stem Change	slipped	→	slip	slip	ped	slip	ϵ	ped	→	ϵ
Stem Change	grew	→	grow	gr	ew	gr	ow	ow	→	ew
Only	wrote	→	write	wr	ote	wr	ite	ite	→	ote
No Changes	cut	→	cut	cut	ϵ	cut	ϵ	ϵ	→	ϵ
	read	→	read	read	ϵ	read	ϵ	ϵ	→	ϵ

Table 3.6: End-of-string changes $\delta'_s \rightarrow \delta_s$ in English past tense as modeled by the Affix model with no affix inventory as $\gamma_s\delta'_s \rightarrow \gamma_s\delta_s$

wrote appears in test data and is aligned to the root *write*, it is considered correct, even if the analysis under a particular model (*ote* → *ite* in this case) is not "linguistically correct" under any current theory of morphology.

The morphological distance measure used in this model is $P(\delta'_s \rightarrow \delta_s,\ \pi \mid \gamma_s\delta'_s)$ where π is the part of speech. As formulated, this is equivalent to $P(\gamma_s\delta_s,\ \pi \mid \gamma_s\delta'_s)$ because $\gamma_s\delta_s$ is uniquely specified by the pair $< \gamma_s\delta'_s,\ \delta'_s \rightarrow \delta_s >$. This is true because, as previously mentioned, no rule will have δ'_s and δ_s starting with the same letter.

$$P(\text{root}, \text{part of speech} \,|\, \text{inflection}) = P(\gamma_s\delta_s, \pi \,|\, \gamma_s\delta'_s) = P(\delta'_s \rightarrow \delta_s, \pi \,|\, \gamma_s\delta'_s) \tag{3.1}$$

The alignment probability for a proposed root and part of speech given an inflection is formulated using a backoff model as in (3.2) where λ_i could be determined by the relative training data size, and $last_k(root)$ indicates the final k characters of the root. In all of the experiments done here, $\lambda_i = 0.1$. However, in cases where there is a relatively small amount of training data, or where the training data is noisy, λ_i could be increased, thereby placing

less weight on the leaves of the trie.

$$
\begin{aligned}
P(\delta'_s \rightarrow \delta_s, \pi \,|\, \gamma_s\delta'_s) \approx \lambda_1 P(\delta'_s \rightarrow \delta_s) \cdot P(\pi) + (1-\lambda_1)\cdot \\
\Big[\lambda_2 P(\delta'_s \rightarrow \delta_s, \pi) + (1-\lambda_2)\cdot \\
\Big[\lambda_3 P(\delta'_s \rightarrow \delta_s, \pi \,|\, last_1(\gamma_s\delta'_s)) + (1-\lambda_3)\cdot \\
\Big[\lambda_4 P(\delta'_s \rightarrow \delta_s, \pi \,|\, last_2(\gamma_s\delta'_s)) + (1-\lambda_4)\cdot \\
\Big[\lambda_5 P(\delta'_s \rightarrow \delta_s, \pi \,|\, last_3(\gamma_s\delta'_s)) + (1-\lambda_5)\cdot \Big[\ldots\Big] \quad \Big] \quad \Big] \quad \Big] \quad \Big]
\end{aligned}
\tag{3.2}
$$

The backoff model takes advantage of the observation that inflections with similar endings often undergo the same end-of-string stem change (for a particular π). Table 3.7 illustrates this using Romanian inflection-root verb pairs. The backoff model captures the notion that *ea* $\rightarrow$ *i* is a highly productive pattern for words which end in *ea* (seen in 57.5% of the examples). For inflections ending *gea*, though, a more informed pattern selection would be $a \rightarrow \epsilon$ (74.1%). For inflections ending in *ăgea*, *ăgea* $\rightarrow$ *age* (100.0%) is better still.

The rule $\delta'_s \rightarrow \delta_s$ is considered *applicable* to the inflection $\gamma_s\delta'_s$ if and only if δ'_s represents the final $|\delta'_s|$ characters of $\gamma_s\delta'_s$, which follows from the definition of these strings. All probabilities in (3.2) are interpreted as conditioned also on the applicability of the rule. In other words, $P(\delta'_s \rightarrow \delta_s, \pi \,|last_k(\gamma_s\delta'_s))$ should be written as

$$
P(\text{correct rule is}\,\delta'_s \rightarrow \delta_s, \text{correct POS is}\,\pi \,|last_k(\text{inflection}), \delta'_s \rightarrow \delta_s \,\text{is applicable to}\,\gamma_s\delta'_s)
$$

Inflection Context, h	Stem Change $\delta'_s \rightarrow \delta_s$	$P(\delta'_s \rightarrow \delta_s \mid h)$	count$(\delta'_s \rightarrow \delta_s \mid h)$	Example Inflections
ϵ	ţi → ϵ	4.4%	1048	înjuraţi, vorbiţi, ...
ϵ	ϵ → a	1.1%	264	import, precipit, ...
ϵ	ea → i	0.61%	146	folosea, preferea, ...
ϵ	a → ϵ	0.52%	125	ţopăia, stingea, ...
... a	ea → i	44.6%	146	folosea, preferea, ...
... a	a → ϵ	38.2%	125	ţopăia, stingea, ...
... a	a → î	4.0%	13	târa, bora
... a	a → e	1.2%	4	descria, subscria
... ea	ea → i	57.5%	146	folosea, preferea, ...
... ea	a → ϵ	33.9%	86	stingea, cerea ...
... ea	ăgea → age	2.0%	5	retrăgea, extrăgea, ...
... gea	a → ϵ	74.1%	20	stingea, alegea ...
... gea	ăgea → age	18.5%	5	retrăgea, extrăgea, ...
... gea	ea → i	7.5%	2	fugea, lărgea, ...
... ăgea	ăgea → age	100.0%	5	retrăgea, extrăgea, ...

Table 3.7: Inflections with similar endings often exhibit the same end-of-string stem change.

$$= \begin{cases} \frac{P(\text{correct rule is } \delta'_s \rightarrow \delta_s, \text{correct POS is } \pi \mid last_k(\text{inflection}))}{\sum_{R \text{ is applicable}} P(\text{correct rule is } R \mid last_k(\text{inflection}))} & \text{if } \delta'_s \rightarrow \delta_s \text{ is applicable to } \gamma_s \delta'_s, \\ 0 \quad \text{otherwise.} \end{cases}$$

For example, continuing with the examples from Table 3.7, the rule $ea \rightarrow i$ is applicable to the inflection *folosea*, but the rule $ţi \rightarrow \epsilon$ is not. Therefore, even with $\lambda_1 = 1.0$, $P(\text{ţi} \rightarrow \epsilon, \pi | \text{folosea}) = 0$. In addition,

This probability is not well defined when there are no applicable rules for analyzing a particular inflection. In this case, the probability of all analyses is 0.

Morphological training data, especially when obtained through unsupervised methods, can be lacking fine-grained part-of-speech tags. This is not a problem for many IR and WSD applications, where morphological analysis is performed on words for which the fine-grained part-of-speech is not needed, and the coarse-grained part-of-speech is known or

can be determined. In these cases, the analyzer is trained on only a single part-of-speech, such as VERB or NOUN and (3.2) can be re-written as (3.3).

$$
\begin{aligned}
P(\delta'_s \rightarrow \delta_s \,|\, \gamma_s\delta'_s) \approx \lambda_1 P(\delta'_s \rightarrow \delta_s) + (1-\lambda_1)\cdot \\
\Big[\lambda_2 P(\delta'_s \rightarrow \delta_s \,|\, last_1(\gamma_s\delta'_s)) + (1-\lambda_2)\cdot \\
\Big[\lambda_3 P(\delta'_s \rightarrow \delta_s \,|\, last_2(\gamma_s\delta'_s)) + (1-\lambda_3)\cdot \\
\Big[\lambda_4 P(\delta'_s \rightarrow \delta_s \,|\, last_3(\gamma_s\delta'_s)) + (1-\lambda_4)\cdot \Big[\ldots\Big] \quad \Big] \quad \Big] \quad \Big]
\end{aligned}
\tag{3.3}
$$

One then backs off (as necessary) up to amount available in the training data. The following example from French, (3.4), illustrates (3.3).

$$
\begin{aligned}
P(\text{broyer} \,|\, \text{broie}) = \\
P(\text{yer} \rightarrow \text{ie} \,|\, \text{broie}) \approx \lambda_1 P(\text{yer} \rightarrow \text{ie}) + (1-\lambda_1)\cdot \\
\Big[\lambda_2 P(\text{yer} \rightarrow \text{ie} \,|\, \text{e}) + (1-\lambda_2)\cdot \\
\Big[\lambda_3 P(\text{yer} \rightarrow \text{ie} \,|\, \text{ie}) + (1-\lambda_3)\cdot \\
\Big[\lambda_4 P(\text{yer} \rightarrow \text{ie} \,|\, \text{oie}) + (1-\lambda_4)\cdot \\
\Big[\lambda_5 P(\text{yer} \rightarrow \text{ie} \,|\, \text{roie}) + (1-\lambda_5)\cdot \\
P(\text{yer} \rightarrow \text{ie} \,|\, \text{broie}) \quad \Big] \quad \Big] \quad \Big] \quad \Big]
\end{aligned}
\tag{3.4}
$$

Figure 3.1 shows how the trie-based data structure is used to store these backoff probabilities. In practice, the trie is smoothed using the λ_i values from (3.3) after the patterns are generated to improve run-time performance. As illustrated, the values are left unsmoothed for clarity. Note that in the top 3 nodes of the trie, not all rules are shown. (If they were,

their probabilities would sum to 1 at each node.)

A dictionary of root forms for the target language is often available, and ideally, if this dictionary is complete, one would like a filter to exclude any potential inflection-root alignments for which the root is not listed in the dictionary. When the dictionary is not complete, or when a wordlist is derived from corpus data, instead of excluding alignments, one can downweight the alignment score for roots not found in this list using a weighted filter correlated with the completeness and/or cleanness of the dictionary. Using this root weighting factor, $\omega(\text{root})$, and the stem change probability of (3.3), the similarity score between inflection and root is as in (3.5), with $\omega(root)$ set experimentally to 1.0 for roots found in the dictionary, and 0.001 for roots not found in the dictionary.

$$MorphSim(\text{inflection}, \text{root}) = P(\text{root}, \text{POS}|\text{inflection}) * \omega(root) \tag{3.5}$$

3.3.2 Model Effectiveness

The base stem change model is effective at generating productive stem change patterns when the training pairs exhibit suffixation and point-of-suffixation changes. Table 3.8 shows some high frequency patterns modeled in this way.

This stem change model is much less effective at capturing internal changes and prefixation, as shown in Table 3.9. Here, the simple, productive prefixations of *nina* in Swahili and *maka* in Tagalog have been modeled as whole-word replacements which apply to no other examples in the training set. The combination of a point-of-suffixation

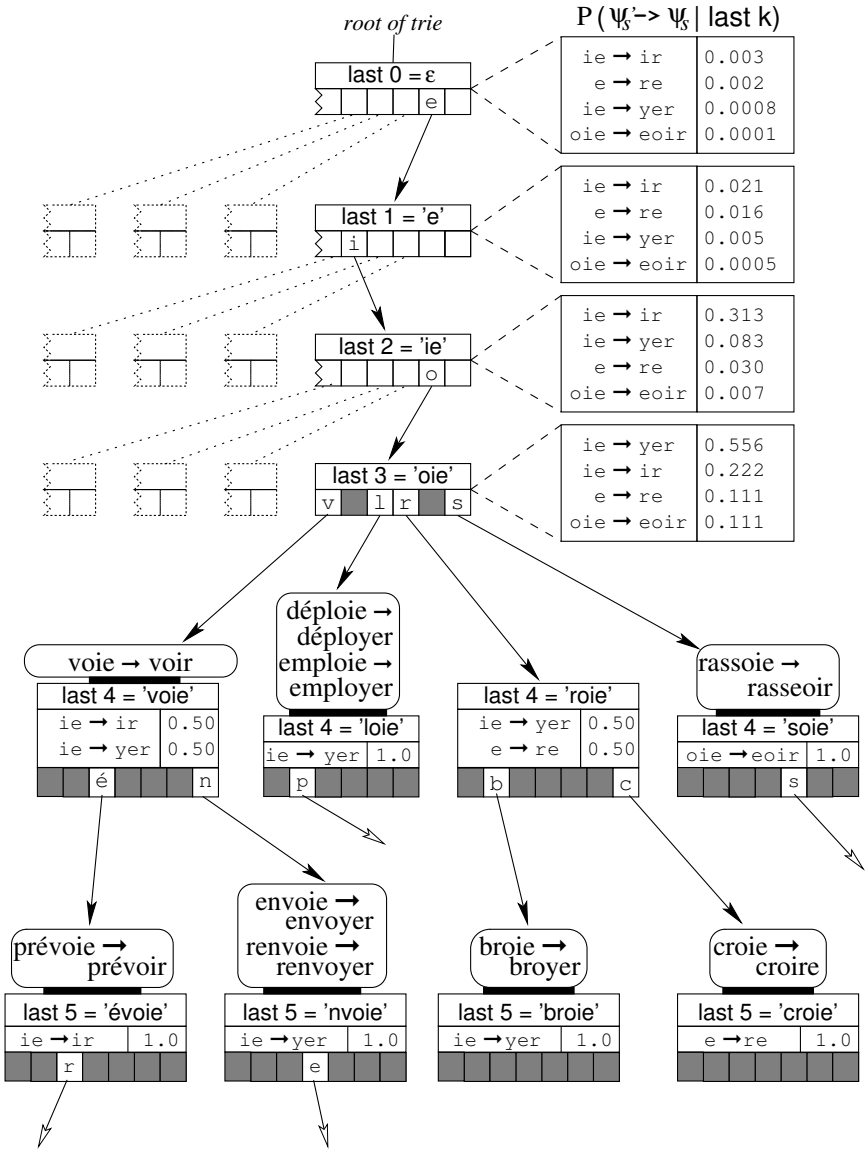


Figure 3.1: The trie data structure is used to compactly store probabilities from Eq. 3.3.

inflection	→	root	δ'_s	→	δ_s	C($\delta'_s \rightarrow \delta_s$)	language
closes	→	close	s	→	ϵ	1090	English
flies	→	fly	ies	→	y	51	English
closed	→	close	d	→	ϵ	435	English
abiatutako	→	abiatu	tako	→	ϵ	211	Basque
språkade	→	språka	de	→	ϵ	1498	Swedish
smältde	→	smälta	de	→	a	186	Swedish
parle	→	parler	ϵ	→	r	1477	French
parlent	→	parler	nt	→	r	1336	French
trek	→	trekken	ϵ	→	ken	33	Dutch
hangir	→	hanga	ir	→	a	258	Icelandic

Table 3.8: Training pairs exhibiting suffixation and point-of-suffixation changes.

change, $t \rightarrow d$, and a low frequency[8] suffix, *ecekmişsiniz*[9], along with the canonical ending *mek* creates a stem change pattern applicable to only one other inflection-root pair, *edecekmişsiniz* → *etmek*.

This naïve behavior is not always a disaster: sometimes poorly modeled transformations are effective at modeling numerous other forms. For example, the Dutch whole-word replacement pattern *getrokken* → *trekken*, although completely mis-modeling the linguistically plausible explanation of a prefix and a vowel shift, is able to explain 5 other inflection-root pairs as shown in Table 3.10. The *nearly complete* rewrite pattern *iennes* → *enir* found in French is able to model 21 other examples which are formed by prefixation from *venir*, as shown in Table 3.11.

Aside from its inability to capture internal changes and prefixation, the Base model has no capacity to model point-of-suffixation changes separately from the suffix. Table 3.12 illustrates this with an example from Estonian.

[8] *ecekmişsiniz* occurs without a point-of-suffixation change 26 times in training data
[9] This suffix is agglutinative as well, a phenonemon not directly handled by this model

inflection	→	root	δ'_s	→	δ_s	$C(\delta'_s \to \delta_s)$	language
viennes	→	venir	iennes	→	enir	22	French
getrokken	→	trekken	getrokken	→	trekken	6	Dutch
trok	→	trekken	ok	→	ekken	3	Dutch
pead	→	pidama	ead	→	idama	1	Estonian
muestren	→	mostrar	uestren	→	ostrar	1	Spanish
ninasongea	→	songea	ninasongea	→	songea	1	Swahili
makagasta	→	gasta	makagasta	→	gasta	1	Tagalog
gidecekmişsiniz	→	gitmek	decekmişsiniz	→	tmek	2	Turkish
gwewyf	→	gwau	ewyf	→	au	1	Welsh

Table 3.9: Training pairs exhibiting internal changes and prefixation, as well as highly irregular training pairs, generate patterns which often can not be applied to any other test examples.

inflection	→	root	δ'_s	→	δ_s
getrokken	→	**trekken**	**getrokken**	→	**trekken**
ingetrokken	→	intrekken	getrokken	→	trekken
voorgetrokken	→	voortrekken	getrokken	→	trekken
afgetrokken	→	aftrekken	getrokken	→	trekken
uitgetrokken	→	uittrekken	getrokken	→	trekken
aangetrokken	→	anntrekken	getrokken	→	trekken

Table 3.10: Dutch infl-root pairs with the stem-change pattern *getrokken* → *trekken*

abstiennes	adviennes	appartiennes	circonviennes	contiennes
contreviennes	conviennes	deviennes	détiennes	entretiennes
interviennes	maintiennes	obtiennes	parviennes	préviennes
retiennes	reviennes	soutiennes	subviennes	surviennes
	tiennes		**viennes**	

Table 3.11: French infl-root pairs with the stem-change pattern *iennes* → *enir*

Inflection $\gamma_s\delta'_s$	Root $\gamma_s\delta_s$	Proposed stem change $\delta'_s \rightarrow \delta_s$	Frequency $C(\delta'_s \rightarrow \delta_s)$	Suffix ψ'_s	Root ending ψ_s
pakuvad	pakkuma	uvad → kuma	5	vad	ma
pakutakse	pakkuma	utakse → kuma	4	takse	ma
pakutav	pakkuma	utav → kuma	4	tav	ma
pakume	pakkuma	ume → kuma	5	me	ma
pakutagu	pakkuma	utagu → kuma	4	tagu	ma
pakutud	pakkuma	utud → kuma	4	tud	ma
pakutaks	pakkuma	utaks → kuma	4	taks	ma
pakutama	pakkuma	utama → kuma	4	tama	ma
pakuta	pakkuma	uta → kuma	4	ta	ma
pakute	pakkuma	ute → kuma	5	te	ma
pakuti	pakkuma	uti → kuma	4	ti	ma
pakutavat	pakkuma	utavat → kuma	4	tavat	ma
pakub	pakkuma	ub → kuma	5	b	ma
pakud	pakkuma	ud → kuma	5	d	ma
pakun	pakkuma	un → kuma	5	n	ma
Total inflection-root training pairs exhibiting point-of-affixation u→ku: **71**					

Table 3.12: Inability to generalize point-of-affixation changes in the Base model is demonstrated with examples from Estonian. If these point-of-affixation changes were modeled properly, only one $\delta'_s \rightarrow \delta_s$ rule, u→ku, would be necessary to explain these 15 examples, together with a set of canonical suffixes which are also used for other inflections, and a single canonical ending, shown in the last two columns.

With robust training data, this is not problematic. However, this inability to accurately model the internal change separately causes problems with sparse training data. The two training pairs and single test inflection shown in Table 3.13 illustrate the issue. While training data is available for the suffix *taks*, and training data is available for the point-of-suffixation spelling change $u \rightarrow ku$, because the Base model concatenates the stem change and the suffix, there is no way to generalize from these examples to other suffixes exhibiting the same stem change. It is this inadequacy which serves as motivation for the Affix model presented in Section 3.4.

TRAINING PAIRS	Patterns generated
soovitaks → soovitama	taks → ma
pakutud → pakkuma	utud → kuma
TEST INFLECTION	Patterns applied
pakutaks → *pakutama**	taks → ma

Table 3.13: Modeling stem changes which occur across suffixes is difficult with sparse data in the Base model. The correct inflection should be *pakutaks* → *pakkutama*, but the Base model did not capture the *u* → *ku* generalization. from the single observed *utud* → *kuma* pattern.

3.3.3 Experimental Results on Base Model

Three sets of experiments are presented in this section. In the first, the root weight factor $\omega(root)$ (see (3.5)) was used as a filter such that proposed roots not found in a root list extracted from the evaluation data were eliminated.[10] This experiment (Table 3.15) shows the model's upper bound on performance (for a given training set) because all inflections have a root in the rootlist, and all roots are guaranteed to be the root of some inflection in the inflection list.

In the second experiment, $\omega(root)$ was again used as a filter, but here the proposed roots were matched against the union of the evaluation data rootlist and a larger set of potentially noisy roots extracted from various dictionaries.[11] No experiment was run with $\omega(root)$ being used as a filter matched against only the dictionary since this would essentially be a test of the dictionary's coverage, and not the model's performance.

In the final experiment, $\omega(root)$ was set to the constant value 1 for all proposed roots.[12] This experiment is testing only the performance of the trie model since the weight-

[10] To do this, $\omega(root) = 1.0$ for roots found in the training data, 0.0 otherwise.
[11] As above, $\omega(root) = 1.0$ for roots in the training data rootlist or in a dictionary, 0.0 otherwise.
[12] This is equivalent to saying that the rootlist is empty and that all proposed roots are equally bad.

Language	$\omega(root)$ from evaluation data	$\omega(root)$ from evaluation data $\cup$ dictionary	$\omega(root) = 1$ for all roots
Spanish	94.62%	93.97%	89.81%
Portuguese	97.35%	97.18%	92.55%
Catalan	84.53%	-	71.59%
Occitan	89.58%	-	83.56%
French	99.04%	99.02%	95.83%
Italian	98.06%	98.01%	93.52%
Romanian	96.96%	96.93%	92.20%
Latin	88.47%	-	78.04%
English	98.33%	97.57%	90.98%
Danish	96.56%	95.82%	89.32%
Norwegian	93.71%	-	82.45%
Swedish	97.77%	97.41%	94.97%
Icelandic	84.15%	-	74.21%
Hindi	84.77%	-	80.47%
Sanskrit	87.75%	-	81.03%
Estonian	82.81%	82.55%	63.76%
Tamil	90.95%	-	79.23%
Finnish	97.35%	-	88.22%
Turkish	99.36%	98.71%	89.49%
Uzbek	99.44%	-	95.91%
Basque	94.54%	94.39%	85.28%
Czech	78.70%	78.62%	72.27%
Polish	97.20%	97.04%	93.27%
Russian	85.84%	84.07%	77.92%
Greek	15.62%	15.62%	14.06%
German	92.04%	91.91%	87.73%
Dutch	86.44%	86.08%	79.74%
Irish	43.87%	-	43.27%
Welsh	87.58%	-	69.05%
Tagalog	0.76%	-	0.34%
Swahili	2.94%	2.94%	2.93%
Klingon	0.00%	0.00%	0.00%

Table 3.14: Effect of the weight factor $\omega(root)$ on accuracy. Appendix A contains information on the size of the dictionary available for each language.

Language	All Words	Regular	Semi-Regular	Irregular	Obsolete/ Other
Spanish	94.62%	95.95%	89.31%	78.76%	93.79%
Portuguese	97.35%	97.45%	-	83.33%	30.00%
Catalan	84.53%	92.68%	81.30%	60.13%	-
Occitan	89.58%	96.92%	78.75%	45.66%	18.42%
French	99.04%	99.61%	97.97%	92.52%	81.46%
Italian	98.06%	98.55%	99.43%	91.10%	98.20%
Romanian	96.96%	98.83%	86.96%	79.76%	85.79%
Latin	88.47%	94.88%	60.28%	62.76%	69.47%
English	98.33%	98.99%	98.66%	28.12%	100.00%
Danish	96.56%	97.69%	92.27%	80.49%	-
Norwegian	93.71%	96.53%	90.62%	57.48%	-
Swedish	97.77%	-	-	-	-
Icelandic	84.15%	96.71%	61.95%	30.58%	-
Hindi	84.77%	98.58%	33.33%	14.29%	-
Sanskrit	87.75%	-	-	-	-
Estonian	82.81%	-	-	-	-
Tamil	90.95%	93.55%	81.82%	-	-
Finnish	97.35%	98.72%	-	92.93%	96.27%
Turkish	99.36%	99.91%	95.02%	88.17%	-
Uzbek	99.44%	-	-	-	-
Basque	94.54%	-	-	-	-
Czech	78.70%	-	-	-	-
Polish	97.20%	-	-	-	-
Russian	85.84%	-	-	-	-
Greek	15.62%	-	-	-	-
German	92.04%	93.36%	97.54%	81.88%	90.43%
Dutch	86.44%	85.24%	95.20%	69.70%	-
Irish	43.87%	95.35%	20.48%	0.00%	-
Welsh	87.58%	88.32%	86.02%	29.29%	100.00%
Tagalog	0.76%	-	-	-	-
Swahili	2.94%	2.94%	-	0.00%	-
Klingon	0.00%	0.00%	-	-	-

Table 3.15: Performance of Base model when $\omega(root)$ is used as a filter to eliminate roots not found in the evaluation data. Appendix A contains information on the amount of evaluation data available for each language.

ing factor is irrelevant.

When available, the results are sub-divided between the performance on regular, semi-regular, irregular and obsolete inflections. These distinctions were assigned to each inflection-root pair by a third party not affiliated with this research. Although spot checking has revealed some inconsistencies in these classifications, no corrections or alterations of the classification labels have been done.

All of the experiments on the supervised models, unless otherwise specified, were performed by using 10-fold cross-validation on the evaluation data, which was a list of aligned inflection-root pairs. No frequency was attached to the words, and since part-of-speech taggers were not available for most languages investigated, these frequencies could not be accurately extracted from a corpus.[13] Therefore, all results, unless otherwise specified, are based on type accuracy, not token accuracy.

Table 3.14 shows the performance of this model for each of the three experiments. When expanding the rootlist to include words found in a dictionary, the accuracy is nearly as good as using the ideal evaluation data as the source of $\omega(root)$. However, the decrease when setting $\omega(root)$ equal to 1 for all proposed roots causes large dropoffs in performance for every language.[14]

Table 3.15 shows the performance of the analyzer on different classes of inflections: regular, semi-regular, irregular, and obsolete/other. Performance drops sharply from the regulars to the semi-regulars, and then further on the irregulars. However, the average

[13]Without a part of speech tagger, it would not be possible to determine whether a word was an inflected verb. For example, in English, many verbs are used directly as nouns. Counting all matching word forms as inflections would not be a valid indicator of token frequency.

[14]With the exception of Tagalog, Swahili, and Klingon, which didn't have much room to drop.

performance on the regular inflections (excluding Swahili and Klingon) is only 95.9%, with many results well below 99%, even for those languages for which the concatenative suffix model would seem to be most effective. This is largely due to the Base model's inability to model point-of-affixation changes efficiently or well, as will be shown in Section 3.4.

Tagalog, Swahili and Klingon, which make heavy use of prefixation, are completely mis-modeled by the Base and Affix models which support only suffixation.

3.4 The Affix model

The Base morphological similarity measure (Section 3.3) models suffixation and point-of-affixation changes as a single combined transformation, $\delta'_s \rightarrow \delta_s$. As seen in Table 3.12, this is not able to generalize point-of-affixation changes that are shared across many inflections of the same root. The Affix model not only addresses this shortcoming by separating the representation of the stem change from that of the suffix, but it also allows for limited handling of purely concatenative prefixation.

3.4.1 Model Formulation

The morphological distance measure used in the Affix model remains $P(\delta'_s \rightarrow \delta_s, \pi|\gamma_s\delta'_s)$ where π is the part of speech. However, instead of modeling the probability of the combined point-of-suffixation change/suffix as the Base model does, the Affix model presented here represents the suffixation and end-of-string changes as separate processes. In addition, this model handles limited concatenative prefixation. To do this, the inflection and root are represented as $\psi'_p\gamma_s\delta'_s\psi'_s$ and $\gamma_s\delta_s\psi_s$, respectively, where ψ'_p is a prefix, ψ'_s is a

			prefix			suffix			ending	point-of-suffix. change		
inflection	→	root	ψ'_p	γ_s	δ'_s	ψ'_s	γ_s	δ_s	ψ_s	δ'_s	→	δ_s
hopping	→	hop	ϵ	hop	p	ing	hop	ϵ	ϵ	p	→	ϵ
hopped	→	hop	ϵ	hop	p	ed	hop	ϵ	ϵ	p	→	ϵ
abatte	→	abattre	ϵ	abatt	ϵ	e	abatt	ϵ	re	ϵ	→	ϵ
chante	→	chanter	ϵ	chant	ϵ	e	chant	ϵ	er	ϵ	→	ϵ
gefallen	→	fallen	ge	fall	ϵ	en	fall	ϵ	en	ϵ	→	ϵ

Table 3.16: Examples of inflection-root pairs as analyzed by the Affix model.

suffix, ψ_s is an ending, and, as before, δ'_s and δ_s represent the point-of-suffixation change with γ_s as the remaining similar substring. (See Table 3.16 for some examples of this framework.)

As described, this model does not allow for point-of-*prefixation* changes. This has the consequence that, while simple prefixation can be accommodated, it will only be modeled correctly if the prefixation causes no point-of-prefixation change. In addition, prefixation is only allowed if the prefix being added is provided in the prefix list. The WFBase model presented Section 3.5 will address both of these problems by handling prefixation and point-of-prefixation changes without requiring a prefix list.

The $\delta'_s \rightarrow \delta_s$ stem changes are stored in the trie based not on the full inflection, $\psi'_p\gamma_s\delta'_s\psi'_s$, but based on the inflection with the prefix and suffix removed, $\gamma_s\delta'_s$, so the formula (3.2) can still be used. This combination of separating the stem change from the affix and storing these stem change probabilities in the trie independent of these affixes will allow modeling of the end-of-string changes across all of the inflections of a single root.

For (3.2) to apply, the prefix ψ'_p, suffix ψ'_s, and ending ψ_s must be split off. If there are multiple ways to do this (as in Table 3.18), the probabilities of these splits must

be summed. This issue is fudged by making approximations and strong independence assumptions:

$$
\begin{aligned}
&P(\text{root, POS} \mid \text{inflection}) \\
&= P(\gamma_s\delta_s\psi_s, \pi | \psi'_p\gamma_s\delta'_s\psi'_s) \\
&= \sum_{\psi'_p,\gamma_s\delta'_s,\psi'_s} P(\gamma_s\delta_s\psi_s, \pi | \psi'_p, \gamma_s\delta'_s, \psi'_s) \cdot P(\psi'_p, \gamma_s\delta'_s, \psi'_s | \psi'_p\gamma_s\delta'_s\psi'_s) \\
&\approx \max_{\psi'_p,\gamma_s\delta'_s,\psi'_s} P(\gamma_s\delta_s\psi_s, \pi | \psi'_p, \gamma_s\delta'_s, \psi'_s) \cdot P(\psi'_p, \gamma_s\delta'_s, \psi'_s | \psi'_p\gamma_s\delta'_s\psi'_s) \\
&= \max_{\psi'_p,\gamma_s\delta'_s,\psi'_s} P(\psi_s | \gamma_s\delta_s, \psi'_p, \gamma_s\delta'_s, \psi'_s, \pi) \cdot P(\gamma_s\delta_s, \pi | \psi'_p, \gamma_s\delta'_s, \psi'_s) \cdot P(\psi'_p, \gamma_s\delta'_s, \psi'_s | \psi'_p\gamma_s\delta'_s\psi'_s)
\end{aligned}
$$

Making the independence assumptions that all possible splits are equally likely,

as are all canonical endings.

$$
\begin{aligned}
&= \max_{\psi'_p,\gamma_s\delta'_s,\psi'_s} \frac{1}{\#\text{endings}} \cdot P(\gamma_s\delta_s, \pi | \gamma_s\delta'_s) \cdot \frac{1}{\#\text{splits}} \\
&= \max_{\psi'_p,\gamma_s\delta'_s,\psi'_s} \frac{1}{\#\text{endings}} \cdot P(\delta'_s \rightarrow \delta_s, \pi | \gamma_s\delta'_s) \cdot \frac{1}{\#\text{splits}} \qquad \text{(See Section 3.3.1 for justification)} \\
&= \text{constant} \cdot P(\delta'_s \rightarrow \delta_s, \pi | \gamma_s\delta'_s)
\end{aligned}
\tag{3.6}
$$

The sum of the splits is approximately equal to the max over all splits because one summand is usually much more likely than the others. All splits are assumed to be equally likely, and all canonical endings, ψ_s, are equally likely, regardless of context. This means that the correct $\delta'_s \rightarrow \delta_s$ rule only depends on $\gamma_s\delta'_s$ once $\gamma_s\delta'_s$ is known, as was true in the presentation of the Base model (Section 3.3.1).

The weighting factor, $\omega(root)$, is used identically in this model as in the Base model.

3.4.2 Additional resources required by the Affix model

The sets Ψ'_p, Ψ'_s, and Ψ_s contain, respectively, the canonical prefixes, suffixes and root endings for the part-of-speech being analyzed. These sets must be hand-entered or automatically acquired by other means, since this model does not create these sets directly from training data. However, as these lists are often small and readily available from grammar reference books, doing so is straightforward.

The notion of a set of standard endings for roots of a particular part of speech (such as nouns or verbs),[15] while not language-universal, is well established across a broad range of language families (Table 3.17). Such canonical root endings can be found in, but are not limited to, Greek, Germanic languages such as German and Dutch, Turkic languages such as Turkish and Uzbek, Romance languages such as French, and Portuguese, Slavic languages such as Czech and Polish, and Italic languages such as Latin. When found in a language, these sets of root endings for different parts of speech should not only constitute the common endings of roots, but also an exhaustive set of all the possible endings for each part of speech.[16]

To indicate that sets of endings generally are exhaustive sets determined by the affixation rules of the language, these sets will be referred to as the sets of *canonical endings*, and each member of such a set as a *canonical ending*.

[15] First introduced in Section 3.1.2.

[16] Although not done here, one could accept a probability distribution over these canonical endings which could be used in (3.6).

Language Family	Language	Example Endings
Greek	Greek	$-\omega$
Germanic	German	$-$en
Germanic	Dutch	$-$en
Turkic	Turkish	$-$mak, $-$mek
Turkic	Uzbek	$-$moq
Romance	French	$-$er, $-$ir, $-$re
Romance	Portuguese	$-$ar, $-$ir, $-$or, $-$er
Slavic	Czech	$-$at, $-$it, $-$nout, $-$ovat
Slavic	Polish	$-$ać, $-$eć
Italic	Latin	$-$o

Table 3.17: Endings for root verbs found across language families

While the set of canonical root endings is usually short and easily enumerated, the members of the sets of canonical prefixes and suffixes are often much harder to exhaustively enumerate. Fortunately, the sets of canonical prefixes and suffix provided to the Affix model need not be complete, so long as $\epsilon \in \Psi'_s$, because that always gives the Affix model a fallback position. For every necessary prefix that is not included in the prefix set, words with this null prefixation will be modeled as they were in Base model, using whole word replacement. For every necessary suffix that is not included in the suffix set, inflections with these suffixes will have the inflectional stem changes and suffixes grouped together as in the Base model. In fact, the previously presented Base model is simply a special case of this Affix model where $\Psi'_p = \Psi'_s = \Psi_s = \{\epsilon\}$.

3.4.3 Analysis of Training Data

In the Base model, the analysis of the inflection and root into $\gamma_s\delta'_s$ and $\gamma_s\delta_s$ was straightforward: γ_s was the longest common prefix between the inflection and root. In the

SUFFIX LIST	PROPOSED ANALYSES OF TRAINING DATA									
	ψ'_p	γ_s	δ'_s	ψ'_s	γ_s	δ_s	ψ_s	δ'_s	$\rightarrow$	δ_s
erait	ϵ	aim	ϵ	erait	aim	ϵ	er	ϵ	$\rightarrow$	ϵ
rait	ϵ	aim	e	rait	aim	ϵ	er	e	$\rightarrow$	ϵ
ait	ϵ	aim	er	ait	aim	ϵ	er	er	$\rightarrow$	ϵ
it	ϵ	aim	era	it	aim	ϵ	er	era	$\rightarrow$	ϵ
t	ϵ	aim	erai	t	aim	ϵ	er	erai	$\rightarrow$	ϵ
ϵ	ϵ	aim	erait	ϵ	aim	ϵ	er	erait	$\rightarrow$	ϵ

Table 3.18: Competing analyses of French training data in the Affix model with $\Psi'_p = \{\epsilon\}, \Psi'_s = \{$erait, rait, ait, it, t, $\epsilon\}$, and $\Psi_s = \{$er, ir, re$\}$. In this example, Ψ'_s is a subset of the true set of suffixes for French (see Table 3.19). Generally, $\epsilon \notin \Psi_s$ unless $\Psi_s = \{\epsilon\}$.

Suffix	Inflection	$\rightarrow$	Root	Part-of-Speech
-erait	aimerait	$\rightarrow$	aimer	3rd Singular Present Conditional
-rait	entendrait	$\rightarrow$	entendre	3rd Singular Present Conditional
-ait	aimait	$\rightarrow$	aimer	3rd Singular Imperfect Indicative
-it	entendit	$\rightarrow$	entendre	3rd Singular Simple Past
-t	finit	$\rightarrow$	finir	3rd Singular Present Indicative
ϵ	va	$\rightarrow$	aller	3rd Singular Present Indicative

Table 3.19: Examples of the suffixes presented in Table 3.18. The suffix ϵ is most often associated with irregular forms such as *va*$\rightarrow$*aller* above.

Affix model, the analysis of a training pair into the pair $(\psi'_p\gamma_s\delta'_s\psi'_s, \gamma_s\delta_s\psi_s)$ is less obvious. For some training pairs, the sets Ψ'_p, Ψ'_s, and Ψ_s provide multiple possible analyses. For example, given the training pair *aimerait* $\rightarrow$ *aimer*, a subset of the French affixes, and the canonical ending *er*, Table 3.18 shows the competing analyses.

For training purposes, the affix sets are treated as ranked lists such that the highest ranking combined (prefix, suffix, ending) triple which analyzes the training data is accepted with ties broken in favor of the ending first, and the suffix second. In all of the experiments presented, the affix lists were ordered by length, such that longer matching affixes (both

prefixes and suffixes) were preferred over shorter ones.[17] In this way, the competing analyses shown in Table 3.18 can be ranked, and the pattern selected is $\epsilon \rightarrow \epsilon$.

Unfortunately, this method of choosing the longest matching affix does not always work properly. An example of this is with regular French verbs which have the canonical ending *er*, but more specifically, happen to end in with *rer*, such as *abjurer*. The 3rd person singular imperfect indicative of *abjurer* is *abjurait*. Using the same set of prefixes as from Table 3.18, and using the longest-matching-affix method, the analysis of *abjurait* would be to remove the suffix *rait*, with a resulting point-of-suffixation change $\epsilon \rightarrow r$. The correct analysis should be to remove the suffix *ait* with a resulting $\epsilon \rightarrow \epsilon$ change.

3.4.4 Model Effectiveness

As observed in Table 3.20, this affix based representation allows for modeling a change, such as $f \rightarrow v$, separately from the suffix. This is compared with the previous representation where the stem change pattern combined the suffix and the point-of-affixation change (Table 3.21). Recomputing the stem changes for Estonian originally presented in Table 3.12 using the Affix model yields much more robust stem change statistics, as shown in Table 3.22. The repeated stem change $u \rightarrow ku$[18] is correctly observed across the inflections of the root *pakkuma*.

This model is not effective at modeling deeper internal stem changes. due to suffixation. Often, these changes are orthographic vowel changes representative of underlying

[17] There are three other obvious ways to handle this planned for future work. The first is to allow a human to provide this ranking, the second is to choose the affix which results in the simplest stem change, and the third is to give each analysis partial weight in the stem change counts.

[18] $k \rightarrow kk$ would be a more acceptable linguistic explanation of this phenomenon, but $u \rightarrow ku$ is effective here since it is stored in the trie with the contextual history indicating that it is a preferred stem change only when following the letter k.

inflection	→	root	ψ'_p	γ_s	δ'_s	ψ'_s	γ_s	δ_s	ψ_s	δ'_s	→	δ_s
annoteer	→	annoteren	ϵ	annote	er	ϵ	annote	r	en	er	→	r
annoteert	→	annoteren	ϵ	annote	er	t	annote	r	en	er	→	r
annoterend	→	annoteren	ϵ	annoter	ϵ	end	annoter	ϵ	en	ϵ	→	ϵ
schrijf	→	schrijven	ϵ	schrij	f	ϵ	schrij	v	en	f	→	v
schrijft	→	schrijven	ϵ	schrij	f	t	schrij	v	en	f	→	v
schrijvend	→	schrijven	ϵ	schrijv	ϵ	end	schrijv	ϵ	en	ϵ	→	ϵ

Table 3.20: Morphological processes in Dutch as modeled by Affix model

			BASE MODEL			AFFIX MODEL					
INFLECTION	→	ROOT	δ'_s	→	δ_s	ψ'_p	ψ'_s	δ'_s	→	δ_s	ψ_s
annoteer	→	annoteren	er	→	ren	ϵ	ϵ	er	→	r	en
annoteert	→	annoteren	ert	→	ren	ϵ	t	er	→	r	en
annoterend	→	annoteren	d	→	ϵ	ϵ	end	ϵ	→	ϵ	en
schrijf	→	schrijven	f	→	ven	ϵ	ϵ	f	→	v	en
schrijft	→	schrijven	ft	→	ven	ϵ	t	f	→	v	en
schrijvend	→	schrijven	d	→	ϵ	ϵ	end	ϵ	→	ϵ	en

Table 3.21: Comparison of the representations by the Base model and Affix model on the Dutch example from Table 3.20

inflection $\psi'_p\gamma_s\delta'_s\psi'_s$	root $\gamma_s\delta_s\psi_s$	prefix ψ'_p	suffix ψ'_s	stem change δ'_s	$\rightarrow$	δ_s	ending ψ_s	C($\delta'_s \rightarrow \delta_s$)	C(ψ'_s)
pakuvad	pakkuma	ϵ	vad	u	$\rightarrow$	ku	ma	71	144
pakutakse	pakkuma	ϵ	takse	u	$\rightarrow$	ku	ma	71	109
pakutav	pakkuma	ϵ	tav	u	$\rightarrow$	ku	ma	71	109
pakume	pakkuma	ϵ	me	u	$\rightarrow$	ku	ma	71	437
pakutagu	pakkuma	ϵ	tagu	u	$\rightarrow$	ku	ma	71	111
pakutud	pakkuma	ϵ	tud	u	$\rightarrow$	ku	ma	71	107
pakutaks	pakkuma	ϵ	taks	u	$\rightarrow$	ku	ma	71	114
pakutama	pakkuma	ϵ	tama	u	$\rightarrow$	ku	ma	71	109
pakuta	pakkuma	ϵ	ta	u	$\rightarrow$	ku	ma	71	264
pakute	pakkuma	ϵ	te	u	$\rightarrow$	ku	ma	71	437
pakuti	pakkuma	ϵ	ti	u	$\rightarrow$	ku	ma	71	108
pakutavat	pakkuma	ϵ	tavat	u	$\rightarrow$	ku	ma	71	111
pakub	pakkuma	ϵ	b	u	$\rightarrow$	ku	ma	71	145
pakud	pakkuma	ϵ	d	u	$\rightarrow$	ku	ma	71	459
pakun	pakkuma	ϵ	n	u	$\rightarrow$	ku	ma	71	437

Table 3.22: Point-of-affixation stem changes in Affix model (examples from Estonian)

phonological vowel shifts. Table 3.23 shows examples of the Spanish vowel shift $ue \rightarrow o$. The examples listed each have a different analysis under the Affix model. A simpler description of this process could model this shift separately from the end-of-string changes, prefixes, suffixes and canonical endings. The Wordframe model (Section 3.5) addresses this problem.

In addition to having to provide sets of affixes and canonical endings for the Affix model, there is one another major drawback of the implementation of the Affix model relative to the Base model: the Affix model does not conditionalize the stem change on the suffix, and does not conditionalize the canonical ending on the suffix, or on the changed stem.

Illustrating the first point, the Affix model's representation of the English past

INFLECTION	→	ROOT	ψ'_p	ψ'_s	δ'_s	→	δ_s	ψ_s
acuerto	→	acortar	ϵ	o	uert	→	ort	ar
acuesto	→	acostar	ϵ	o	uest	→	ost	ar
almuerzo	→	almorzar	ϵ	o	uerz	→	orz	ar
aluengo	→	alongar	ϵ	o	ueng	→	ong	ar
apruebo	→	aprobar	ϵ	o	ueb	→	ob	ar
cuelgo	→	colgar	ϵ	o	uelg	→	olg	ar
concuerdo	→	concordar	ϵ	o	uerd	→	ord	ar
conmuevo	→	conmover	ϵ	o	uev	→	ov	er
consuelo	→	consolar	ϵ	o	uel	→	ol	ar
cuento	→	contar	ϵ	o	uent	→	ont	ar
desenvuelvo	→	desenvolver	ϵ	o	uelv	→	olv	er
duermo	→	dormir	ϵ	o	uerm	→	om	ir
muero	→	morir	ϵ	o	uer	→	or	ir
muestro	→	mostrar	ϵ	o	uestr	→	ostr	ar
ruedo	→	rodar	ϵ	o	ued	→	od	ar
ruego	→	rogar	ϵ	o	ueg	→	og	ar
suelto	→	soltar	ϵ	o	uelt	→	olt	ar
trueno	→	tronar	ϵ	o	uen	→	on	ar
vuelco	→	volcar	ϵ	o	uelc	→	olc	ar

Table 3.23: Inability of Affix model to handle internal vowel shifts.

tense training pair *smiled* → *smile* is a stem change pattern $\epsilon \rightarrow e$ with suffix *+ed*. This is a reasonable pattern for English inflections ending in *+ed* and *+ing*, but not for inflections ending in *+s*. While this rule is common for inflections ending in *+ed* (*smiled* → *smile*) and for inflections ending in *+ing* (*smiling* → *smile*), it is extremely uncommon for inflections ending in *+s* (*smiles* → *smile*) to require that an *+e* be replaced at the end of the root.[19] Such a rule, applied to *smiles*, would yield the incorrect root *smilee**.

An illustration of the second point can be found in almost any language for which the canonical ending of the verb defines a paradigm of inflectional suffixes which are at

[19]The only English roots for which the final *+e* of the root is dropped when creating the 3rd person singular present tense inflection are *to have (has)* and *to be (is)*, hardly among the most regular verbs in the language.

least partially unique to that canonical ending. For example, the French suffix *âmes* only attaches to verbs ending in *er*. Since the probability $P(\psi'_s|\psi_s)$ is not part of the statistical model, inflections ending with *âmes* are as likely to generate roots ending in *ir* or *re* as ending in the correct paradigm *er*.

This failure to model the conditional probability $P(\psi'_s|\psi_s)$ does not affect the Base model because the Base model represents the end of string stem change and suffix as a single string transformation $\delta'_s \rightarrow \delta_s$. which is approximately equivalent to $\delta'_s\psi'_s \rightarrow \delta_s\psi_s$ in the Affix model.[20] From this, one observes that the Base model is approximating $P(\delta'_s\psi'_s \rightarrow \delta_s\psi_s|\psi'_p\gamma_s\delta'_s\psi'_s)$, which is, for any given split, conditionalizing the addition of the canonical ending, ψ_s, on the inflection ending in ψ'_s.

The key point being made here is that for simple phenomena, the Base model may be a more effective representation of the analysis than the more complex Affix model. On the other hand, the Affix model is able handle more complex phenomena that the Base model cannot effectively model. This will serve as the basis for Section 3.6.2, where the supervised models are combined to achieve accuracies higher than the stand-alone models.

3.4.5 Performance of the Affix Model

Of the thirty-two languages for which results were presented on the Base model, only eleven are evaluated here with canonical prefix, suffix, and ending lists. Of the remaining languages, an additional ten languages are evaluated with canonical root endings lists but empty suffix and prefix lists. This deficiency is due to the difficulty in obtaining clean

[20]This is only approximately equivalent. In French, for example, *chante*→*chanter* would be represented as $\epsilon \rightarrow r$ by the Base model, but as *e*→*er* by the Affix model.

lists of these types for all of the languages. This difficulty naturally points to a need for a system which can perform without such lists, or for a system which can provide such lists automatically.[21]

As with the Base model, three sets of experiments showing the performance of this model including the weight factor $\omega(root)$ are presented. In addition, a fourth experiment is included which shows the performance of this model when the canonical endings are provided but suffix and prefix lists are not.

As with the Base model, all of the experiments in this section were performed by doing 10-fold cross-validation on the evaluation data.

In 11 of the 14 languages for which results are presented (Table 3.25), the Affix model was an improvement over the Base model. The reason for the increased accuracy (or lack of increased accuracy) was, in all cases, the difference in coverage.[22] In every case, chosing the model with the larger coverage yielded the highest accuracy.

In Tagalog, Swahili and Klingon the increase in coverage and accuracy was huge. This is a direct result of the Affix model's ability to handle simple concatenative prefixation, something that the Base model could not do.

The performance loss incurred when increasing the rootlist to include a broad-coverage dictionary was, as in the Base model, less than 1% for all languages (Table 3.26).

Table 3.27 presents results when including prefixes, suffixes and canonical endings, as well as results when including only canonical endings. For 6 of the 7 languages, when prefix and suffix lists were available, they were an improvement over using only the canonical

[21] Such as [Goldsmith, 2001]

[22] Coverage indicates the percentage of test examples for which an analysis existed whose MorphSim score was above a preset threshold.

Language	All Words	Regular	Semi-Regular	Irregular	Obsolete/ Other
Spanish	96.48%	96.90%	94.95%	90.99%	100.00%
French	99.32%	99.59%	99.64%	95.96%	88.08%
Italian	98.12%	98.58%	99.55%	91.55%	98.20%
English	98.73%	99.32%	99.16%	32.26%	100.00%
Danish	94.86%	95.61%	94.55%	81.46%	-
Swedish	97.35%	-	-	-	-
Estonian	96.22%	-	-	-	-
Basque	94.03%	-	-	-	-
Czech	98.15%	-	-	-	-
Greek	99.48%	-	-	-	-
Dutch	93.76%	92.89%	98.76%	85.71%	-
Tagalog	91.77%	-	-	-	-
Swahili	93.84%	93.84%	-	100.00%	-
Klingon	100.00%	100.00%	-	-	-

Table 3.24: Performance on regular, semi-regular and irregular inflectional morphology by the Affix model. Empty cells indicate where classifications were not available or not present.

endings. Comparing only using canonical endings versus not using any affixes (the Base model), canonical endings helped as much as they hurt: for 8 languages, accuracy improved, for another 8 language, accuracy decreased[23].

3.5 Wordframe models: WFBase and WFAffix

The Wordframe models handle two issues left unresolved by the Affix model: the inability to model internal vowel shifts efficiently, and the inability to model prefixation without a list of provided prefixes. The Wordframe (WF) model is built upon the trie-based architecture of either the Base model or the Affix model, yielding the Wordframe Base model (WFBase) and the Wordframe Affix model (WFAffix). The strengths and

[23]Excluding Swahili.

Language	BASE MODEL			Acc. Increase	AFFIX MODEL		
	Acc.	Covg.	Prec.		Acc.	Covg.	Prec.
Spanish	94.62%	94.66%	99.95%	1.97%	**96.48%**	97.02%	99.44%
French	99.04%	99.12%	99.92%	0.28%	**99.32%**	99.72%	99.60%
Italian	98.06%	98.11%	99.95%	0.06%	**98.12%**	98.17%	99.95%
English	98.43%	98.46%	99.97%	0.30%	**98.73%**	98.76%	99.97%
Danish	**96.56%**	96.70%	99.85%	-1.76%	94.86%	95.01%	99.85%
Swedish	**97.77%**	98.16%	99.60%	-0.43%	97.35%	98.03%	99.31%
Estonian	82.81%	84.00%	98.58%	16.21%	**96.22%**	96.86%	99.34%
Basque	**94.54%**	94.59%	99.95%	-0.54%	94.03%	94.20%	99.82%
Czech	78.70%	78.85%	99.81%	24.72%	**98.15%**	98.80%	99.34%
Greek	15.62%	15.62%	100.00%	536.67%	**99.48%**	99.48%	100.00%
Dutch	86.44%	86.46%	99.98%	8.48%	**93.76%**	94.45%	99.27%
Tagalog	0.76%	0.98%	78.02%	11921.77%	**91.77%**	93.42%	98.24%
Swahili	2.94%	3.64%	80.63%	3096.42%	**93.84%**	94.05%	99.78%
Klingon	0.00%	1.35%	0.00%	n/a	**100.00%**	100.00%	100.00%

Table 3.25: Performance of Base model vs. Affix model for languages with full affix lists when $\omega(root)$ is used as a filter to eliminate roots not found in the evaluation data

Language	BASE MODEL			AFFIX MODEL		
	$\omega(root)$ from eval.data	$\omega(root)$ from eval.data + dict.	$\omega(root)$ from no rootlist	$\omega(root)$ from eval.data	$\omega(root)$ from eval.data + dict.	$\omega(root)$ from no rootlist
Spanish	94.62%	93.97%	89.81%	96.48%	96.12%	89.34%
French	99.04%	99.02%	95.83%	99.32%	99.23%	91.90%
Italian	98.06%	98.01%	93.52%	98.12%	98.08%	93.58%
English	98.33%	97.57%	90.98%	98.62%	98.00%	94.68%
Danish	96.56%	95.82%	89.32%	94.86%	94.31%	78.04%
Swedish	97.77%	97.41%	94.97%	97.35%	96.75%	87.51%
Basque	94.54%	94.39%	85.28%	94.03%	93.87%	81.24%
Czech	78.70%	78.62%	72.27%	98.13%	97.44%	85.09%
Dutch	86.44%	86.08%	79.74%	93.76%	92.76%	74.23%
Tagalog	0.76%	-	0.34%	91.77%	-	80.34%
Swahili	2.94%	2.94%	2.93%	93.84%	93.76%	75.77%
Klingon	0.00%	0.00%	0.00%	100.0%	100.0%	100.0%

Table 3.26: Effect of the weight factor $\omega(root)$ on accuracy in the Base and Affix models

Language	BASE MODEL	AFFIX MODEL ENDINGS ONLY	AFFIX MODEL AFFIX+ENDINGS
Spanish	94.62%	94.43%	**96.48%**
Portuguese	**97.35%**	97.33%	-
Catalan	84.53%	**84.73%**	-
Occitan	**89.58%**	88.74%	-
French	99.04%	99.08%	**99.32%**
Italian	98.06%	**98.12%**	-
Romanian	**96.96%**	95.73%	-
English	98.33%	-	**98.62%**
Danish	**96.56%**	-	94.86%
Swedish	97.77%	**97.80%**	97.35%
Icelandic	**84.15%**	84.02%	-
Estonian	82.81%	**82.96%**	-
Finnish	**97.35%**	**97.35%**	-
Turkish	99.36%	**99.41%**	-
Uzbek	**99.44%**	**99.44%**	-
Basque	**94.54%**	93.19%	94.03%
Czech	78.70%	78.72%	**98.15%**
Polish	**97.20%**	97.19%	-
Greek	15.62%	-	**99.48%**
German	**92.04%**	91.87%	-
Dutch	86.44%	86.48%	**93.76%**
Tagalog	0.76%	-	**91.77%**
Swahili	2.94%	2.93%	**93.84%**
Klingon	0.00%	-	**100.00%**

Table 3.27: Performance differences when using no affixes (i.e. the Base model), using only canonical root endings, and when using canonical root endings and affixes

weaknesses of each of the underlying suffix models will be preserved in the created WF models. Each of the underlying models (Base and Affix) is evaluated separately for use as the foundation of the WF model.

3.5.1 Wordframe Model Formulation

The Wordframe model is an extension of the previously presented suffix models. This extension can be added to either the Base model or the Affix model, so the underlying morphological distance measure used remains based on $P(\delta'_s \rightarrow \delta_s, \pi | \gamma_s \delta'_s)$.

The WF model built upon the Affix model represents the transformation from inflection to root as

$$\psi'_p \delta'_p \gamma_p \delta'_v \gamma_s \delta'_s \psi'_s \rightarrow \delta_p \gamma_p \delta_v \gamma_s \delta_s \psi_s \tag{3.7}$$

which may be more easily pictured as

WORDFRAME MODEL

	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection	ψ'_p	δ'_p	γ_p	δ'_v	γ_s	δ'_s	ψ'_s
root		δ_p		δ_v		δ_s	ψ_s

In this formulation, ψ'_p, ψ'_s and ψ_s remain the prefix, suffix, and canonical root ending as before δ'_p and δ_p form the point-of-*prefixation* stem change $\delta'_p \rightarrow \delta_p$, and the point-of-*suffixation* stem change is represented $\delta'_s \rightarrow \delta_s$. Previously, the common substring between the inflection and root was γ_s; here, γ_s contains a single orthographic vowel change[24], $\delta'_v \rightarrow \delta_v$ which splits γ_s into γ_p and γ_s. Both δ'_v and δ_v represent vowel clusters (V*) which can be zero or more consecutive vowels. It is this $\gamma_p[\delta'_v/\delta_v]\gamma_s$ which is the

[24]Which may be an identity transformation or the empty transformation $\epsilon \rightarrow \epsilon$.

Wordframe described by this model. In many circumstances, this may be equivalent to the linguistically plausible stem of the word.

As discussed in Section 3.4.2, the Base model is a special case of the Affix model where the prefix, suffix and canonical endings lists all contain only the empty string. When the Wordframe model is built upon the Base model, ψ'_p, ψ'_s and ψ_s drop out of the representation.

WFBASE: WORDFRAME MODEL BUILT ON TOP OF THE BASE MODEL

	prefix	point-of-prefixation change	common substring	vowel change	common substring	point-of-suffixation change	suffix/ ending
inflection		δ'_p	γ_p	δ'_v	γ_s	δ'_s	
root		δ_p		δ_v		δ_s	

Since this transformation from Affix model to Base model is straightforward[25], the discussion that follows will assume the Affix model is used.

The alignment probability is computed for every combination of suffix, prefix, canonical ending (all provided externally), as well as every internal vowel shift found in the analyzed training data. The probabilities of the three potential changes are multiplied to obtain the final alignment probability (3.8).

$$P(\text{root}, \text{POS} \,|\, \text{inflection}) =$$

$$P(\delta_p\gamma_p\delta_v\gamma_s\delta_s\psi_s, \pi \,|\, \psi'_p\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s\psi'_s) =$$

$$\max_{\psi'_p\in\Psi'_p,\psi'_s\in\Psi'_s,\psi_s\in\Psi_s,(\delta'_v,\delta_v)\in \mathrm{I}} P\left(\delta'_p \rightarrow \delta_p \,|\, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \pi\right) * P\left(\delta'_s \rightarrow \delta_s \,|\, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \pi\right) * P\left(\delta_v|\delta'_v, \pi\right) \tag{3.8}$$

In the above equation, the internal vowel shift (δ'_v, δ_v) is chosen from a set, I, of internal vowel shifts which are isolated when analyzing training data. $P\left(\delta_v|\delta'_v\right)$ is the

[25]Simply set $\Psi'_p = \Psi'_s = \Psi_s = \{\epsilon\}$.

derived from the unsmoothed counts as in (3.9), making the naïve assumption that the context in which a vowel change occurs in irrelevant.

$$P\left(\delta_v|\delta'_v\right) = \frac{C\left(\delta'_v, \delta_v\right)}{C\left(\delta'_v\right)} \tag{3.9}$$

For every training pair, and once the segments have been identified (Section 3.5.2), the stem change $\delta'_s \rightarrow \delta_s$ is stored in a suffix trie, and $\delta'_p \rightarrow \delta_p$ is stored in a separate prefix trie. These tries maintain the backoff probabilities described in (3.13) and (3.12).

$$
\begin{aligned}
&P(\text{root, POS} \,|\, \text{inflection}) \\
&= P(\delta_p\gamma_p\delta_v\gamma_s\delta_s\psi_s, \pi|\psi'_p\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s\psi'_s) \\
&= \sum_{\psi'_p,\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s,\psi'_s} P(\delta_p\gamma_p\delta_v\gamma_s\delta_s\psi_s, \pi|\psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s) \cdot P(\psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s|\psi'_p\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s\psi'_s) \\
&\approx \max_{\psi'_p,\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s,\psi'_s} P(\delta_p\gamma_p\delta_v\gamma_s\delta_s\psi_s, \pi|\psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s) \cdot P(\psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s|\psi'_p\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s\psi'_s) \\
&= \max_{\psi'_p,\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s,\psi'_s} P(\psi_s|\delta_p\gamma_p\delta_v\gamma_s\delta_s, \psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s, \pi) \cdot P(\delta_p\gamma_p\delta_v\gamma_s\delta_s, \pi|\psi'_p, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s, \psi'_s) \cdot \\
&P(\psi'_p, \gamma_s\delta'_s, \psi'_s|\psi'_p\gamma_s\delta'_s\psi'_s)
\end{aligned}
$$

Making the independence assumptions that all possible splits are equally likely,

as are all canonical endings.

$$
\begin{aligned}
&= \max_{\psi'_p,\gamma_s\delta'_s,\psi'_s} \frac{1}{\#\text{endings}} \cdot P(\delta_p\gamma_p\delta_v\gamma_s\delta_s, \pi|\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s) \cdot \frac{1}{\#\text{splits}} \\
&= \text{constant} \cdot P(\delta_p\gamma_p\delta_v\gamma_s\delta_s, \pi|\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s) \\
&= \text{constant} \cdot P(\delta'_v \to \delta_v, \delta'_p \to \delta_p, \delta'_s \to \delta_s, \pi|\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s)
\end{aligned}
\tag{3.10}
$$

The previous step is justified because the rules in the trie, and the requirement that only the final vowel cluster may change, allow only one way of rewriting $\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s$ into $\delta_p\gamma_p\delta_v\gamma_s\delta_s$. Expand using the chain rule:

$$=\text{constant}\cdot$$
$$P(\delta'_v \rightarrow \delta_v|\pi, \delta'_p, \gamma_p\delta'_v\gamma_s, \delta'_s) \cdot P(\delta'_p \rightarrow \delta_p|\pi, \delta'_p\gamma_p\delta'_v\gamma_s, \delta'_s) \cdot P(\delta'_s \rightarrow \delta_s, \pi|\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s) \tag{3.11}$$

The probabilities of the point-of-suffixation and point-of-prefixation changes are found by using a mixture model from rules stored in the trie, as before. As before, the point-of-suffixation probabilities below are implicitly conditioned on the applicability of the change to $\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s$. The point-of-prefixation probabilities are implicitly conditioned on the applicability of the change to $\delta'_p\gamma_p\delta'_v\gamma_s$, i.e. once δ'_s has been removed. The vowel change probability is conditioned on the applicability of the change to the last non-final vowel cluster in $\gamma_p\delta'_v\gamma_s$.

$$\begin{aligned} P\left(\delta'_s \rightarrow \delta_s, \pi \,|\, \delta'_p\gamma_p\delta'_v\gamma_s\delta'_s\right) \approx \lambda_{s_1} P(\delta'_s \rightarrow \delta_s) \cdot P(\pi) + (1-\lambda_{s_1})\cdot \\ \Big[\lambda_{s_2} P(\delta'_s \rightarrow \delta_s, \pi) + (1-\lambda_{s_2})\cdot \\ \Big[\lambda_{s_3} P(\delta'_s \rightarrow \delta_s, \pi \,|\, last_1(\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s))) \\ + (1-\lambda_{s_3}) \cdot \Big[\ldots\Big] \quad \Big] \quad \Big] \end{aligned} \tag{3.12}$$

$$P\left(\delta'_p \rightarrow \delta_p \,|\pi, \delta'_p\gamma_p\delta'_v\gamma_s, \delta'_s\right) \approx \lambda_{p_1} P(\delta'_p \rightarrow \delta_p) + (1-\lambda_{p_1})\cdot$$
$$\Big[\lambda_{p_2} P(\delta'_p \rightarrow \delta_p \,|\,\pi) + (1-\lambda_{p_2})\cdot$$
$$\Big[\lambda_{p_3} P(\delta'_p \rightarrow \delta_p \,|\pi, first_1(\delta'_p\gamma_p\delta'_v\gamma_s\delta'_s)))$$
$$+ (1-\lambda_{p_3})\cdot\Big[\ldots\Big]\quad\Big]\quad\Big] \tag{3.13}$$

$$P(\delta'_v \rightarrow \delta_v|\pi, \delta'_p, \gamma_p\delta'_v\gamma_s, \delta'_s) \approx \quad P(\delta'_v \rightarrow \delta_v) \tag{3.14}$$

3.5.2 Analysis of Training Data

The key difference between the suffix models and the Wordframe model is how training pairs are analyzed. Before being able to model the prefix and suffix stem changes, the inflection and root must each be segmented into their constituent parts. As with the Affix model, the affix lists are treated as ranked lists and these affixes, ψ'_p, ψ'_s, and ψ_s are all stripped from the inflection and root. The analysis of the remaining substrings is the same as analysis of the full strings in the WFBase model.

In the previous Affix model, once the affixes were removed, γ_s was the longest common prefix. This is not sufficient here since the Wordframe model is also representing a point-of-prefixation change. Furthermore, γ_s can not simply represent the longest common *substring* because this model is also representing internal vowel changes within γ_s. Instead, the substring pair $(\gamma_p\delta'_v\gamma_s,\ \gamma_p\delta_v\gamma_s)$ is defined to be the longest common substring with at most one internal vowel cluster ($V* \rightarrow V*$) transformation. Should there be multiple

inflection → root	ψ'_p	δ'_p	γ_p	δ'_v	γ_s	δ'_s	ψ'_s	δ_p	γ_p	δ_v	γ_s	δ_s	ψ_s
ENGLISH													
kept → keep			k	e	p		t		k	ee	p		
sang → sing			s	a	ng				s	i	ng		
SPANISH													
acuerto → acortar			ac	ue	rt		o		ac	o	rt		ar
conmuevo → conmover			conm	ue	v		o		conm	o	v		ar
CZECH													
nepase → pást	ne		p	a	s		e		p	á	s		t
GERMAN													
gestunken → stinken	ge		st	u	nk		en		st	i	nk		en
gefielt → gefallen	ge		f	ie	l		t		f	a	l	l	en
DUTCH													
gedroogd → drogen	ge		dr	oo	g		d		dr	o	g		en
afgedroogd → afdrogen		afge	dr	oo	g		d	af	dr	o	g		en
TAGALOG													
pinutulan → putol		pin	ut	u	l		an	p	ut	o	l		

Table 3.28: Effectiveness of Wordframe model to handle internal vowel shifts. The final two examples show the mishandling of infixation. For clarity, positions storing ϵ have been left blank.

"longest" substrings, the substring closest to the start of the inflection is chosen.[26] In practice, there is rarely more than one such "longest" substring.

Table 3.28 shows how this representation handles the internal vowel shifts in a number of languages.

3.5.3 Additional resources required by the Wordframe models

From the problem definition, it follows that both δ'_v and δ_v must contain only vowels since the representation $\delta'_v \rightarrow \delta_v$ is meant to model the internal vowel changes due

[26]This places a bias in favor of the changes happening at the end of the string and is motivated by the large number of languages which are suffixal and the relative few that are not. This could be adjusted for prefixal languages.

to inflection. In order to determine this, a list of vowels for a language is required.[27] Since the relationship between phonology and orthography is often haphazard, this vowel list is, at best, a representation of those letters that may be involved in such phonological vowel shifts.

These lists are not the definitive lists for vowels in each of the languages. Such a list would be difficult to create since there are some "sometimes" vowels, such as the English *y*. In general, the decision to include or exclude letters in these lists was an arbitrary decision based purely on a visual inspection of the data.

Specifically, for all languages[28], this list was composed of the letters *a, e, i, o, u* and all of their accented versions (e.g. *á, ö*). For some languages, this list was augmented with extra vowels: *y* in Czech, Icelandic, Norwegian, Polish, Swedish, Uzbek, and both *y* and *w* in Welsh.

If the WF model is used with the Base model architecture (WFBase), no additional resources are required (see Section 3.5.1). If the Affix model is used, then lists of prefixes, suffixes and canonical endings are required, as before.

3.5.4 Wordframe Effectiveness

In Table 3.23, it was observed that the Affix model was not capable of modeling the internal vowel shift *ue* → *o* found in many Spanish verb inflections. The Wordframe model is able to effectively isolate this vowel change, as shown in Table 3.29.

The WF model is also able to model prefixes as word initial changes when no prefix

[27] If one wishes to model arbitrary internal changes, this "vowel" list could be made to include every letter in the alphabet. Results are not presented for this configuration of the vowel list.

[28] Except Greek and Russian, where the character set is different.

INFLECTION	→	ROOT	δ'_v	→	δ_v	δ'_s	→	δ_s	ψ'_s	ψ_s
acuerto	→	acortar	ue	→	o	ϵ	→	ϵ	o	ar
acuesto	→	acostar	ue	→	o	ϵ	→	ϵ	o	ar
almuerzo	→	almorzar	ue	→	o	ϵ	→	ϵ	o	ar
aluengo	→	alongar	ue	→	o	ϵ	→	ϵ	o	ar
apruebo	→	aprobar	ue	→	o	ϵ	→	ϵ	o	ar
cuelgo	→	colgar	ue	→	o	ϵ	→	ϵ	o	ar
concuerdo	→	concordar	ue	→	o	ϵ	→	ϵ	o	ar
conmuevo	→	conmover	ue	→	o	ϵ	→	ϵ	o	er
consuelo	→	consolar	ue	→	o	ϵ	→	ϵ	o	ar
cuento	→	contar	ue	→	o	ϵ	→	ϵ	o	ar
desenvuelvo	→	desenvolver	ue	→	o	ϵ	→	ϵ	o	er
duermo	→	dormir	ue	→	o	ϵ	→	ϵ	o	ir
muero	→	morir	ue	→	o	ϵ	→	ϵ	o	ir
muestro	→	mostrar	ue	→	o	ϵ	→	ϵ	o	ar
ruedo	→	rodar	ue	→	o	ϵ	→	ϵ	o	ar
ruego	→	rogar	ue	→	o	ϵ	→	ϵ	o	ar
suelto	→	soltar	ue	→	o	ϵ	→	ϵ	o	ar
trueno	→	tronar	ue	→	o	ϵ	→	ϵ	o	ar
vuelco	→	volcar	ue	→	o	ϵ	→	ϵ	o	ar

Table 3.29: Modeling the Spanish $eu \rightarrow o$ vowel shift. ψ'_p, δ'_p, and δ_p, always ϵ in this example, have been omitted for clarity.

INFLECTION	→	ROOT	δ'_p	γ_p	δ'_v	γ_s	δ'_s	δ_p	γ_p	δ_v	γ_s	δ_s
pevul	→	vul	pe	ϵ	ϵ	vul	ϵ	ϵ	ϵ	ϵ	vul	ϵ
pevum	→	vum	pe	ϵ	ϵ	vum	ϵ	ϵ	ϵ	ϵ	vum	ϵ
SuQeq	→	Qeq	Su	ϵ	ϵ	Qeq	ϵ	ϵ	ϵ	ϵ	Qeq	ϵ
pevup	→	vup	pe	ϵ	ϵ	vup	ϵ	ϵ	ϵ	ϵ	vup	ϵ
bIHutlh	→	Hutlh	bI	ϵ	ϵ	Hutlh	ϵ	ϵ	ϵ	ϵ	Hutlh	ϵ
matlhu'	→	tlhu'	ma	ϵ	ϵ	tlhu'	ϵ	ϵ	ϵ	ϵ	tlhu'	ϵ

Table 3.30: Modeling Klingon prefixation as word-initial stem changes. Notice that both $\delta'_v \rightarrow \delta_v$ and γ_p are null.

list is provided. The Klingon examples in Table 3.30 point out not only the capability to isolate prefixes, but also the special case where there is no internal change.

One weakness of the WF model, as illustrated by the the final two examples shown in Table 3.28, is the inability to isolate infixation. There are a number of segmental issues involved with identifying infixes in training data and no solutions are being proposed here.

3.6 Evaluation

3.6.1 Performance

As with the suffix models, all of the experiments in this section were performed by doing 10-fold cross-validation on the evaluation data.

Table 3.32 compares the performance of the WF models to the non-WF models on which they are based. Because of the huge performance gains in Greek, Irish, Tagalog, Swahili, and Klingon, an unweighted average of all accuracies, as well as an average with these five exceptional languages removed, are presented in the table. Regardless of which average is used, the best performing model overall is the simple Affix model.

δ'_v	$\rightarrow$	δ_v	$\mathrm{P}(\delta'_v \rightarrow \delta_v)$
ie	$\rightarrow$	e	0.637496
ie	$\rightarrow$	i	0.287451
ie	$\rightarrow$	ie	0.062291
ie	$\rightarrow$	a	0.002735
ie	$\rightarrow$	ϵ	0.001823

Table 3.31: Example internal vowel shifts extracted from Spanish training data. The high probability for the non-identity substitution *ie* $\rightarrow$ *e* causes performance degradation in the Wordframe model.

This is due in large part to the over-application of the internal vowel change in the Wordframe especially languages which do not exhibit such internal vowel change phenomena and hence any application is potentially spurious. Because vowel changes are not conditionalized on context or position within the word, these changes can often be applied in positions and contexts where they had never been seen before.[29] Table 3.31 contains a list of five internal vowel changes observed in Spanish training data along with the probabilities assigned to each. Most surprising in this list is that the identity change *ie* $\rightarrow$ *ie* is far less likely than the change *ie* $\rightarrow$ *e*. Along with others, this adversely affects the Spanish roots *adiestrar*, *alienar*, *arriesgar*, and *orientar* found in the evaluation set. All contain an internal *ie* which, in training data, never exhibited the transformation *ie* $\rightarrow$ *e*. Using *orientemos*, an inflection of *orientar*, as an example, the analysis for the correct root *orientar* is over 10 times less likely than the analysis for *orentar* since the identity vowel change is 10 times less likely than the *ie* $\rightarrow$ *e* change. This is because the probabilities for the point-of-prefixation change ($\epsilon \rightarrow \epsilon$), and the point-of-suffixation change ($\epsilon \rightarrow \epsilon$ assuming that the suffix *emos* is in the suffix list and the ending *ar* is in the endings list) are equivalent

[29] Making the internal vowel changes sensitive to position and context remains as future work.

in both of the analyses to *orientar* and *orentar*.

3.6.2 Model combination

While the Wordframe models do underperform the non-Wordframe models on average across languages in isolation, there is substantial potential benefit to using them, especially for those languages with prefixation or large numbers of inflections with internal vowel changes. In order to capture or derive the benefits of both models, simple model combination was initially used. In the following three tables, Tables 3.33, 3.34 and 3.35, the combinations were done using a linear combination of the models with each model weighted equally.

The combination of the Base model and the WFBase model (those models not requiring external lists of prefixes, suffixes and endings) results in the best performance in all but four languages (Table 3.33). In three of these remaining four languages less than a 0.1% decrease in performance was observed when using the combined model.

The paired combination of the Affix model and the WFAffix model (Table 3.34) results in the best model for all languages except two (Danish and Tagalog) where the performance decrease in the combined model relative to the single best model was less than 0.04%.

Within 0.1%, the combination of all four models (Table 3.35) was the best performing model relative to all of the individual models and all of the combined pairs of models.

Language	Base	Affix	WFBase	WFAffix
Spanish	94.62%	**96.48%**	90.18%	95.24%
Portuguese	97.35%	97.33%	**97.85%**	97.51%
Catalan	84.53%	84.73%	**85.81%**	85.51%
Occitan	89.58%	88.74%	**92.22%**	91.86%
French	99.04%	**99.32%**	96.09%	98.66%
Italian	98.06%	**98.12%**	95.63%	97.68%
Romanian	**96.96%**	95.73%	95.46%	96.08%
Latin	**88.47%**	**88.47%**	82.90%	83.87%
English	98.43%	**98.73%**	98.24%	98.57%
Danish	96.56%	94.86%	97.16%	**97.78%**
Norwegian	93.71%	-	**95.50%**	-
Swedish	97.77%	97.35%	97.96%	**98.06%**
Icelandic	84.15%	84.02%	91.58%	**91.87%**
Hindi	**84.77%**	-	**84.77%**	-
Sanskrit	87.75%	-	**88.94%**	-
Estonian	82.81%	96.22%	82.62%	**96.33%**
Tamil	**90.95%**	-	89.95%	-
Finnish	**97.35%**	**97.35%**	96.78%	96.76%
Turkish	99.36%	**99.41%**	98.85%	98.52%
Uzbek	**99.44%**	**99.44%**	99.11%	99.39%
Basque	94.54%	94.03%	94.83%	**95.02%**
Czech	78.70%	**98.15%**	96.91%	**98.15%**
Polish	97.20%	**97.22%**	97.14%	97.07%
Russian	**85.84%**	-	85.18%	-
Greek	15.62%	**99.48%**	17.71%	91.15%
German	92.04%	92.05%	**94.95%**	94.69%
Dutch	86.44%	**93.76%**	83.21%	79.21%
Irish	43.87%	-	**89.11%**	-
Welsh	**87.58%**	-	87.53%	-
Tagalog	0.76%	91.77%	89.81%	**95.97%**
Swahili	2.94%	93.84%	96.77%	**96.93%**
Klingon	0.00%	**100.00%**	**100.00%**	**100.00%**
Average	79.60%	**95.06%**	90.34%	94.88%
Partial Average	92.00%	**94.83%**	92.49%	94.66%

Table 3.32: Stand-alone MorphSim accuracy differences between the four models. The *Average* row is an average of all languages presented (when available). The *Partial Average* row excludes the five languages with exceptionally large differences between the Suffix and Wordframe models (Tagalog, Swahili, Klingon, Greek and Irish).

Language	Base	WFBase	Combined Base+WFBase
Spanish	94.66%	90.23%	**95.04%**
Portuguese	97.37%	**97.88%**	97.82%
Catalan	84.92%	86.67%	**90.31%**
Occitan	89.60%	92.56%	**93.16%**
French	99.05%	96.25%	**99.13%**
Italian	98.09%	95.72%	**98.36%**
Romanian	96.97%	95.58%	**97.74%**
Latin	88.60%	83.30%	**91.38%**
English	98.43%	98.32%	**98.54%**
Danish	96.63%	**97.23%**	**97.23%**
Norwegian	93.71%	95.60%	**95.85%**
Swedish	97.85%	98.05%	**98.17%**
Icelandic	84.29%	91.98%	**92.20%**
Hindi	**84.77%**	**84.77%**	**84.77%**
Sanskrit	87.75%	88.99%	**89.44%**
Estonian	83.01%	83.34%	**84.71%**
Tamil	**90.95%**	89.95%	**90.95%**
Finnish	97.47%	96.98%	**97.48%**
Turkish	99.40%	99.02%	**99.41%**
Uzbek	**99.44%**	99.16%	99.42%
Basque	94.66%	95.00%	**95.24%**
Czech	81.32%	97.05%	**98.08%**
Polish	97.23%	97.19%	**97.47%**
Russian	86.01%	85.84%	**90.80%**
Greek	15.62%	17.71%	**25.52%**
German	92.60%	95.03%	**97.87%**
Dutch	87.23%	83.90%	**95.44%**
Irish	45.69%	89.49%	**95.46%**
Welsh	87.66%	87.90%	**88.55%**
Tagalog	1.59%	**89.92%**	88.24%
Swahili	2.94%	**96.77%**	96.66%
Klingon	0.00%	**100.0%**	**100.0%**

Table 3.33: Accuracy of combined Base MorphSim models. Model combination done by taking an unweighted average of the probabilities of the proposed roots. Roots proposed by one model and not proposed by the other were averaged with 0.

Language	Affix	WFAffix	Combined Affix+WFAffix
Spanish	96.48%	95.24%	**96.60%**
Portuguese	97.33%	97.54%	**97.57%**
Catalan	84.90%	86.18%	**87.63%**
Occitan	88.75%	92.19%	**92.27%**
French	99.32%	98.67%	**99.33%**
Italian	98.15%	97.73%	**98.22%**
Romanian	95.74%	96.14%	**96.43%**
Latin	88.60%	84.28%	**91.23%**
English	98.73%	98.65%	**98.87%**
Danish	94.86%	**97.80%**	97.78%
Norwegian	-	-	-
Swedish	97.39%	98.11%	**98.31%**
Icelandic	84.02%	92.06%	**92.39%**
Hindi	-	-	-
Sanskrit	-	-	-
Estonian	96.24%	96.44%	**96.90%**
Tamil	-	-	-
Finnish	97.47%	96.97%	**97.48%**
Turkish	99.44%	98.97%	**99.48%**
Uzbek	99.44%	99.44%	**99.45%**
Basque	94.18%	95.12%	**95.21%**
Czech	98.16%	98.18%	**98.31%**
Polish	97.25%	97.14%	**97.46%**
Russian	-	-	-
Greek	**99.48%**	91.15%	**99.48%**
German	92.65%	94.73%	**97.74%**
Dutch	93.82%	79.36%	**97.89%**
Irish	-	-	-
Welsh	-	-	-
Tagalog	91.95%	**96.03%**	95.99%
Swahili	93.84%	**96.93%**	**96.93%**
Klingon	**100.0%**	**100.0%**	**100.0%**

Table 3.34: Accuracy of combined Affix MorphSim models. (See Table 3.33)

Language	Base + WFBase	Affix + WFAffix	Base+WFBase + Affix+WFAffix
Spanish	95.04%	96.60%	**97.28%**
Portuguese	97.82%	97.57%	**97.89%**
Catalan	90.31%	87.63%	**90.65%**
Occitan	93.16%	92.27%	**93.39%**
French	99.13%	99.33%	**99.58%**
Italian	98.36%	98.22%	**98.43%**
Romanian	97.74%	96.43%	**97.84%**
Latin	**91.38%**	91.23%	91.36%
English	98.54%	98.87%	**99.05%**
Danish	97.23%	97.78%	**97.87%**
Norwegian	95.85%	-	95.85%
Swedish	98.17%	98.31%	**98.44%**
Icelandic	92.20%	92.39%	**92.58%**
Hindi	84.77%	-	84.77%
Sanskrit	89.44%	-	89.44%
Estonian	84.71%	**96.90%**	96.83%
Tamil	90.95%	-	90.95%
Finnish	97.48%	**97.48%**	97.48%
Turkish	99.41%	**99.48%**	99.46%
Uzbek	99.42%	99.45%	**99.46%**
Basque	95.24%	95.21%	**96.05%**
Czech	98.08%	98.31%	**98.62%**
Polish	97.47%	97.46%	**97.52%**
Russian	90.80%	-	90.80%
Greek	25.52%	99.48%	**100.0%**
German	97.87%	97.74%	**97.93%**
Dutch	95.44%	97.89%	**98.35%**
Irish	95.46%	-	95.46%
Welsh	88.55%	-	88.55%
Tagalog	88.24%	95.99%	**97.48%**
Swahili	96.66%	**96.93%**	96.91%
Klingon	100.0%	100.0%	100.0%

Table 3.35: Accuracy of all combined MorphSim models. (See Table 3.33)

	$\omega(root) = 1$ (no rootlist)	$\omega(root)$ from evaluation data			$\omega(root)$ from evaluation+dictionary		
	Accuracy	Accuracy	Coverage	Precision	Accuracy	Coverage	Precision
Spanish	86.48%	95.24%	95.90%	99.31%	94.83%	95.97%	98.81%
Portuguese	92.21%	97.51%	97.80%	99.71%	97.41%	97.94%	99.45%
Catalan	67.90%	85.51%	86.28%	99.12%	-	-	-
Occitan	82.25%	91.86%	91.90%	99.96%	-	-	-
French	90.39%	98.66%	99.04%	99.61%	98.58%	99.05%	99.52%
Italian	93.00%	97.68%	97.72%	99.95%	97.64%	97.85%	99.78%
Romanian	89.83%	96.08%	96.17%	99.90%	96.04%	96.30%	99.73%
Latin	69.36%	83.87%	84.36%	99.42%	-	-	-
English	93.11%	98.46%	98.51%	99.95%	97.49%	99.43%	98.04%
Danish	81.00%	97.78%	97.97%	99.80%	97.01%	98.30%	98.69%
Norwegian	80.40%	95.50%	95.85%	99.63%	-	-	-
Swedish	87.38%	98.06%	98.85%	99.20%	97.21%	99.12%	98.07%
Icelandic	70.91%	91.87%	92.88%	98.92%	-	-	-
Hindi	81.64%	84.77%	84.77%	100.0%	-	-	-
Sanskrit	79.68%	88.94%	88.99%	99.94%	-	-	-
Estonian	81.81%	96.33%	96.65%	99.67%	96.26%	96.66%	99.58%
Tamil	79.23%	89.95%	89.95%	100.0%	-	-	-
Finnish	85.22%	96.76%	96.92%	99.84%	-	-	-
Turkish	92.59%	98.52%	98.59%	99.93%	97.68%	98.98%	98.69%
Uzbek	95.74%	99.39%	99.42%	99.97%	-	-	-
Basque	80.11%	95.02%	95.19%	99.82%	94.78%	95.43%	99.32%
Czech	85.16%	98.15%	98.80%	99.34%	97.66%	98.89%	98.76%
Polish	92.34%	97.07%	97.31%	99.75%	96.90%	97.64%	99.24%
Russian	67.30%	85.18%	87.47%	97.38%	79.66%	89.14%	89.37%
Greek	85.94%	91.15%	91.15%	100.0%	91.15%	91.15%	100.0%
German	84.60%	94.69%	95.25%	99.42%	94.16%	95.55%	98.54%
Dutch	58.22%	79.21%	79.90%	99.14%	77.75%	81.24%	95.70%
Irish	70.80%	89.11%	89.11%	100.0%	-	-	-
Welsh	66.05%	87.53%	87.78%	99.71%	-	-	-
Tagalog	81.66%	95.97%	98.29%	97.64%	-	-	-
Swahili	74.43%	96.93%	96.99%	99.94%	96.84%	96.99%	99.85%
Klingon	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Table 3.36: Effect of $\omega(root)$ weighting factor set using the evaluation data root list and enhanced with a dictionary. All inflections for which the filter $\omega(root)$ produced no valid analyses are not answered. As with the Affix and the Base model, the performance decrease when adding a dictionary is less than 1% on average. (When $\omega(root) = 1$, coverage is 100%)

	All Words		Regular		Semi-Regular		Irregular	
Language	Base + WFBase	Affix + WFAffix	Base + WFBase	Affix + WFAffix	Base + WFBase	Affix + WFAffix	Base + WFBase	Affix + WFAffix
Spanish	95.04%	**96.60%**	96.13%	**97.02%**	91.95%	**95.14%**	81.33%	**91.19%**
Catalan	**90.31%**	87.63%	**95.96%**	95.51%	83.91%	83.91%	**74.41%**	64.18%
Occitan	**93.16%**	92.27%	**98.20%**	98.18%	96.64%	**98.32%**	**53.08%**	43.52%
French	99.13%	**99.33%**	**99.64%**	99.60%	98.92%	**99.59%**	93.04%	**96.02%**
Italian	**98.36%**	98.22%	**98.74%**	98.60%	99.52%	**99.61%**	**92.88%**	92.53%
Romanian	**97.74%**	96.43%	**98.94%**	97.54%	94.35%	**94.67%**	**84.34%**	83.19%
English	98.54%	**98.87%**	99.19%	**99.35%**	98.66%	**99.50%**	34.38%	**40.62%**
Danish	97.23%	**97.78%**	98.03%	**98.48%**	93.18%	**95.00%**	86.83%	**87.80%**
Norwegian	95.85%	95.85%	97.57%	97.57%	90.62%	90.62%	76.38%	76.38%
Icelandic	92.20%	**92.39%**	**97.71%**	97.54%	**97.79%**	97.35%	62.71%	**64.95%**
Hindi	84.77%	84.77%	98.58%	98.58%	33.33%	33.33%	14.29%	14.29%
Turkish	99.41%	**99.48%**	99.94%	**99.96%**	95.27%	**95.77%**	88.71%	88.71%
Welsh	88.55%	88.55%	89.27%	89.27%	86.69%	86.69%	32.84%	32.84%
Rel. Avg.	100.0%	99.81%	100.0%	99.95%	99.31%	100.0%	99.97%	100.0%

Table 3.37: Performance of the combined Base models vs the combined Affix models on different types of inflections. On regular inflections, the simpler model (Base+WFBase) is more successful than the more complex model; however, on semi-regular inflections, those including point-of-affixation changes and internal vowel changes, the Affix+WFAffix model is more successful. Irregular inflections are not handled well by either model.

3.6.3 Training size

As expected, the amount of available training data has a major effect on the performance of the supervised methods presented. This effect is particularly pronounced for highly inflected and agglutinative languages. However, languages with minimal inflection (English and Danish) or a fairly regular inflection space (French) show much less pronounced drops in accuracy as training size decreases.

To demonstrate this, Figures 3.2 and 3.3 show the performance of the WFBase model as the number of training samples is reduced. In these experiments, and all of the experiments in Section 3.6.3, 10% of the evaluation data[30] was held out as test and the remaining 1% - 90% was used as training data.

Figures 3.4 and 3.5 show a more detailed look at the effect of training size on French. On their own, the WFBase and WFAffix models underperform the Base and Affix models, regardless of training data size (Figure 3.4). However, combinations of the two Base models, the two Affix models, and all four models outperform their individual components when used alone. In addition, the combination of all four models outperforms all of the other models, regardless of the training size.

Model combination was done by taking an unweighted average of the probabilties for each proposed root for each model. Roots proposed by one model but not proposed by another model were averaged with 0.

Figure 3.5 shows the performance of regular, semi-regular, and irregular inflections with respect to training size. As expected, variations in the training size have a large effect

[30] By type.

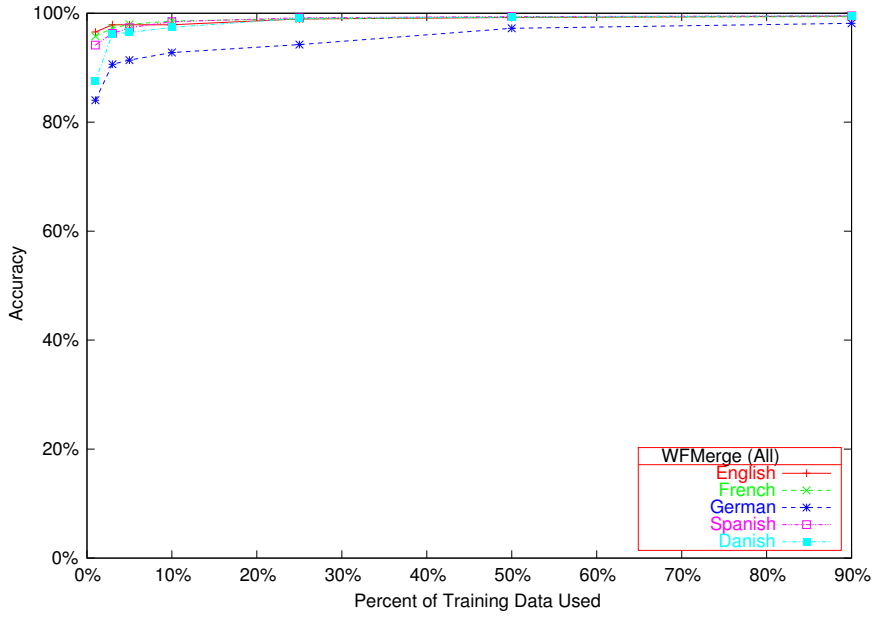


Figure 3.2: Effect of training size on the accuracy of the combined models (Base, Affix, WFBase and WFAffix) in five non-agglutinative suffixal languages

on the irregular inflections, but only a minimal effect on the regular inflections.

3.7 Morphological Generation

Morphological generation is the inverse of morphological analysis. In analysis, the goal is to find the mapping (inflection) → (root, POS); whereas in generation, one finds the reverse mapping, (root, POS) → (inflection).

This relation is actually symmetric as (root, POS) ↔ (inflection). Because of this bidirectionality, the root-inflection pairs discovered in morphological analysis can correctly be used directly as the root-inflection pairs for morphological generation.

However, determining the part of speech of an inflection in analysis is often non-

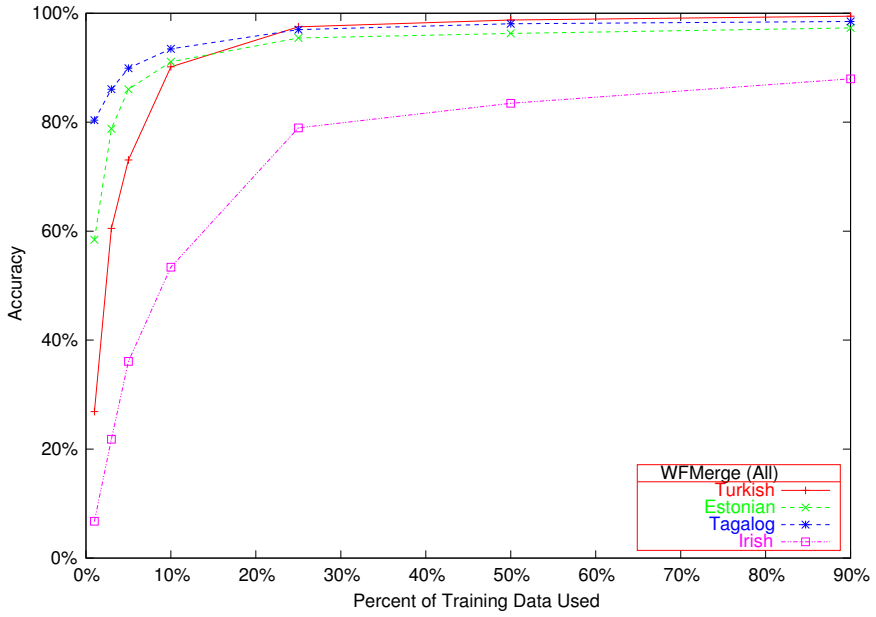


Figure 3.3: Effect of training size on the accuracy of the combined models (Base, Affix, WFBase and WFAffix) in agglutinative and prefixal languages.

essential for many tasks such as WSD and IR;[31] the dimensionality reduction achieved by clustering multiple inflections to their common lemma is the goal, and this is often insensitive to the part of speech. In generation, however, part of speech can never be optional.[32] This is because while morphological analysis is a many-to-one mapping from inflection to root,[33] generation is a one-to-many mapping. In other words, while it is correct to say that the root of the English verb *jumping* is *jump*, it is not correct to say that *the* inflection of *jump* is *jumping*, since *jumps* and *jumped* are also valid inflections of *jump*. Unless one's goal is to generate the full inflectional paradigm, part of speech is necessary in order to generate

[31] And is omitted from the evaluation of the previous sections

[32] This has the consequence that in practice it is often necessary to train a separate morphological generation system rather than simply use the output of a morphological analyzer.

[33] There exist a few scattered exceptions to this, including the English plural noun *axes* whose root can be either *axe* or *axis*.

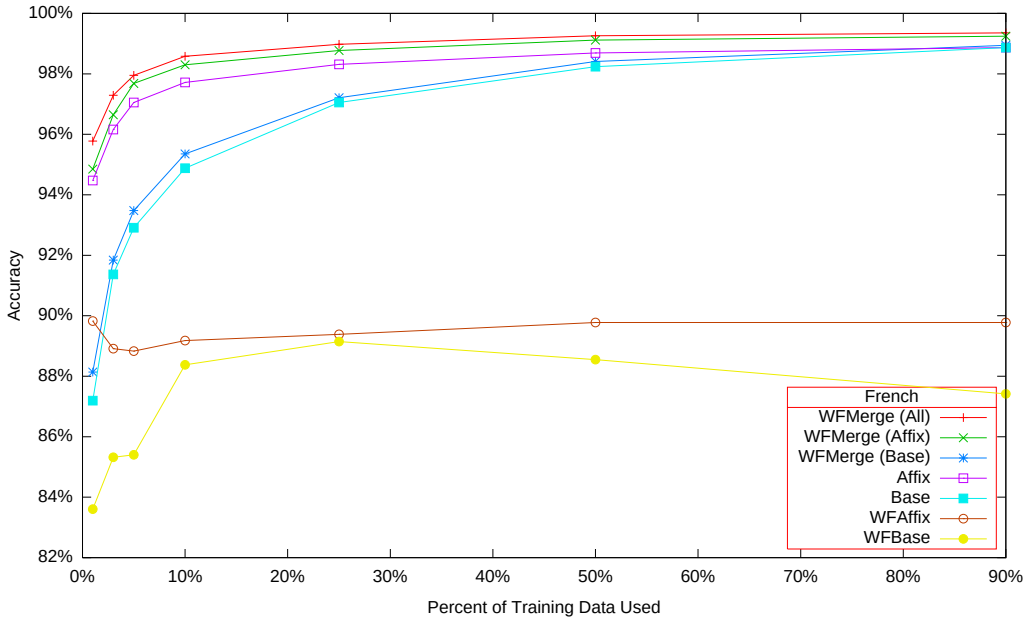
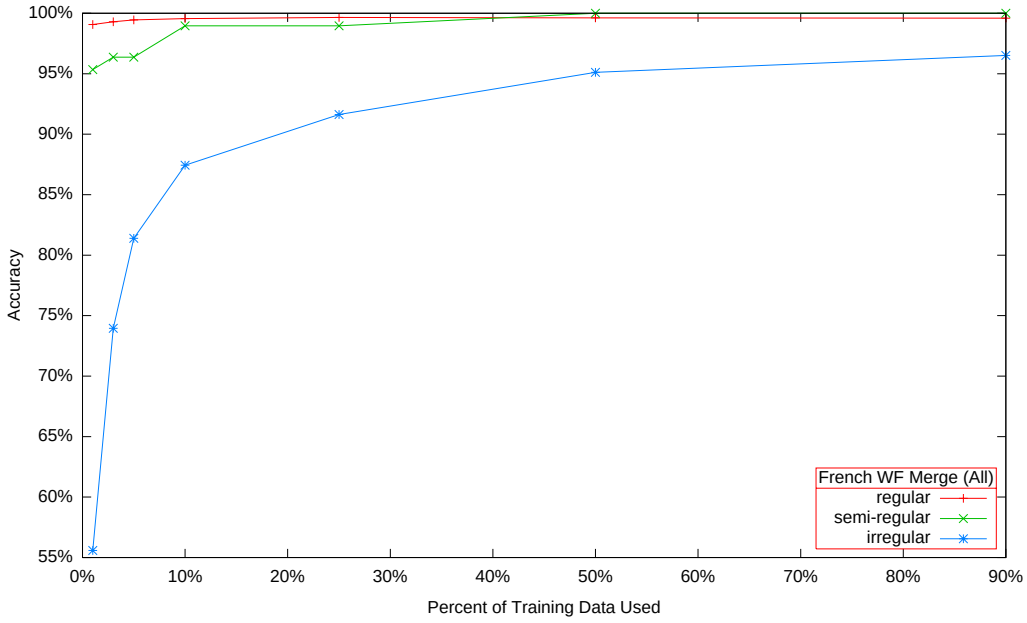


Figure 3.4: Effect of training size on the accuracy of seven models in French. *WFMerge(All)* refers to a combination of a all four models, whereas *WFMerge(Affix)* and *WFMerge(Base)* refer to the combination of the Affix model with the WFAffix model, WFMerge(Affix), and the Base model with the WFBase model, WFMerge(Base).



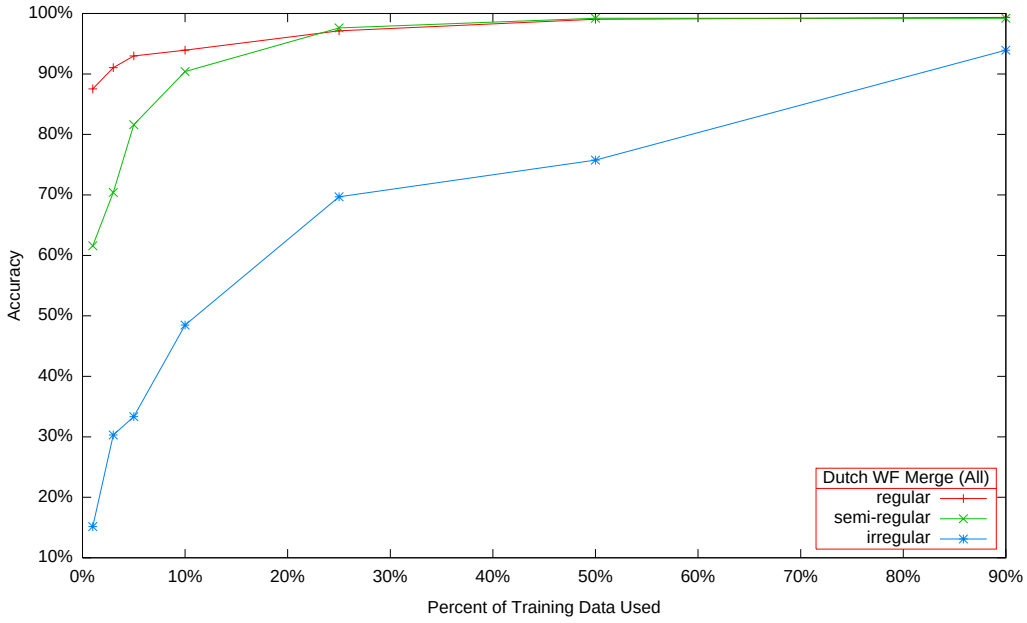
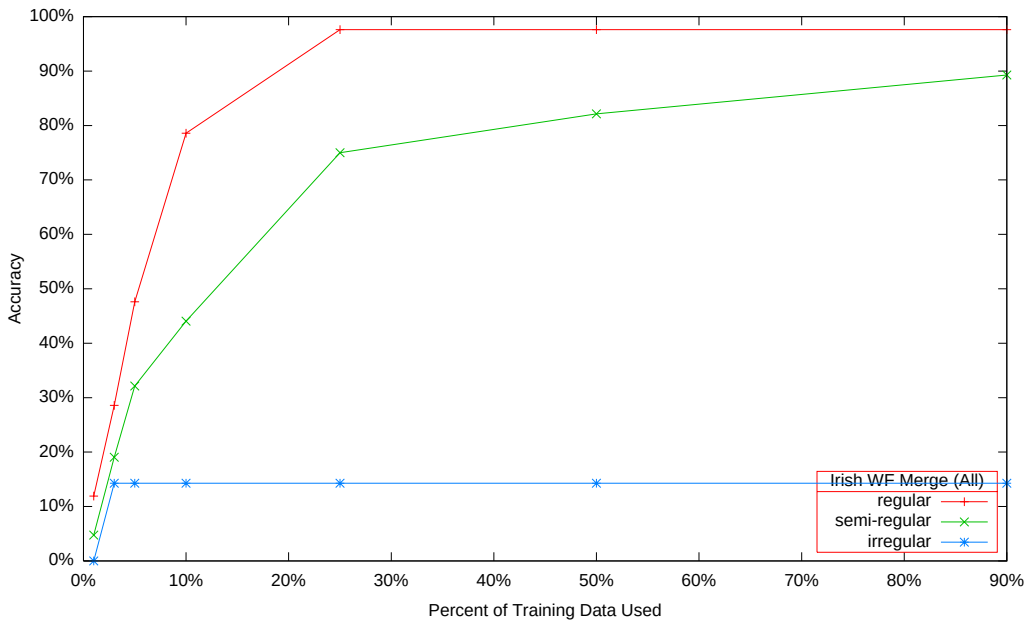


Figure 3.5: Effect of training size on regular and irregular inflections in French and Dutch



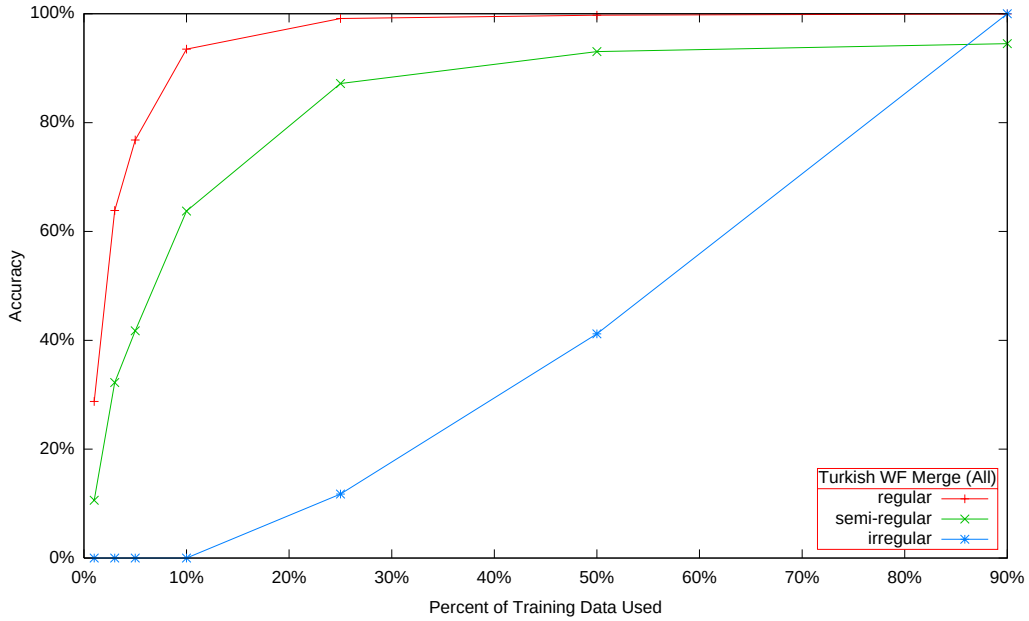


Figure 3.6: Effect of training size on regular and irregular inflections in Irish and Turkish

suffix (ψ'_s)	canonical ending (ψ_s)	part of speech	training data
erait	-er	3rd person present conditional	*aiderait* - *aider*
rait	-re	3rd person present conditional	*prendrait* - *prendre*
ait	-er	3rd person imperfect indicative	*aidait* - *aider*
it	-ir	3rd person past indicative	*applaudit* - *applaudir*
t	-re	3rd person present indicative	*dit* - *dire*

Table 3.38: Partial suffix inventory for French and associated training data

the single correct inflection or the intended part of speech.

This required part of speech makes the problem of generation using a supervised learner a simpler task than the problem of analysis. In analysis, since the part of speech of the inflection is not typically known ahead of time, the training data must include inflection-root pairs for all parts of speech. In comparison, one can think of a morphological generator for a language being composed of separate morphological generators for each tense (e.g. for verbs), each separately trained on inflection-root pairs for only one fine-grained part of speech.[34] Then, at generation time, the part of speech of the desired inflection will determine which sub-generator will be used.

As an illustration, the French 3rd-person-present-conditional inflection-root pair *aimerait-aimer* is presented using the point-of-suffixation change notation of the Affix model:

$\gamma_s\delta'_s\psi'_s$	$\rightarrow$	$\gamma_s\delta_s\psi'_s$	γ_s	δ'_s	ψ'_s	$\rightarrow$	γ_s	δ_s	ψ_s	δ_s	$\rightarrow$	δ'_s	ψ'_s
aimerait	$\rightarrow$	aimer	aim	ϵ	erait	$\rightarrow$	aim	ϵ	er	ϵ	$\rightarrow$	ϵ	erait

However, a partial suffix inventory, like that shown in Table 3.38 would yield a number of competing analyses (Table 3.39). Note that these competing analyses are not

[34]Training these models separately makes point-of-affixation changes, which might generalize across parts of speech, more difficult to learn. Though not presented here, one could combine a model trained on only a single part of speech with a model trained on all parts of speech.

$\gamma_s\delta_s\psi'_s$	$\rightarrow$	$\gamma_s\delta'_s\psi_s$	γ_s	δ_s	ψ'_s	$\rightarrow$	γ_s	δ'_s	ψ_s	δ'_s	$\rightarrow$	δ_s	ψ'_s
aimerait	$\rightarrow$	aimer	aim	ϵ	erait	$\rightarrow$	aim	ϵ	er	ϵ	$\rightarrow$	ϵ	erait
aimerait	$\rightarrow$	aimere	aime	ϵ	rait	$\rightarrow$	aime	ϵ	re	ϵ	$\rightarrow$	ϵ	rait
aimerait	$\rightarrow$	aimerer	aimer	ϵ	ait	$\rightarrow$	aimer	ϵ	er	ϵ	$\rightarrow$	ϵ	ait
aimerait	$\rightarrow$	aimerair	aimera	ϵ	it	$\rightarrow$	aimera	ϵ	ir	ϵ	$\rightarrow$	ϵ	it
aimerait	$\rightarrow$	aimeraire	aimerai	ϵ	t	$\rightarrow$	aimerai	ϵ	re	ϵ	$\rightarrow$	ϵ	t

Table 3.39: Multiple competing analyses based on the partial suffix inventory for French

easily disambiguated since they all involve an $\epsilon \rightarrow \epsilon$ change. Without a dictionary to indicate that *aimer* was a root but not *aimere*, *aimerer*, *aimerair*, or *aimeraire*,

In fact, this is just a subset of the possibilities. In a contrived example, training on *only* the training data shown in Table 3.38, using *only* those suffixes as shown in the suffix inventory, and using *no* root list, analyzing *aimerait* provides the inflection-root analysis confidence scores and rankings shown in Table 3.40.

In generation, there is no such ambiguity. Since one builds generation modules for each fine-grained part of speech separately, the only training data used to generate the 3$^{\text{rd}}$-person-present-conditional of *aimer* are the pairs *aider-aiderait* and *prendrait-prendre*. In practice, the generation models can also be trained separately for canonical root endings, reflecting the tendency for canonical endings to be markers for inflectional paradigms. In this case, the only remaining training pair is *aider-aiderait*.

Although a contrived example, the ability for the morphological generator to effectively remove training examples from the training set which are not of the correct part of speech and do not end in the correct canonical root ending means that a morphological generator will be trained on a more selective set of examples than is possible with the morphological analyzer, which is typically trained on all inflections of all parts of speech.

inflection	root	confidence	rank
aimerait	aimeraire	0.94	1
aimerait	aimerre	0.40	2
aimerait	aimre	0.40	2
aimerait	**aimer**	**0.40**	**2**
aimerait	aimeer	0.40	2
aimerait	aimere	0.40	2
aimerait	aimerare	0.40	2
aimerait	aimeraer	0.40	2
aimerait	aimerer	0.40	2
aimerait	aimerair	0.20	10
aimerait	aimerir	0.20	10
aimerait	aimeir	0.20	10
aimerait	aimir	0.20	10
aimerait	aimeraier	0.04	14
aimerait	aimeraiir	0.02	15

Table 3.40: Incorrect analysis of French inflection *aimerait* due to competing suffixes, limited training data, and no root list. All of the proposed roots are combinations of suffixes from Ψ'_s and endings from Ψ_s. If *aimere, aimerer, aimerair,* or *aimeraiir* existed as French roots, *aimer* would be one of its morphologically regular inflections.

Of course, this is a very limited example, based only on a handful on training instances. Using larger amounts of training data to train a generation model to handle only 3$^{\text{rd}}$ person present conditional of French *-er* verbs, using the suffix *-erait* (and a backoff to the null suffix, ϵ) yields the larger set of $\delta'_s \rightarrow \delta_s$ point-of-suffixation change patterns shown in Table 3.41.

Notice that the roots *agneler* and *haceler* are shown with ambiguous inflections. While it is rare for the same inflection to have two different roots, many roots will have an alternate inflection for the same part of speech. In some rare cases, there can be three or more valid inflections for the same (root, POS) pair. For example, the Welsh verb *gwybod*, meaning "to know" has three alternative conjugations for the present impersonal tense:

pattern	δ'_s	$\rightarrow$	δ_s	ψ'_s	$P(\delta'_s \rightarrow \delta_s)$	example inflections
1	ϵ	$\rightarrow$	ϵ	erait	0.975	abaisserait, abandonnerait, abdiquerait, ...
2	y	$\rightarrow$	i	erait	0.011	aboierait, atermoierait, balaierait, ...
3	ϵ	$\rightarrow$	t	erait	0.0033	feuilletterait, jetterait, projetterait, ...
4	et	$\rightarrow$	èt	erait	0.0027	achèterait, furèterait, rachèterait, ...
5	el	$\rightarrow$	èl	erait	0.0027	agnèlerait, démantèlerait, harcèlerait, ...
6	uy	$\rightarrow$	oi	erait	0.0020	appoierait, ennoierait, essoierait
7	ϵ	$\rightarrow$	l	erait	0.0013	agnellerait, harcellerait
8	oy	$\rightarrow$	er	erait	0.0013	enverrait, renverrait
9	all	$\rightarrow$	irait	ϵ	0.0007	irait

Table 3.41: $\delta'_s \rightarrow \delta_s$ point-of-suffixation patterns generated from 1494 training pairs of French *-er* <root, 3rd person present conditional inflection> pairs

		analysis								
root	inflection	γ_s	δ'_s	ψ_s	$\rightarrow$	γ_s	δ_s	ψ'_s	pattern	probability
aimer	aimerait	aim	ϵ	er	$\rightarrow$	aim	ϵ	erait	1	0.999995
aimer	aimterait	aim	ϵ	er	$\rightarrow$	aim	t	erait	3	0.0000034
aimer	aimlerait	aim	ϵ	er	$\rightarrow$	aim	l	erait	7	0.0000013
soudoyer	soudoierait	soudo	y	er	$\rightarrow$	soudo	i	erait	2	0.753
soudoyer	souderrait	soud	oy	er	$\rightarrow$	soud	er	erait	8	0.207
soudoyer	soudoyerait	soudoy	ϵ	er	$\rightarrow$	soudoy	ϵ	erait	1	0.040
soudoyer	soudoyterait	soudoy	ϵ	er	$\rightarrow$	soudoy	t	erait	3	0.00004
soudoyer	soudoylerait	soudoy	ϵ	er	$\rightarrow$	soudoy	l	erait	7	0.00001

Table 3.42: Correct generation of the French 3rd person present conditional inflections *aimerait* and *soudoylerait*. The column labeled *pattern* refers to the pattern number shown in Table 3.41 above.

gwybyddir, *gŵys*, and *wyddys*.

There may be semantic, syntactic or dialectal constraints which strongly favor one form over another. For instance, the English verb "to lie", when used to mean "to stay at rest" has the participle *lain* and past tense *lay*; when meaning "to not tell the truth", has the participle and past tense form *lied*. In context, there is typically no ambiguity. However, in isolation, it is impossible to determine which is correct. In evaluating the results for morphological generation, generating any one of the valid inflections will be considered correct, regardless of other possible selectional preferences.

Table 3.42 shows the result of generating the 3$^{\text{rd}}$ person present conditional of *aimer* and *soudoyer*. Since the overwhelming amount of training examples are analyzed using the regular inflectional process $\epsilon \rightarrow \epsilon$ +erait (97.5% as indicated in Table 3.41), *aimerait* is correctly generated with extremely high confidence. The trie-based contextual backoff model is instrumental in the correct choice of y $\rightarrow$ i +erait for the semi-regular verb *soudoyer*.

Language	Part of Speech	Accuracy	Coverage	Precision
French	1S Present	0.987972	0.987972	1.000000
	2S Present	0.989065	0.989065	1.000000
	3S Present	0.988518	0.989065	0.999447
	1P Present	0.997813	0.997813	1.000000
	2P Present	0.997266	0.997813	0.999452
	3P Present	0.989065	0.989065	1.000000
English	3S Present	0.995892	0.995892	1.000000
	1S Past	0.967954	0.967954	1.000000
	Gerund	0.992611	0.992611	1.000000
	Participle	0.970370	0.970370	1.000000
German	1S Present	0.971146	0.971146	1.000000
	2S Present	0.962902	0.963726	0.999145
	3S Present	0.964551	0.964551	1.000000
	1S Perfect	0.887001	0.889550	0.997135
	1S Preterit	0.896125	0.896125	1.000000

Table 3.43: Accuracy of generating inflections by training individual generation models for the listed parts of speech.

Chapter 4

Morphological Alignment by Similarity Functions

Chapter 3 discussed a highly successful method for performing morphological analysis by creating a probabilistic model for describing orthographic morphological changes from training data consisting of inflection-root pairs. All of the training data used in presenting this model were derived from relatively clean exemplars which were taken from available grammars and morphology references.

In many languages, however, such references do not exist or are limited in their coverage. While it is possible to create these references by using native or fluent speakers to hand-enter training data, the cost of doing so for a single language, let alone a broad range of languages, would be prohibitive.

This chapter will address this limitation, presenting four similarity functions which can be run as unsupervised alignment models capable of automatically deriving training

data from the distributional patterns in text based on relative corpus frequency, contextual similarity, weighted Levenshtein-based string similarity and multilingual projection.

Chapter 5 will show how these models can be iteratively retrained, how they can be used to seed the supervised method presented previously, and how the supervised method can use these similarity functions as a way of handling unseen morphological phenomenon.

4.1 Overview

The goal of the similarity models discussed in this chapter is to create alignments from inflection to root using *no* paired inflection-root examples for training and *no* prior seeding of legal morphological transformations. Chapter 5 will show how these models can be iteratively retrained when potentially noisy inflection-root pairs are available. Because three of the models presented here do not use orthographic information, they have the potential to perform equally well on both regular and highly irregular forms (such as *brought* → *bring*). In fact, these methods will prove to be particularly effective at inducing the highly irregular forms that are not typically handled by other string-based morphology induction algorithms.

The Frequency Similarity model, presented in Section 4.3, uses word counts derived from a corpus to match inflections with potential roots. The motivation behind this model is that distributional properties of words can be used to help to choose the correct analysis in a set of the competing analyses of one inflection. For example, *singed*, the past tense of *singe*, may be misanalyzed as the application of the regular past tense rule *+ed* and yielding the root *sing*. In a large corpus used in the English experiments, the word occurs

sing occurs 1204 times, whereas the words *singed* and *singe* occur 9 and 2 times, respectively. The disparity in the relatively frequency between *sing* and *singed* argues for the analysis *singe*, rather than *sing*.

The Context Similiarity model, presented in Section 4.4, tries to identify the root of an inflection (or vice versa) by comparing the contexts in which the root and inflection are used in a corpus. For each inflection, the context in which it occurs is compared with the context of every other word in the corpus using the cosine similarity measure. This model relies on the fact that an inflection is often used in the context of the same words as its root. In particular, different inflections of the same verb tend to share equivalent selectional preferences (e.g. $<drink \leftrightarrow juice>$) relatively insensitive to the verb tense. For example: "Jack often *drank* orange juice for breakfast." and "Jill tries to *drink* orange or cranberry juice every morning."

The Levenshtein Distance model (Section 4.5) uses a weighted string-edit distance measure to find the root of an inflection. This model takes advantage of the fact that inflections and their roots usually have a very similar orthographic representation.

The Translingual Bridge Similiarity model, introduced in Section 4.6, uses a word-aligned bilingual corpus (bitext) to project morphological information from a language where an analyzer already exists (such as English) into the target language where an analyzer is desired. A French-English bitext with alignments between *chanter* $\leftrightarrow$ *sing* and *chantas* $\leftrightarrow$ *sang* helps to indicate a relationship between *chantas* and *chanter* based on the known relationship between *sang* and *sing*.

4.2 Required and Optional Resources

The alignment models based on relative distributional frequency, contextual similarity, and Levenshtein-based string similarity require only an unannotated monolingual corpus. The projection model requires an aligned bilingual text corpus such that the second language has an existing morphological analyzer. Additional, optional, resources can be incorporated by these models when available.

In a further clarification of these requirements, all of the alignment models assume, and are based on only the following limited set of (often optional[1]) available resources:

(a) (Optional) Used only by the Levenshtein model and the Frequency model, a table (such as Table 4.1) of the inflectional parts of speech of the given language, along with a list of the canonical suffixes, prefixes and/or canonical endings for each part of speech. These affix lists serve to isolate the actual orthographic changes that occur in the root form when adding an affix. Additionally, they can also be used to obtain a noisy set of candidate examples for each part of speech. [2]

(b) (Required) A large *unannotated* text corpus used by the Context model and the Frequency model.

(c) (Optional) All of the models presented in this chapter can take advantage of a (potentially noisy or incomplete) list of the candidate roots for each POS in the language (typically obtainable from a dictionary), and any rough mechanism for identifying the

[1]When present, the optional resources can provide higher precision systems; however, the inclusion of an optional resource does not ensure precision increases, as will be seen throughout this chapter.

[2]The lists need not be exhaustive. In addition, a list of affixes, without the mapping from affix to POS, is sufficient for Levenshtein. These lists are the same lists used by the Affix model in Section 3.4.

English :

Part of Speech	VB	VBD	VBZ	VBG	VBN
Canonical Suffixes	+ϵ	+ed (+t) +ϵ	+s	+ing	+en +ed (+t) +ϵ
Examples (*not* used for training)	jump announce take	jumped announced took	jumps announces takes	jumping announcing taking	jumped announced taken

Spanish:

Part of Speech	VRoot	VPI1s	VPI2s	VPI3s	VPI1p	VPI2p	VPI3p
Canonical Suffixes	+ar +er +ir	+o	+as +es	+a +e	+amos +emos +imos	+áis +éis +ís	+an +en

Table 4.1: List of canonical affixes, optionally with a map to part of speech

candidate parts of speech of the remaining vocabulary. This can be based on aggregate models of context or tag sequence or an existing POS tagger for the language. The major function is to partially limit the potential alignment space from unrestricted word-to-word alignments across the entire vocabulary.

Other work, including Cucerzan and Yarowsky [2000], focuses on the problem of bootstrapping approximate tag probability distributions by modeling relative word-form occurrence probabilities across indicative lexical contexts (e.g. "*the* <NOUN> *are*" and "*been* <VBG> *the*"), among other predictive variables, with the goal of co-training with the models presented here. It is not necessary to select the part of speech of a word in any given context, only provide an estimate of the candidate tag distributions across a full corpus.

The source of these candidate tag estimates is unimportant, however, and the lists can be quite noisy. Since all of the models compare each inflection to all words in a given root list, false positives in this list can introduce false alignments, whereas

false negatives ensure that an inflection will not be aligned with its root. Chapter 5 will use the output from these models as training data for the noise-robust supervised methods of Chapter 3 where such misalignments can be tolerated.

(d) (Optional) A list of the consonants and vowels of the language can be used to seed an initial transition cost matrix for the Levenshtein model.

(e) (Optional) While not essential to the execution of the algorithm, a list of common function words of the given language is useful to the extraction of similarity features in the Context model.

(f) (Required) A (potentially noisy) word-aligned bitext between the target language and a second language for which a morphological analysis (or morphological analyzer) already exists is required for the Translingual Bridge Similarity model.

(g) (Optional) If available, model parameters tuned to a previously studied language can be useful as model parameters for the target language, especially if these languages are closely related (e.g. Spanish and Italian, or Turkish and Uzbek).[3]

4.3 Lemma Alignment by Frequency Similarity

The motivating dilemma behind the use the Frequency Similarity measure is the question of how one determines that the past tense of *sing* is *sang* and not *singed*, a potential alternative. The pairing *sing*→*singed* requires only simple concatenation with the canonical suffix, *+ed*, and *singed* is indeed a legal word in the vocabulary (the past tense

[3]These parameters will be explained in detail throughout this chapter.

	VBD,VB	$\frac{VBD}{VB}$	$ln(\frac{VBD}{VB})$
sang/sing	1427/1204	1.19	0.17
singed/sing	9/1204	0.007	-4.90
singed/singe	9/2	4.5	1.50
sang/singe	1427/9	158.5	5.06
All VBD/VB		.85	-0.16

Table 4.2: Frequency distributions for sang-sing and singed-singe

of *singe*). And while few irregular verbs have a true word occupying the slot that would be generated by a regular morphological rule, a large corpus is filled with many spelling mistakes or dysfluencies such as *taked* (observed with a frequency of 1), and such errors can wreak havoc in naïve alignment-based methods, especially when they correspond to a form predicted by regular or semi-regular processes.

How can this problem be overcome? Relative corpus frequency is one useful evidence source. Observe in Table 4.2 that in an 80 million word collection of newswire text the relative frequency distribution of *sang/sing* is 1427/1204 (or 1.19/1), which indicates a reasonably close frequency match, while the *singed/sing* ratio is 0.007/1, a substantial disparity.

However, simply looking for close relative frequencies between an inflection and its candidate root is inappropriate, given that some inflections are relatively rare and *expected* to occur much less frequently than the root form. This is especially true for highly inflected languages.

Thus in order to be able to rank the *sang/sing* and *singed/sing* candidates effectively, it is necessary to be able to quantify how well each fits (or deviates from) expected frequency distributions. To do so, we use simple non-parametric statistics to calculate the probability of a particular $\frac{VBD}{VB}$ ratio by examining how frequently other such ratios in a

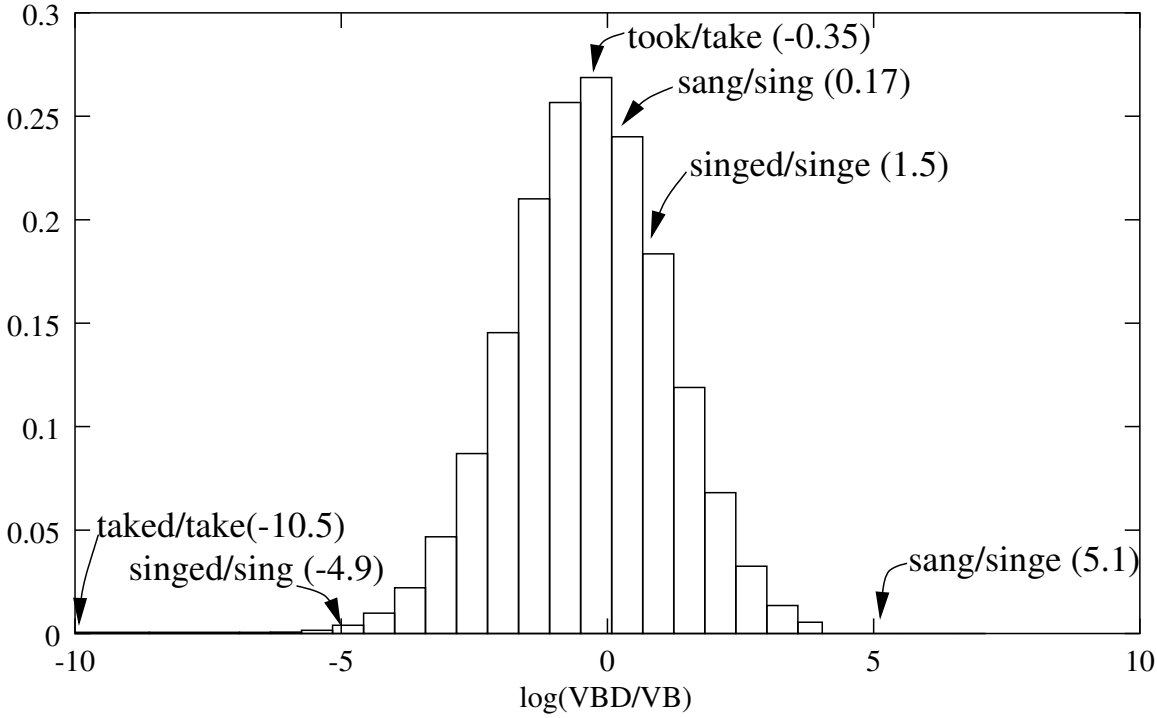


Figure 4.1: Using the $log(\frac{VBD}{VB})$ Estimator to rank potential VBD/VB pairs in English

similar range have been seen in the corpus. Figure 4.1 illustrates such a histogram (based on the log of the ratios to focus more attention on the extrema). The histogram is then smoothed and normalized (such that the sum under the curve is 1) and is used as an approximation of the probability density function for this estimator ($log(\frac{VBD}{VB})$), which we can then use to quantify to what extent a given candidate $log(\frac{VBD}{VB})$, such as log(sang/sing)=.17, fits the empirically motivated expectations. The relative position of the correct and incorrect candidate pairings on the graph suggests that this estimator is indeed informative given the task of ranking potential root-inflection pairings.

However, estimating these distributions presents a problem given that the true alignments (and hence frequency ratios) between inflections are not assumed to be known

VerbType	$\frac{VBD}{VB}$	$\frac{VBG}{VB}$	Avg. Lemma Freq[4]
Regular	.847	.746	861
Irregular	.842	.761	17406

Table 4.3: Consistency of frequency ratios across regular and irregular verb inflections

in advance. Thus to approximate this distribution automatically, one uses the simplifying assumption that the frequency ratios between inflections and roots (largely an issue of tense and usage) is not significantly different between regular and irregular morphological processes.

Table 4.3 and Figure 4.2 illustrate that this simplifying assumption is supported empirically. Despite large lemma frequency differences between regular and irregular English verbs, their relative tense ratios for both $\frac{VBD}{VB}$ and $\frac{VBG}{VB}$ are quite similar in terms of their means and density functions.

Thus, initially the VBD/VB ratios are approximated from an automatically extracted (and noisy) set of verb pairs exhibiting simple and uncontested suffixation with the canonical *+ed* suffix. This distribution is re-estimated as alignments improve, but a single function continues to predict frequency ratios of unaligned (largely irregular) word pairs from the observed frequency of previously aligned (and largely regular) ones.

Additionally, the ratio POS_i/VB is not the only ratio which can be used to predict the expected frequency of POS_i in the corpus. The expected frequency of a viable past-tense candidate for *sing* should also be estimable from the frequency of any of the other inflectional variants.

Assuming that earlier iterations of the algorithm had filled the SING lemma slots

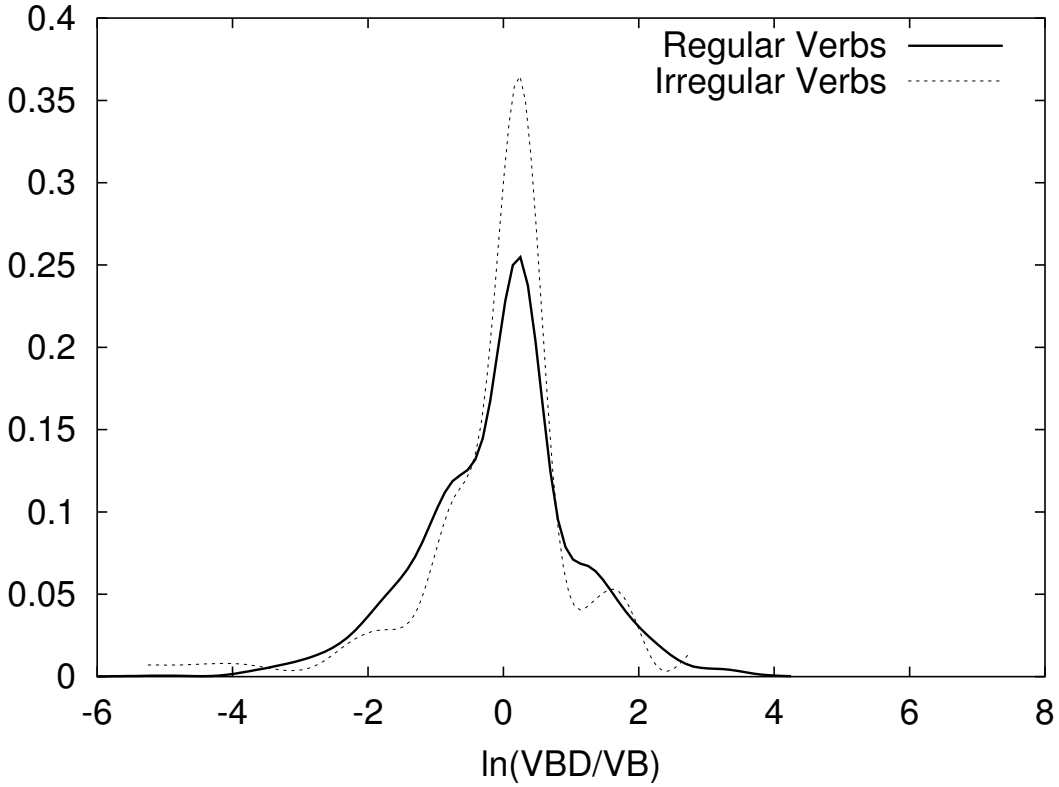


Figure 4.2: Distributional similarity between regular and irregular forms for VBD/VB

for VBG and VBZ in Table 4.4 with regular inflections, $\frac{VBD}{VBG}$ and $\frac{VBD}{VZ}$ can now also used as estimators. Figure 4.3 shows the histogram for the estimator $log(\frac{VBD}{VBG})$. [5]

There are considerable robustness advantages to be gained by averaging the expected frequency of multiple estimators, especially for highly inflected languages where the observed frequency counts may be relatively small for individual tenses. To accomplish this in a general framework, the hidden variable of total lemma frequency (LF) can be estimated (as $\hat{LF}$) via a confidence-weighted average of the observed POS_i frequency and

[5]Using this estimate, a frequency E(VBD)=1567 is predicted, which is a small overestimate relative to the true 1427. In contrast, the distribution for $\frac{VBD}{VBZ}$ is considerably more noisy, given the problems with VBZ forms being confused with plural nouns. This latter measure yields a underestimate of 1184.

	Lemma	VB	VBD	VBG	VBZ	VBN
Word	SING	sing	?	singing	sings	?
Freq	?	1204	?	1381	344	?

Table 4.4: Estimating frequency using non-root estimators

a globally estimated $\frac{\hat{LF}}{\widehat{POS_i}}$ model. Then all subsequent POS_i frequency estimations can be made relative to $\frac{POS_i}{\hat{LF}}$, or a somewhat advantageous variant, $log(\frac{POS_i}{\widehat{LF-POS_i}})$, with this distribution illustrated in Figure 4.4. Another advantage of this consensus approach is that it only requires T rather than T^2 estimators, especially important as the inflectional tag set T grows quite large in some languages.

Also, one can alternately conduct the same frequency-distribution-based ranking experiments over suffixes rather than tags. For example, $log(\frac{+ED}{+ING})$ yields a similar estimator to $log(\frac{VBD}{VBG})$, but with somewhat higher variance.[6] This variance is due to verbs which do not regularly inflect their past tense with $+ed$, such as *knelt* and *kneel*, and also due to those verbs which exhibit different point of suffixation spelling changes when adding $+ed$ versus $+ing$, such as *cried* and *crying*.

Finally, these frequency-based alignment models can be informative even for more highly inflected languages. Figure 4.5 illustrates an estimate of the empirical distribution of the $\frac{VPI3P}{VBINF}$ part-of-speech frequency ratios in Spanish, with this estimator strongly favoring the correct but irregular *juegan/jugar* alignment rather than its orthographically similar

[6]This measure also frees one from any need to do part-of-speech distribution estimation. However, when optional variant suffixes (such as $+ed$ and $+en$) exist in the canonical suffix set, performance can be improved by modeling this distribution separately for verbs with and without observed distinct $+EN$ forms, as the relative distribution of $log(\frac{+ED}{+ING})$ and $log(\frac{+ED}{ROOT})$ change somewhat substantially in these cases. One does not know in advance, however, whether a given test verb belongs to either set. Thus the initial frequency similarity score should be based on the average of both estimators until the presence or absence of the distinct variant form in the lemma can be ascertained on subsequent iterations.

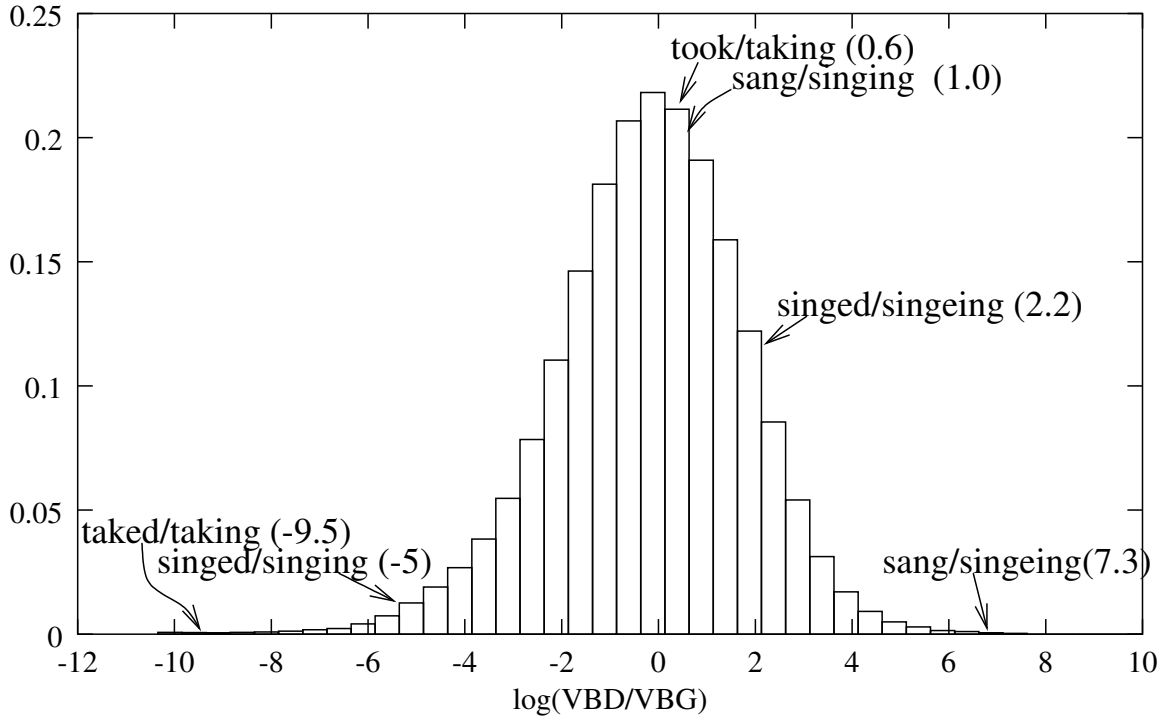


Figure 4.3: Using the $log(\frac{VBD}{VBG})$ Estimator to rank potential VBD-VBG matches in English

competitors.

4.4 Lemma Alignment by Context Similarity

A second powerful measure for ranking the potential alignments between morphologically related forms is based on the contextual similarity of the candidate forms. The traditional cosine similarity between vectors of weighted and filtered context features is used to measure similarity. While this measure also gives relatively high similarity to semantically similar words such as *sip* and *drink*, it is rare even for synonyms to exhibit more similar and idiosyncratic argument distributions and selectional preferences than inflectional variants

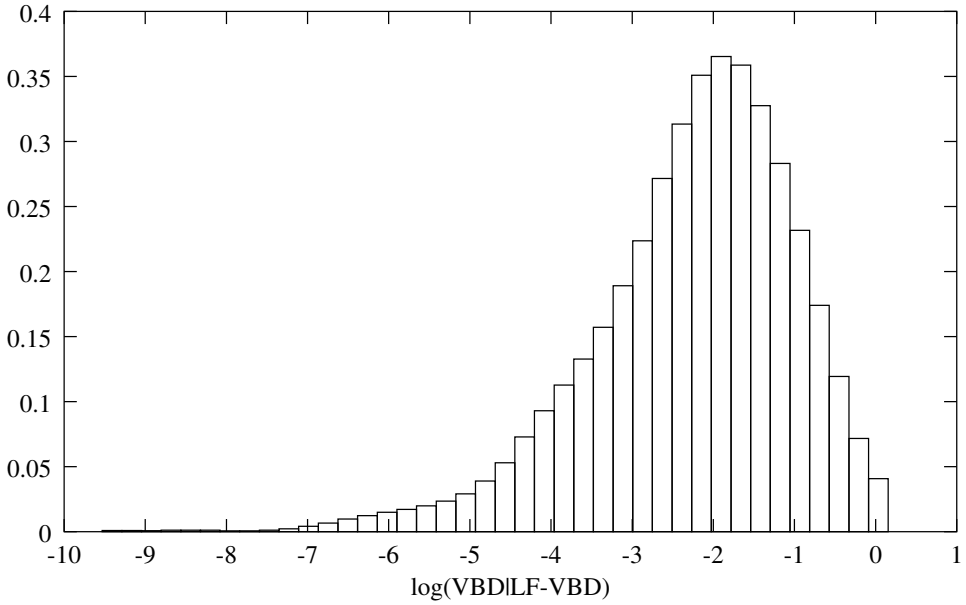


Figure 4.4: Using the $log(\frac{VBD}{\widehat{\neg VBD}})$ Estimator to rank potential VBD-lemma matches in English

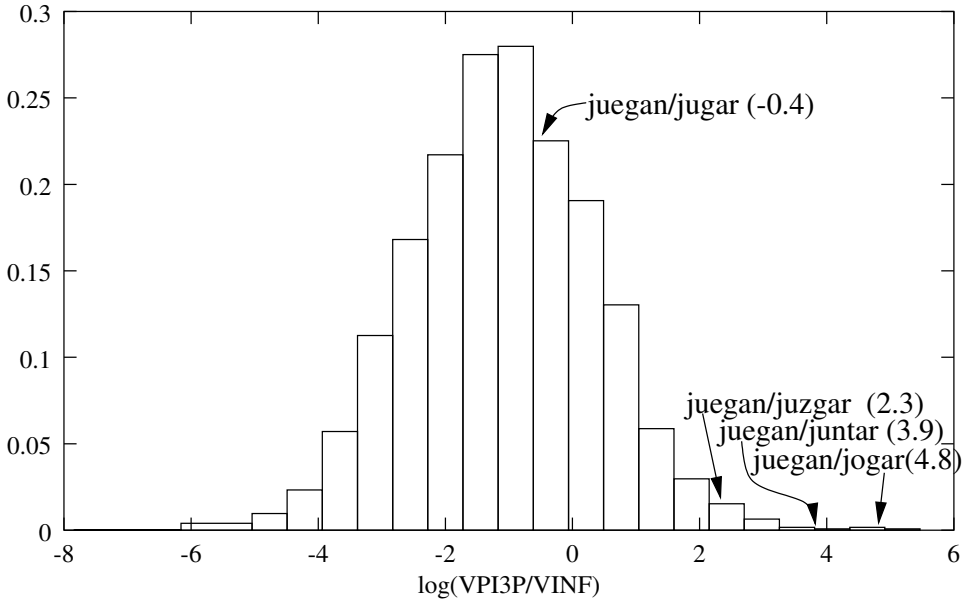


Figure 4.5: Using the $log(\frac{VPI3P}{VINF})$ Estimator to rank potential VBPI3P-VINF pairs in Spanish

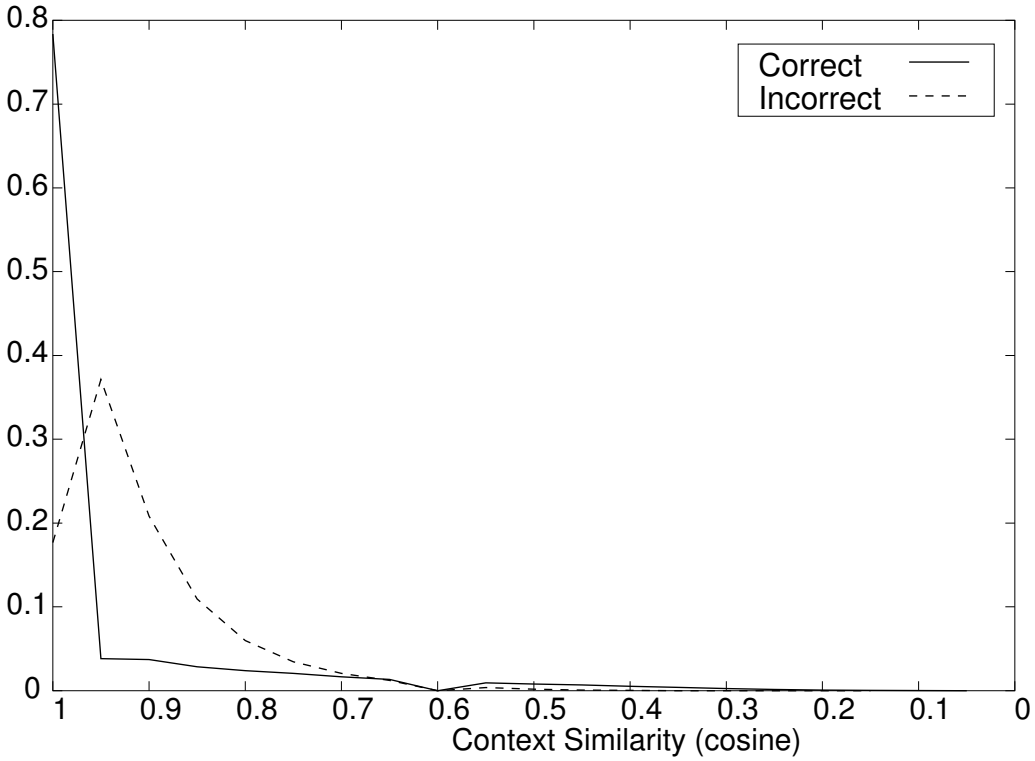


Figure 4.6: Context similarity distributions for correctly and incorrectly aligned inflection-root pairs in French. The distribution for the incorrectly aligned pairs was obtained by aligning an inflection with every root that was not its root. Both distributions have been normalized to sum to 1.

of the same word (e.g. *sipped*, *sipping* and *sip*).

Cosine similarity is computed as below (Equation 4.1), where i and r represent vectors for the inflection and root that are being compared, and w represents all co-occuring words. The vector i (and similarly the vector r) contains weighted counts of each w's co-occurence with i.

$$cos(i, r) = \frac{\sum_w i(w) \cdot r(w)}{\sqrt{\sum_w i(w)^2} \cdot \sqrt{\sum_w r(w)^2}} \tag{4.1}$$

A primary goal in clustering inflectional variants of verbs is to give predominant

vector weight to the head-noun objects and subjects of these verbs which generally provide more information about verb usage than intervening adverbs, adjectives, determiners, prepositions, subordinate clauses, etc. However, to make this maximally language independent, the positions must be approximated by a small set of extremely simple regular expressions over parts of speech, initially including closed-class parts-of-speech and residual content words (CW), e.g.:

$$CW_{subj}\ (\text{AUX}|\text{NEG})^*\ V_{keyword}\ \text{DET?}\ \text{CW}^*\ CW_{obj}.$$

These expressions will clearly extract significant noise and may fail to match many legitimate contexts, but as they can be applied to a potentially unlimited monolingual corpus, the signal-to-noise ratio is tolerable.

A limiting factor in using regular expressions to isolate content words as shown above is that a part of speech tagger is required to identify the content words from the non-content words; however, POS taggers are not available for most languages. As an approximation of a POS tagger, a dictionary can be used to identify a set of POS labels for individual words; however, the coverage of this dictionary across the words in a corpus will be inversely proportional to the degree of inflection the language exhibits (Table 4.5, Figure 4.7). This means that for the most inflected languages, dictionaries will have the least ability in identifying the POS of words in text. Since those languages with limited available NLP resources and with highly inflectional morphological properties are exactly those languages for which the similarity functions presented here are most applicable, heavy reliance can not be placed on using part-of-speech based regular expressions.

It is useful in subsequent iterations of the algorithm to use the previous itera-

Language	avg # of infl per root (avg)	dictionary coverage by token	by type
Swedish	2.47	79.05%	40.30%
English	3.04	82.49%	34.72%
Spanish	29.69	81.66%	29.91%
French	33.75	86.29%	25.15%
Portuguese	36.90	78.94%	27.49%
Italian	38.61	73.74%	24.22%
Turkish	333.83	53.90%	12.77%

Table 4.5: Inflectional degree vs. Dictionary coverage

tion's morphological analysis modules to lemmatize the contextual feature sets. This has the effect of not only condensing the contextual signal and removing potentially distracting correlations with inflectional forms in context, but also increasing the coverage of a dictionary in identifying part-of-speech which can be used in the previously mentioned regular expressions.

4.4.1 Baseline performance

Since the context similarity function is not meant to serve as a standalone morphological analyzer, but rather as a component in a larger system, the actual Top-1 precision can be less important than whether or not the actual root was chosen *near* the top. To estimate this property, Top-10 precision[7] will be used extensively throughout this section when investigating the large parameter space of the context similarity function. Table 4.7 shows the standalone baseline performance of the context similarity function across all the investigated languages with available corpus data for Top-1, Top-5, and Top-10 precision.

[7]Top-10 precision differs from Top-1 precision (or simply "precision") in that an answer is judged correct if the correct answer appears in the top 10 of a weighted list of choices.

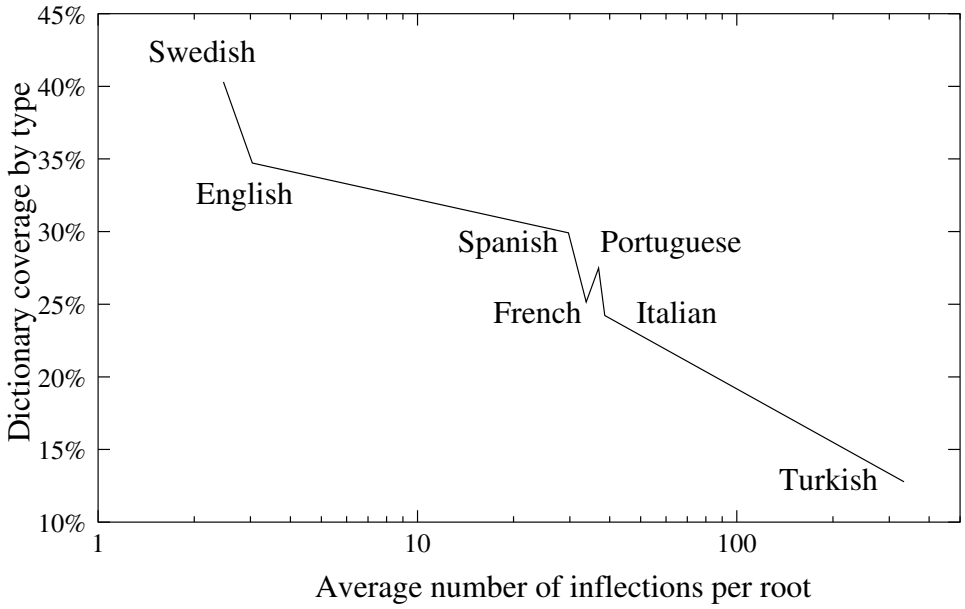


Figure 4.7: Inflectional degree vs. Dictionary coverage

It should be noted that the coverage listed in Table 4.7 does not include evaluation on the full set of available morphological pairs; rather, it is the coverage only for those inflections actually realized in the corpus. For most applications of a morphological analyzer, it is not words in isolation, but words present in a text corpus, that need to be morphologically analyzed. Separate results are presented illustrating the actual coverage of the corpus across the space of infections, roots, and inflection-root pairs in Table 4.6.

In the baseline model, the local context is defined as a bag of words found ± 5 (5x5) tokens from the target word, including all function words and punctuation. While the choice of 5x5 was ad hoc[8], it serves well as a baseline for measuring the performance of other window sizes and configurations that could have been chosen. Performance of the

[8]Chapter 5 shows how this feature can be learned automatically in a fully unsupervised framework from orthogonal similarity models.

	Inflection Coverage		Root Coverage		Infl-Root Coverage	
Language	Types	Pct	Types	Pct	Types	Pct
Basque	5842	100.00%	780	65.82%	5201	89.00%
Czech	23079	97.02%	2971	51.96%	18719	78.59%
Danish	3412	65.65%	812	76.46%	3170	61.00%
Dutch	3145	54.52%	797	78.44%	2993	51.89%
English	3955	80.47%	1142	93.76%	3880	78.94%
Estonian	1728	29.13%	86	58.50%	1475	24.87%
Finnish	7981	10.01%	710	49.51%	6837	8.57%
French	15277	24.04%	1747	95.52%	15201	23.92%
German	4504	31.90%	1188	97.94%	4482	31.74%
Icelandic	1474	36.97%	255	81.21%	1401	35.14%
Italian	11201	17.88%	1414	89.38%	10667	17.02%
Polish	6790	28.62%	522	86.86%	6516	27.46%
Portuguese	5928	26.78%	458	78.42%	5764	26.04%
Romanian	2317	9.31%	356	33.27%	1896	7.62%
Russian	1520	49.54%	147	76.96%	1339	43.64%
Spanish	10295	17.99%	925	77.73%	11948	19.84%
Swahili	1488	5.36%	161	19.68%	902	3.25%
Swedish	12005	86.55%	2574	63.79%	9891	70.63%
Tagalog	2426	25.59%	156	73.58%	2173	22.92%
Turkish	2864	9.83%	68	78.16%	2800	9.61%

Table 4.6: Corpus coverage of evaluation data

Language	Coverage	Top-1 Precision	Top-5 Precision	Top-10 Precision
Basque	88.99%	6.04%	14.02%	19.75%
Czech	66.06%	2.84%	7.76%	11.05%
Danish	45.39%	10.61%	19.90%	24.61%
Dutch	38.16%	4.32%	11.64%	16.32%
English	55.73%	9.97%	18.15%	22.38%
Estonian	24.87%	14.97%	30.15%	42.20%
Finnish	7.90%	3.06%	8.31%	12.09%
French	23.92%	19.45%	32.59%	38.40%
German	23.41%	5.19%	10.60%	14.78%
Icelandic	28.84%	8.01%	18.38%	23.43%
Italian	17.02%	2.42%	6.97%	9.99%
Polish	25.26%	4.85%	13.42%	18.91%
Portuguese	23.98%	10.98%	20.88%	26.58%
Romanian	6.19%	5.00%	12.15%	18.71%
Russian	38.85%	15.15%	30.72%	40.91%
Spanish	19.64%	8.87%	16.59%	20.94%
Swahili	2.68%	1.65%	4.68%	8.68%
Swedish	52.97%	2.41%	5.85%	8.49%
Tagalog	21.28%	2.97%	8.82%	15.08%
Turkish	9.61%	12.35%	30.36%	44.40%

Table 4.7: Baseline context similarity precision (Top-x precision is the percentage of the time that the correct answer was chosen in the top x)

context similarity function when stop words and punctuation have been eliminated, as well as variations on window size, position, and positional weighting schemes will be presented in Section 4.4.2.

The IR technique of using tf-idf Salton and McGill [1983] (term frequency inverse document frequency) was applied to the vectors before performing cosine similarity. Inverse document frequency is used to reduce the weight of frequently occurring words, since, intuitively, they are less indicative of topic than those words which are occur less frequently. Tf-idf has become a standard technique when comparing document vectors; however, performance without tf-idf weighting is presented in Section 4.4.2.

Language	1x1	2x2	3x3	4x4	5x5	10x10
Italian	1.17%	**3.05%**	3.03%	2.87%	2.42%	1.44%
Basque	4.71%	**7.25%**	6.64%	6.31%	6.04%	4.73%
Turkish	15.93%	**16.47%**	15.50%	13.68%	12.35%	9.00%
Russian	11.43%	**18.01%**	16.50%	16.16%	15.15%	9.68%
Swahili	2.25%	2.07%	**2.34%**	1.79%	1.65%	1.93%
Swedish	1.08%	2.60%	**2.86%**	2.71%	2.41%	1.69%
Dutch	2.37%	3.95%	**4.41%**	4.18%	4.32%	3.05%
Polish	3.01%	5.40%	**5.58%**	5.17%	4.85%	2.13%
Romanian	3.46%	5.65%	**5.72%**	5.65%	5.00%	3.83%
Danish	6.44%	11.03%	**12.30%**	12.05%	10.61%	7.00%
English	5.59%	9.75%	**10.77%**	10.41%	9.97%	8.00%
Estonian	15.42%	16.97%	**17.12%**	16.11%	14.97%	12.60%
Czech	1.10%	2.42%	2.88%	**2.95%**	2.84%	1.66%
Icelandic	5.22%	7.67%	8.97%	**9.06%**	8.01%	6.01%
Tagalog	2.41%	3.28%	3.13%	**3.49%**	2.97%	3.03%
Spanish	2.68%	7.29%	8.65%	**9.02%**	8.87%	6.67%
Portuguese	4.32%	9.29%	10.66%	**11.15%**	10.98%	8.13%
Finnish	1.70%	2.30%	2.91%	3.00%	**3.06%**	2.50%
German	2.35%	3.68%	5.03%	5.15%	**5.19%**	4.76%
French	3.46%	11.23%	16.83%	18.95%	**19.45%**	15.96%

Table 4.8: Top-1 precision of the context similarity function with varying window sizes

4.4.2 Evaluation of parameters

Context window size

In the baseline context similarity performance, reported in Table 4.7, the local context is a 5x5 bag of equally weighted tokens which includes all function words and punctuation. Since the choice of this window size was ad hoc, performance of the standalone cosine similarity function is presented using a number of variations on the window size in Table 4.8.

From this, it appears that the choice of a 5x5 window size was not ideal, but

	1x1	2x2	3x3	4x4	5x5	10x10
Average	6.38%	8.86%	9.61%	**9.68%**	9.10%	7.17%

Table 4.9: Average Top-1 performance of the context similarity function by window size across all languages

that perhaps 3x3 would have been a better choice since more languages did best in this configuration. Table 4.9, which shows the average performance of the languages across the different window sizes, indicates a slight preference for the 4x4 window size.

However, individual languages show quite a range of preference for this context window size, with performance in French with a window size of 5x5 performing nearly 75% better than with 2x2, and performance in Turkish performing over 33% better with a 2x2 window size than with 5x5. Interestingly, there seems to be little preference for a particular window size for groups of related languages. Some of the most similar languages, Italian, Romanian, Spanish and French, all perform best with different window sizes; the same is true for German and Dutch. This may be an artifact of the corpora used, the examples tested. The orthographic conventions of the language when representing determiners and pronouns, the variable tendency for adjectives to precede or follow head nouns, increasing the distance between the verbs and their noun objects. Or, it may simply be that there is no effective method for choosing the initial window size based on the linguistic properties of the languages.

Positional weighting

One of the reasons that larger window sizes are attractive is that they can often include important context words, such as direct objects separated from the inflected verb by

adjectives and determiners. However, by extending the window size, extraneous words can easily be included. The poor performance of these larger window sizes can be seen in the performance when using a 10x10 window (Table 4.8). While the models used here do not extend windows across sentence boundaries, the further away in the sentence the inflected word is from other words in a context window, the less likely it is to be of any significance.

To reflect the desire for large window sizes, while down-playing the significance of words located further away in the window, a linear weighting function was applied to the words in the context window, where the positional weight for each word is determined as:

$$weight(\text{word in position } d) = \frac{n-d}{n}$$

where the window size is nxn[9], or $\pm n$, and d is the number of intervening words between the inflection and a word in the context window.

The following sentence from English illustrates the weighting used in a 5x5 window:

surprising	that	the	mood	was	**subdued**	after	winning	their	first	match
0.2	0.4	0.6	0.8	1.0	TARGET	1.0	0.8	0.6	0.4	0.2

This method provides a marked improvement over the unweighted models. Table 4.10 shows a comparison of the positionally weighted model using a 5x5 window size and the previous models presented in Table 4.8. Here, both a count-based voting method and the average performance clearly indicate that using the positional weights provides a consistent improvement over all of the unweighted window sizes (Table 4.10).

[9]Or, more generally, mxn or nxm where $n >= m$

Language	1x1	2x2	3x3	4x4	5x5	10x10	weighted 5x5
Basque	4.71%	7.25%	6.64%	6.31%	6.04%	4.73%	**8.08%**
Czech	1.10%	2.42%	2.88%	2.95%	2.84%	1.66%	**3.47%**
Danish	6.44%	11.03%	12.30%	12.05%	10.61%	7.00%	**13.66%**
Dutch	2.37%	3.95%	4.41%	4.18%	4.32%	3.05%	**4.95%**
English	5.59%	9.75%	10.77%	10.41%	9.97%	8.00%	**11.68%**
Estonian	15.42%	16.97%	17.12%	16.11%	14.97%	12.60%	**18.87%**
Finnish	1.70%	2.30%	2.91%	3.00%	3.06%	2.50%	**3.38%**
French	3.46%	11.23%	16.83%	18.95%	**19.45%**	15.96%	17.72%
German	2.35%	3.68%	5.03%	5.15%	**5.19%**	4.76%	5.09%
Icelandic	5.22%	7.67%	8.97%	9.06%	8.01%	6.01%	**9.58%**
Italian	1.17%	3.05%	3.03%	2.87%	2.42%	1.44%	**3.52%**
Polish	3.01%	5.40%	5.58%	5.17%	4.85%	2.13%	**5.99%**
Portuguese	4.32%	9.29%	10.66%	11.15%	10.98%	8.13%	**11.59%**
Romanian	3.46%	5.65%	5.72%	5.65%	5.00%	3.83%	**6.56%**
Russian	11.43%	**18.01%**	16.50%	16.16%	15.15%	9.68%	17.93%
Spanish	2.68%	7.29%	8.65%	9.02%	8.87%	6.67%	**9.69%**
Swahili	2.25%	2.07%	**2.34%**	1.79%	1.65%	1.93%	2.20%
Swedish	1.08%	2.60%	2.86%	2.71%	2.41%	1.69%	**3.32%**
Tagalog	2.41%	3.28%	3.13%	**3.49%**	2.97%	3.03%	3.38%
Turkish	15.93%	16.47%	15.50%	13.68%	12.35%	9.00%	**17.18%**
Average	6.38%	8.86%	9.61%	9.68%	9.10%	7.17%	**10.37%**

Table 4.10: Results of Top-1 precision using 6 different fixed weight context windows and a single 5x5 distance weighted window for context similarity

Language	1x1	2x2	3x3	4x4	5x5	10x10	weighted 5x5
Basque	15.95%	20.67%	20.41%	19.68%	19.75%	17.96%	**23.12%**
Czech	5.92%	9.67%	10.86%	11.13%	11.05%	8.56%	**12.36%**
Danish	20.65%	28.89%	29.36%	27.20%	24.61%	16.29%	**31.52%**
Dutch	11.86%	16.64%	17.09%	16.55%	16.32%	13.27%	**18.77%**
English	17.24%	23.51%	24.24%	23.26%	22.38%	18.22%	**26.87%**
Estonian	39.93%	42.71%	44.44%	42.89%	42.20%	39.55%	**45.68%**
Finnish	7.41%	9.92%	11.32%	11.97%	12.09%	11.58%	**13.53%**
French	14.05%	29.27%	36.37%	38.17%	**38.40%**	32.55%	37.41%
German	9.62%	12.39%	13.89%	14.72%	**14.78%**	13.88%	14.62%
Icelandic	21.50%	23.28%	24.56%	23.61%	23.43%	22.30%	**26.13%**
Italian	4.89%	10.64%	11.34%	10.55%	9.99%	6.86%	**12.33%**
Polish	14.64%	19.97%	21.18%	20.14%	18.91%	10.32%	**21.99%**
Portuguese	17.29%	27.11%	28.61%	27.94%	26.58%	21.45%	**30.34%**
Romanian	13.58%	16.37%	17.15%	18.45%	18.71%	16.44%	**19.62%**
Russian	40.29%	46.15%	46.38%	44.70%	40.91%	32.24%	**47.39%**
Spanish	10.37%	21.13%	22.56%	22.02%	20.94%	17.02%	**24.01%**
Swahili	9.99%	**11.02%**	10.06%	9.50%	8.68%	8.54%	10.06%
Swedish	5.95%	10.67%	10.78%	9.70%	8.49%	5.95%	**12.34%**
Tagalog	11.85%	13.24%	**15.85%**	15.13%	15.08%	15.28%	15.44%
Turkish	45.84%	46.31%	46.52%	45.08%	44.40%	40.80%	**47.93%**
Average	20.57%	25.53%	26.65%	26.30%	25.60%	22.37%	**28.16%**

Table 4.11: Top-10 precision as an indicator of performance in context similarity. The Top-10 precision results exhibit language-specific parameter preferences consistent with the Top-1 precision results.

As mentioned previously, the performance of this context similarity function should not be judged solely on its ability to correctly chose the single most similar word to the target inflection. Rather, performance should be judged on the functions ability to identify a candidate set of words with the most similar contexts by using Top-10 precision. Importantly, the performance gains seen by the positional weighting in the Top-1 performance table (Table 4.10) are reflected in the Top-10 performances shown in Table 4.11.

As discussed previously, choosing the window size for the first iteration of the context similarity model can be difficult. However, when using positional weightings, this

Language	weighted 2x2	weighted 3x3	weighted 4x4	weighted 5x5	weighted 6x6	weighted 7x7
Portuguese	**30.34%**	24.62%	27.98%	29.65%	30.24%	29.86%
Russian	46.15%	47.39%	**48.06%**	47.39%	46.46%	45.29%
Turkish	47.93%	47.96%	**48.43%**	47.82%	47.75%	47.17%
Estonian	43.69%	45.14%	45.33%	**45.68%**	45.06%	44.64%
Average	42.03%	41.28%	42.45%	**42.64%**	42.38%	41.74%

Table 4.12: Varying window size using weighted positioning, measured by Top-10 precision

problem becomes less apparent. Table 4.12 shows that the performance of the positional weighting is remarkably consistent across variations in window size. Here, unlike the results from Table 4.9, the average 5x5 window size with positional weighting outperforms the other window size variations.

The consistency of performance across varying window sizes in the positional weighting scheme is important for two reasons. The first is that it partially minimizes the risk when choosing an initial window size since this consistency indicates that the initial choice may not greatly affect precision of the model. The second reason is performance-based: the context similarity function requires storing in memory, or offline on a disk, a large matrix of word co-occurrence information proportional to the size of the context window used. Since the 2x2, 3x3 and 4x4 performances closely approximate the fixed 5x5 performances, it may be practical to use these smaller window sizes on large corpora without a noticeable loss in performance.

Language	with TF-IDF	without TF-IDF	performance decrease
Basque	**19.75%**	9.08%	-54.04%
Czech	**11.05%**	9.38%	-15.12%
Danish	**24.61%**	10.86%	-55.86%
Dutch	**16.32%**	11.91%	-27.02%
English	**22.38%**	9.27%	-58.56%
Estonian	**42.20%**	28.20%	-33.17%
Finnish	**12.09%**	6.08%	-49.74%
French	**38.40%**	17.12%	-55.42%
German	**14.78%**	11.11%	-24.84%
Icelandic	**23.43%**	15.77%	-32.71%
Italian	**9.99%**	5.66%	-43.29%
Polish	**18.91%**	14.03%	-25.84%
Portuguese	**26.58%**	16.36%	-38.46%
Romanian	**18.71%**	14.16%	-24.31%
Russian	**40.91%**	31.90%	-22.02%
Spanish	**20.94%**	11.46%	-45.27%
Swahili	**8.68%**	6.89%	-20.63%
Swedish	**8.49%**	5.82%	-31.41%
Tagalog	**15.08%**	12.10%	-19.73%
Turkish	**44.40%**	30.28%	-31.79%

Table 4.13: Top-10 precision decreases when tf-idf weighting is removed from the baseline context similarity model

Using tf-idf

Table 4.13 illustrates why tf-idf has become a standard technique when comparing document vectors using cosine similarity. For every language, omitting tf-idf weighting yielded at least a decrease in performance of 15%, with four languages showing performance decreases of over 50%. On average, removing tf-idf weighting caused a 35% drop in performance.

Punctuation and Stop words

As mentioned previously in Section 4.4.1, the baseline context similarity function includes punctuation and function words in the contextual bags of words. While their importance will be minimized by using tf-idf weighting, these are generally not considered relevant indicators of topic. Table 4.14 present results for removing punctuation and stop words from the positionally unweighted baseline bag of words model.

While it may seem simple to remove punctuation, neither the use of punctuation, nor the set of symbols considered to be punctuation, is consistent cross-lingually. For example, removing apostrophes from languages such as Italian and French will take away valuable information for morphologically analyzing pronouns (such as *dell'* and *all'* in Italian, *l'* and *c'* in French), removing apostrophes in English loses possessive information and important verbal contractions (*John's* and *can't*), and Uzbek makes heavy use of the apostrophe word internally[10] (*to'rtko'lda*, *buyog'iga* and *qat'iy*).

[10]Uzbek had been written in Arabic until 1927, when it began being written in a Roman character set. Then the Soviets took over and changed the script to Cyrillic. Officially, it's back to Roman, though a brief peek at the web will indicate that not everyone is convinced. Probably the constant conversion between scripts has caused this somewhat unorthodox use of word internal apostrophes which are especially prevalent when following the letter *o*. Their prevelance is not observed in the closely related language Turkish.

Language	Remove Nothing (Baseline)	Remove Only Punctuation	Remove 100 Most Freq + Punctuation	Remove Function Words + Punctuation
Basque	19.75%	20.53%	**22.62%**	20.53%
Czech	11.05%	**11.06%**	7.42%	7.57%
Danish	24.61%	25.84%	**36.02%**	-
Dutch	16.32%	17.09%	**18.39%**	18.31%
English	22.38%	23.48%	**27.87%**	27.12%
Estonian	42.20%	44.24%	**45.48%**	-
Finnish	12.09%	12.57%	**12.76%**	-
French	38.40%	37.54%	**38.99%**	38.39%
German	14.78%	14.78%	**15.28%**	14.10%
Icelandic	23.43%	23.61%	**27.51%**	-
Italian	9.99%	**10.39%**	10.15%	10.22%
Polish	18.91%	**20.16%**	16.17%	15.85%
Portuguese	26.58%	26.88%	**28.33%**	27.48%
Romanian	18.71%	**18.91%**	16.84%	16.97%
Russian	40.91%	**44.26%**	43.17%	33.85%
Spanish	20.94%	21.82%	**24.82%**	24.73%
Swahili	8.68%	**8.99%**	8.11%	-
Swedish	8.49%	8.85%	9.56%	**10.02%**
Tagalog	15.08%	15.11%	**19.44%**	-
Turkish	44.40%	**44.90%**	43.64%	41.90%

Table 4.14: Using Stop Words in Context Similarity. When removing the most frequent 100 words, frequently occurring inflections are never removed. A part-of-speech tagger or dictionary is required to identify function words.

In the results presented here, ad hoc lists of punctuation were created for each language based on simple character histograms derived from the corpus to identify punctuation, and some minimal knowledge of the languages and brief eye-balling of the text to realize when not to include certain symbols. In all cases, with the exceptions of French, German and Spanish, removing punctuation led to performance increases.[11] In general, these increases were very small, as was expected (given that tf-idf would have downweighted all but the most infrequent punctuation). Such language-dependent tables are generally not available from a source such as the Unicode Data Consortium.

Since function words do not contribute to topic information, it is common practice to remove them from the context vectors before performing the cosine similarity measure. While it is true that many of the most common function words will be downweighted by tf-idf, another important concept in context similarity measures for morphology that differs from other word clustering measures is the need to downweight or eliminate context words such as subject pronouns that strongly correlate with only one or a few inflectional forms. Giving such words too much weight can cause different verbs of the same person/number/etc to appear more similar to each other than do the different inflections of the same verb.

Choosing which words are stop words can be difficult. At a minimum, a part-of-speech tagger or dictionary is required in order to identify the part of speech of the words to be removed. The languages for which these similarity methods are most effective are often the languages which do not have these resources available. As mentioned above, it may be important to remove pronouns. For this work, a stop word was considered any

[11] A single tokenization routine was used for all languages, and perhaps this caused the decrease in performance.

word listed in the dictionary belonging to any one of the following part-of-speech categories: article/determiner, conjunction, pronoun, preposition, or numeral[12].

While there were some notable performance gains in English (+21.2%) and Spanish (+18.1%), half of the languages (7 out of the 14 for which a dictionary was available) performed worse than the baseline, including sizeable drops in Czech (−31.5%), Russian (−17.3%), Polish (−16.2%), and Romanian (−9.3%) when removing these function words.

Since part-of-speech information was not available for 6 of the languages, a stopword list was approximated by simply removing the 100 most frequent words[13] found in the corpus. In nearly every language, removing these frequent words increased performance or was nearly equivalent to not removing them.

Word ordering

In all of the previous experiments, the varied parameters included changing the size of the window, using positional weighting, including or excluding tf-idf, and including or excluding various sets of words. In each case, however, words were chosen based on their appearance in a centered window around the target word. This presumes that words to the left of the target word (words occuring before it in a sentence) are equally good content words as words which occur to the right of the target word.

In order to test this, the performance of the model with a 3x3 window (a 6 word

[12]Numbers were not listed in the stopwords, but words representing numbers, if so identified in the dictionary, such as *twenty* or *eighth*, were removed.

[13]The inflections in the target set were excluded from this set for obvious reasons since we do not wish to remove the words we are searching for.

window centered on the target word) was compared with the performance of the model with a 6x0 window and a 0x6 window (a 6 word window positioned directly before or after the target word).

As shown in Table 4.8, the preferred window position correlates strongly with the grammatical word ordering in the language. Languages which are S-V-O[14] select for a window focused after the target word, indicating that the likely region hosting the Object is a better selector for context similarity than the region hosting the Subject. Likewise, S-O-V languages show a strong preference for a window focused before the target word, not surprising given that both the subject and the object will be appearing before the verb. Languages with allowable free word-order (but with default S-V-O ordering) show a weaker preference toward the right-focused window. Lastly, languages with V2 grammars[15] show preference toward centered windows, suggesting the importance of capturing variable object positions in these languages.

Table 4.16 presents results for a subset of these languages at a finer granularity. These results continue to show correlation between the expected positions of the subject and object and the position of the context window.

Corpus size

Increasing the size of the unlabeled corpus provides a dual performance boost for the context similarity measure. Firstly, as the corpus size increases, the number of inflections from the test set that are found in the corpus increases (Figure 4.8). This leads

[14]Languages whose default word ordering is Subject-Verb-Object.
[15]German, for example

Language	Left 6x0	Center 3x3	Right 0x6
S-V-O			
Spanish	6.37%	22.57%	**29.38%**
Portuguese	12.05%	28.61%	**32.87%**
French	9.08%	36.36%	**45.60%**
Italian	3.69%	11.33%	**14.98%**
Romanian	10.42%	17.15%	**20.86%**
English	13.25%	24.24%	**25.67%**
Danish	7.21%	29.36%	**34.59%**
Swedish	2.09%	13.92%	**18.69%**
Icelandic	10.93%	24.56%	**29.98%**
Estonian	31.87%	**44.44%**	32.21%
Finnish	5.40%	11.32%	**12.15%**
Tagalog	10.10%	15.85%	**17.08%**
Swahili	8.63%	10.05%	**11.02%**
Free / S-V-O			
Czech	3.30%	10.85%	**11.02%**
Polish	8.16%	**21.18%**	20.87%
Russian	19.91%	46.38%	**47.35%**
Verb Second (V2)			
German	9.97%	**13.89%**	9.96%
Dutch	11.78%	**17.09%**	15.50%
S-O-V			
Turkish	**52.66%**	46.53%	25.28%
Basque	**25.88%**	20.41%	6.44%

Table 4.15: The effect of window position in Context Similarity illustrating a performance-based preference for window regions correlated with the canonical location of the verb and object.

Language	Left 6x0	5x1	4x2	Center 3x3	2x4	1x5	Right 0x6
S-V-O							
Portuguese	12.05%	21.38%	27.55%	26.58%	29.01%	29.01%	**32.87%**
Estonian	31.87%	41.43%	44.01%	42.20%	44.28%	**44.70%**	32.21%
Free / S-V-O							
Russian	19.91%	41.08%	47.35%	40.91%	**47.90%**	47.39%	47.35%
Verb Second (V2)							
German	9.97%	11.34%	13.09%	**14.78%**	13.78%	13.34%	9.96%
S-O-V							
Turkish	**52.66%**	49.06%	48.15%	44.40%	47.03%	45.41%	25.28%
Basque	**25.88%**	21.35%	21.91%	19.75%	21.66%	19.81%	6.44%

Table 4.16: The effect of fine-grained window position and grammatical word ordering in Context Similarity.

Language	100K	500K	1M	2M	5M	10M	13.4M-15.7M
English	4.86%	5.08%	6.47%	7.85%	8.92%	9.92%	**11.68%**
French	4.03%	4.89%	5.66%	7.00%	8.76%	11.24%	**17.72%**
Spanish	3.37%	3.72%	4.78%	6.38%	**9.69%**		

Table 4.17: Top-1 performance accuracy relative to corpus size using a 5x5 weighted window measured against only those inflections found in the corpus. English was tested on a maximum of 13.4 million words, French on 15.7 million words.

to higher overall accuracies as the corpus size increases since there are more inflections for which the context similarity model can attempt to find roots (Figure 4.9). However, not all of the increase in performance is due to this increase in the number of inflections found in the corpus. Figure 4.10 presents the Top-10 accuracy relative to the number of inflections present in the corpus which shows, not surprisingly, that more data leads to better estimation of contexts,

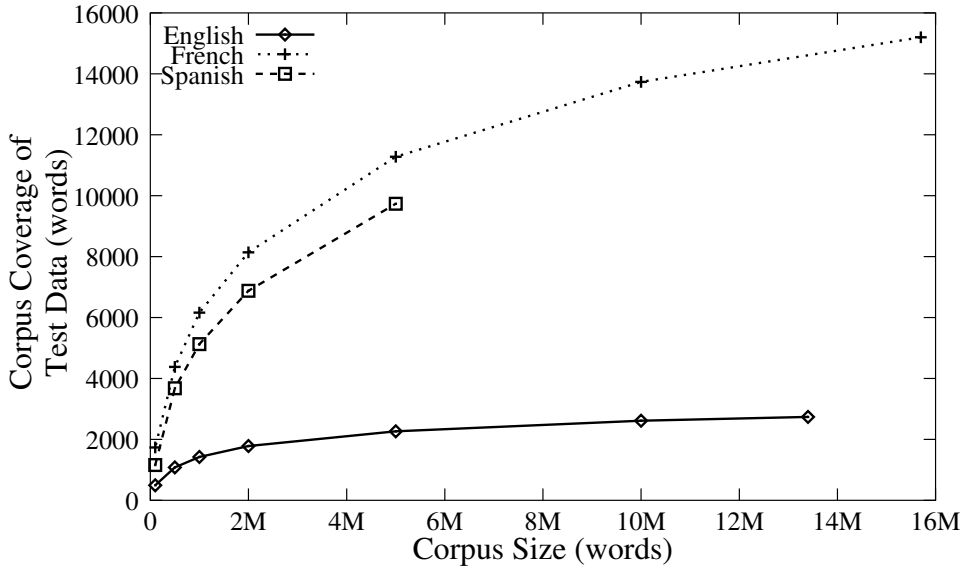


Figure 4.8: Number of inflections from the test set (by type) found in the corpus relative to the size of the corpus

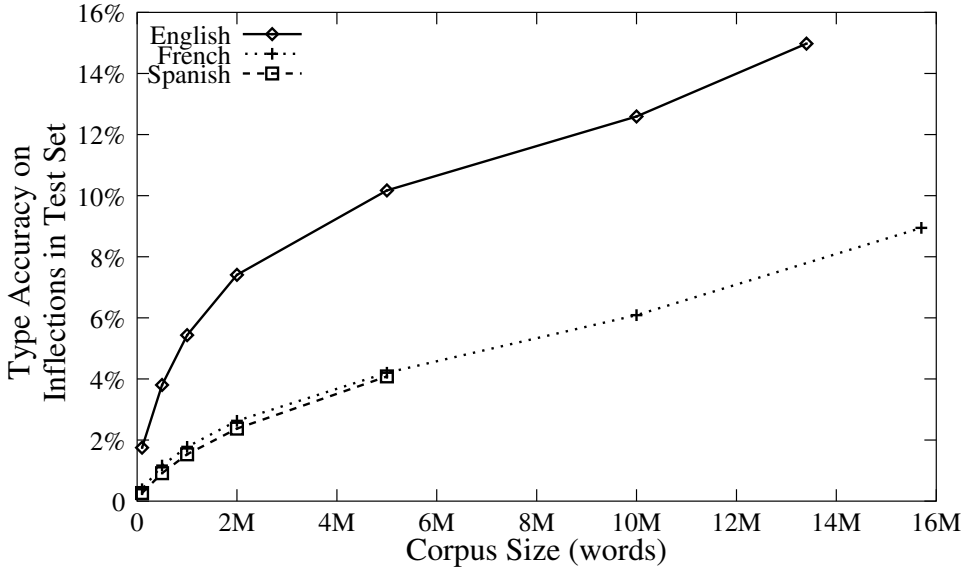


Figure 4.9: Top-10 precision of Context Similarity (5x5 weighted window) measured against all inflections in the test set.

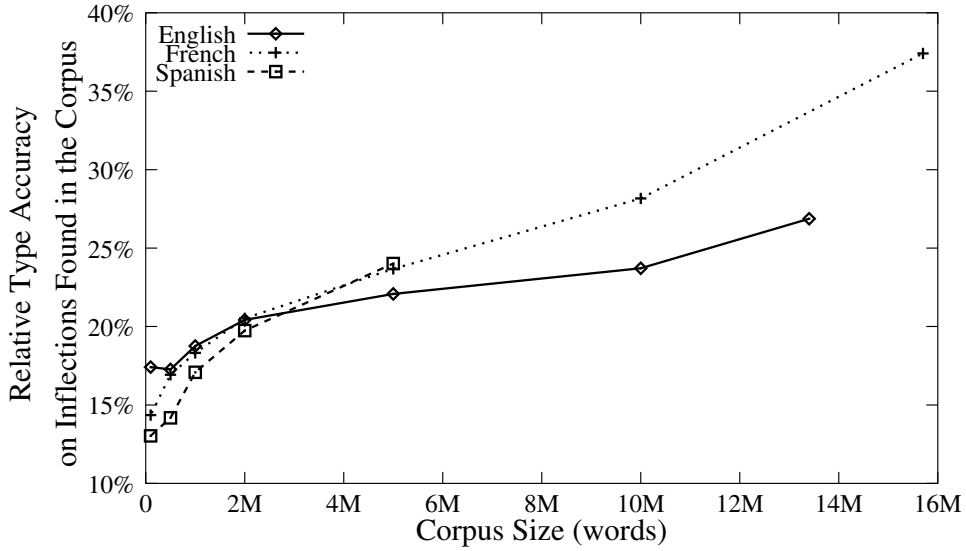


Figure 4.10: Top-10 precision of Context Similarity (5x5 weighted window) measured only against those inflections found in the corpus, as corpus size increases.

4.5 Lemma Alignment by Weighted Levenshtein Distance

4.5.1 Initializing transition cost functions

The third alignment similarity function considers overall string edit distance using a weighted Levenshtein measure. The cost matrix used here, and presented in Table 4.18, treats clusters of vowels and consonants as units of type V+ or C+ (one or more Vowel/Consonant) with initial distance costs δ_0 = IDENTITY, δ_1 = SUB(V+,V+), δ_2 = SUB(C+,C+), δ_3 = SUB(C+,V+) or SUB(V+,C+), δ_4 = INS/DEL(V+), and δ_5 = INS/DEL(C+). Additionally, since the Unicode Data Consortium provides cross-lingual information on removing the diacritics from base letters, the cost of adding or removing an accent from a letter can be modeled separately as δ_6 = INS/DEL(ACCENT) or SUB(ACCENT,

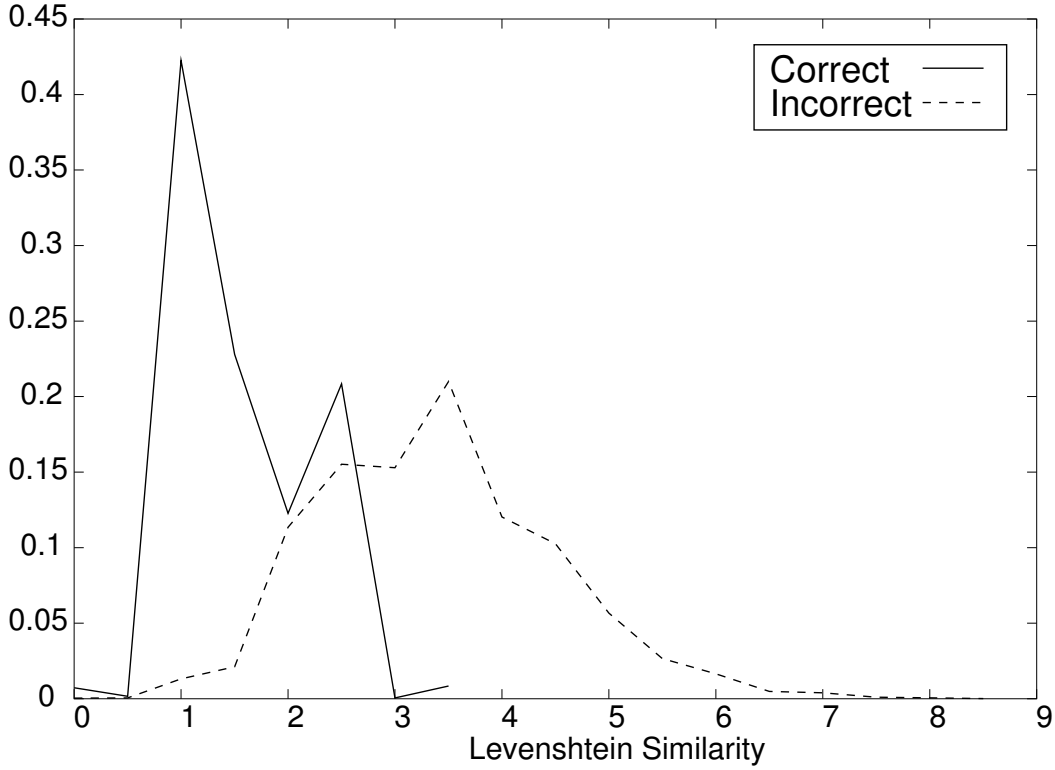


Figure 4.11: Levenshtein similarity distributions for correctly and incorrectly aligned inflection-root pairs in French. The distribution for the incorrectly aligned pairs was obtained by aligning an inflection with every root that was not its root. Both distributions have been normalized to sum to 1.

ACCENT).[16]

This cost matrix configuration is a reflection of morphological systems worldwide, where vowels and vowel clusters are relatively mutable through morphological processes, while consonants in general tend to have a lower probability of change during inflection.

Initial values for δ_0-δ_6, shown in Table 4.19, are a relatively arbitrary assignment

[16]Note that it would have been possible to model the substitution of accents for vowels or consonants; however, there was no language for which this seemed to be a plausible phenomenon. Changes such as SUB(é, ee) would have been modeled as SUB(´,e), an unlikely substitution, rather than the far more likely explanation, SUB(é, ee), effectively modeled as SUB(V+,V+).

	a	o	ue	m	n	mm	...	ACCENT	DELETE
a	δ_0	δ_1	δ_1	δ_3	δ_3	δ_3	...		δ_4
o	δ_1	δ_0	δ_1	δ_3	δ_3	δ_3	...		δ_4
ue	δ_1	δ_1	δ_0	δ_3	δ_3	δ_3	...		δ_4
m	δ_3	δ_3	δ_3	δ_0	δ_2	δ_2	...		δ_5
n	δ_3	δ_3	δ_3	δ_2	δ_0	δ_2	...		δ_5
mm	δ_3	δ_3	δ_3	δ_2	δ_2	δ_0	...		δ_5
...	...	...	...	...	...		...		...
ACCENT									δ_6
INSERT	δ_4	δ_4	δ_4	δ_5	δ_5	δ_5	...	δ_6	

Table 4.18: Initial transition cost matrix layout reflecting an intial bias toward equally weighted vowel-to-vowel transformations, equally weighted consonant-to-consonant transformations, and equally weighted vowel-to-consonant (and vice versa) transformations.

variable	value	description	
δ_0	0.0	IDENTITY	all identity substitutions
δ_1	0.2	SUB(V+,V+)	substitute V+ for V+
δ_2	1.0	SUB(C+,C+)	substitute C+ for C+
δ_3	1.3	SUB(C+,V+), SUB(V+,C+)	substitute C+ for V+ (or vice versa)
δ_4	0.3	INS(V+), DEL(V+)	insert or delete a V+
δ_5	1.0	INS(C+), DEL(C+)	insert or delete a C+
δ_6	0.1	INS(ACCENT), DEL(ACCENT)	insert or delete an accent

Table 4.19: Variable descriptions for initial transition cost matrix shown in Table 4.18. These initial values reflect the tendency for vowels to be relatively mutable, consonants to be relatively immutable, and to accent substitution to be a relatively low-cost operation.

variable	baseline	acl-2000	2Δ	equal	$\frac{1}{2}\Delta$	swap
δ_0	0.0	0.0	0	0	0	0.0
δ_1	0.2	0.3	1	1	0.5	1.0
δ_2	1.0	0.95	1	1	0.5	0.2
δ_3	1.3	1.0	1	1	0.5	1.3
δ_4	0.3	0.7	0.5	1	1	1
δ_5	1.0	0.95	0.5	1	1	0.3
δ_6	0.1	0.1	0.5	1	1	0.1

Table 4.20: Initial transition cost matrices. The *baseline* transition costs were selected to reflect the tendency for vowel transformations to be lower cost than consonant transformations. The *acl-2000* costs were the initial costs associated with the Levenshtein distance measure as presented in [Yarowsky and Wicentowski, 2000]. The 2Δ and $\frac{1}{2}\Delta$ matrices treat all substitutions as having twice (or half) the cost of insertions and deletions. The *equal* transition cost matrix treats all insertions, deletions and substitutions equally. To highlight the intuition behind the *baseline* model, the *swap* matrix has swapped the costs of the vowel transformations with the consonant transformations.

reflecting this tendency. However, as subsequent algorithm iterations proceed, this matrix is re-estimated with empirically observed character-to-character stem-change probabilities from the algorithm's current best weighted alignments. More optimally, the initial state of this matrix could be seeded with values partially borrowed from previously trained matrices from other related languages.[17]

While it may seem beneficial to set the initial distances to be partially sensitive to phonological similarities (with SUB(/*d*/,/*t*/) << SUB(/*d*/,/*f*/) for example), doing so requires additional information about the language, and is particularly sensitive to the nature and faithfulness of the mapping between orthography and phonology. Since these types of distinctions readily emerge after iterative re-estimation from the baseline model shown in Table 4.18, results for seeding the cost matrix in this fashion are not presented.

[17]Chapter 5 will present results on choosing a transition cost matrix through iterative retraining.

Language	baseline	acl2k	2Δ	equal	$\frac{1}{2}\Delta$	swap
Basque	73.91%	71.40%	**92.47%**	92.25%	62.80%	73.90%
Catalan	88.02%	86.84%	86.94%	**90.21%**	77.47%	68.54%
Czech	78.26%	79.00%	78.24%	**88.79%**	78.27%	56.14%
Danish	90.94%	92.86%	95.39%	**96.65%**	85.42%	67.74%
Dutch	78.08%	76.76%	**86.06%**	83.96%	66.42%	54.68%
English	84.80%	85.15%	**95.84%**	95.79%	70.02%	65.11%
Estonian	78.52%	81.04%	90.32%	**92.06%**	78.24%	74.90%
Finnish	62.88%	64.17%	79.89%	**82.38%**	52.05%	51.92%
French	88.09%	89.07%	94.97%	**95.39%**	83.37%	81.29%
German	91.44%	91.36%	92.02%	**94.26%**	85.25%	73.34%
Greek	96.35%	95.31%	95.31%	**97.92%**	90.10%	92.71%
Hindi	**96.48%**	91.41%	91.41%	93.36%	82.81%	83.98%
Icelandic	88.65%	90.95%	88.49%	**92.33%**	82.75%	79.01%
Irish	87.75%	87.75%	94.18%	**94.70%**	65.73%	70.65%
Italian	91.79%	90.09%	96.31%	**97.42%**	81.52%	65.12%
Klingon	98.74%	97.84%	**99.50%**	99.44%	66.23%	43.78%
Latin	70.60%	68.46%	**86.24%**	83.97%	49.91%	53.98%
Norwegian	91.40%	95.04%	95.65%	**96.62%**	91.45%	69.65%
Occitan	88.63%	88.50%	85.32%	**90.72%**	76.59%	65.05%
Polish	91.71%	88.68%	92.70%	**95.01%**	81.50%	69.69%
Portuguese	93.74%	92.38%	96.93%	**97.78%**	83.53%	78.41%
Romanian	82.67%	81.20%	**93.25%**	90.79%	67.87%	64.02%
Russian	80.87%	83.44%	85.84%	**89.66%**	78.10%	67.96%
Sanskrit	67.43%	62.50%	**79.53%**	77.29%	39.74%	24.15%
Spanish	90.52%	89.67%	92.04%	**94.13%**	77.49%	68.32%
Swahili	68.38%	58.24%	**89.42%**	84.10%	37.32%	43.45%
Swedish	82.74%	87.28%	94.18%	**95.87%**	83.11%	75.81%
Tagalog	61.03%	51.11%	**85.31%**	72.88%	16.88%	11.81%
Tamil	83.42%	82.24%	**97.99%**	97.32%	83.08%	83.58%
Turkish	89.63%	85.48%	**96.79%**	94.90%	58.15%	58.31%
Uzbek	81.02%	77.90%	87.31%	**91.99%**	60.03%	66.25%
Welsh	76.50%	80.44%	90.45%	**91.05%**	66.95%	67.28%

Table 4.21: Levenshtein performance based on initial transition cost

Language	baseline	acl2k	2Δ	equal	$\frac{1}{2}\Delta$	swap
Danish	71.22%	74.15%	58.54%	**79.51%**	69.27%	14.63%
Dutch	48.28%	48.03%	**55.17%**	50.99%	32.76%	7.14%
English	**59.38%**	53.12%	53.12%	**59.38%**	53.12%	0.00%
French	45.53%	50.75%	59.86%	**69.48%**	48.65%	35.90%
German	77.27%	77.50%	62.43%	**78.83%**	63.18%	8.29%
Irish	24.19%	24.19%	**58.06%**	45.16%	12.90%	9.68%
Romanian	52.11%	50.66%	59.16%	**64.64%**	43.80%	35.06%
Turkish	18.82%	11.29%	11.83%	**30.11%**	9.14%	8.60%
Welsh	22.57%	22.76%	24.63%	**33.21%**	21.46%	17.54%

Table 4.22: Levenshtein performance on irregular verbs based on initial transition cost

Language	baseline	acl2k	2Δ	equal	$\frac{1}{2}\Delta$	swap
Danish	61.82%	61.82%	**96.36%**	95.00%	25.00%	45.91%
Dutch	70.70%	71.69%	**85.43%**	84.44%	59.69%	40.56%
English	76.05%	81.91%	84.92%	**89.95%**	59.63%	52.09%
French	86.13%	87.89%	**96.76%**	96.40%	78.66%	83.03%
German	89.19%	83.17%	87.28%	**93.84%**	82.22%	70.72%
Irish	89.52%	91.33%	94.10%	**95.66%**	67.59%	71.20%
Romanian	**85.43%**	82.61%	83.70%	83.59%	56.20%	51.30%
Turkish	85.66%	86.34%	91.36%	**95.84%**	68.41%	67.70%
Welsh	74.24%	75.76%	83.22%	**88.98%**	65.42%	54.83%

Table 4.23: Levenshtein performance on semi-regular verbs based on initial transition cost

Language	baseline	acl2k	2Δ	equal	$\frac{1}{2}\Delta$	swap
Danish	93.72%	95.69%	97.34%	**97.69%**	89.84%	71.91%
Dutch	84.73%	82.39%	**90.26%**	88.02%	73.32%	66.17%
English	86.76%	86.14%	**98.41%**	97.30%	72.24%	68.34%
French	91.22%	91.87%	**97.45%**	97.24%	86.07%	84.47%
German	93.98%	94.28%	**97.36%**	96.92%	89.24%	84.48%
Irish	93.49%	90.00%	99.53%	**100.00%**	69.77%	78.37%
Romanian	85.09%	83.67%	**96.35%**	93.20%	70.37%	66.92%
Turkish	90.56%	85.91%	**97.98%**	95.26%	57.40%	57.66%
Welsh	77.23%	81.27%	91.45%	**91.81%**	67.56%	68.22%

Table 4.24: Levenshtein performance on regular verbs based on initial transition cost

4.5.2 Using positionally weighted costs

One problem with using the Levenshtein distance measure to align inflection and root pairs is that the standard Levenshtein formulation does not take into account the position of the observed change when determining the final distance. This means that word initial changes are as costly as word final changes which, in general, is not a good model for languages which primarily use only prefixation or suffixation for inflection.

To compensate for this shortcoming, the Levenshtein model has been enhanced with a position-senstitive weighting scheme. For each insertion, deletion or substitution which occurs at position i of the inflection[18], the cost of the change, c is computed as

$$weightedCost(c) = (1 + penalty(i)) * cost(c) \tag{4.2}$$

where $cost(c)$ is the cost of the change c as defined in the transition cost matrix, and $penalty(i)$ is a function of the location in the inflection where the change occurs. For prefix penalization (favoring changes at the end of the word), $penalty(i)$ is computed as

$prefixPenalty(i) = p * (length(\text{inflection}) - i)$

where p is a user-supplied or iteratively-determined parameterization to the *prefixPenalty* function.

Table 4.25 shows the results of using small prefix and suffix penalties. Of particular note is the fact that the prefixal languages of Swahili, Tagalog and Klingon do best using a suffix penalty rather than a prefix penalty. For nearly all other languages (with the exception of Czech), the use of a small prefix penalty outpeforms the baseline system which

[18]Or the root if this is being used for generation.

does not use such penalties.

Table 4.26 shows the performance of the weighted Levenshtein model as values of the penalty penalty are increased. For languages which are predominantly suffixal but use prefixation as well, such as German, Polish and Irish, low prefix penalties are most effective. For languages which are solely suffixal, high prefix penalties are most effective. Interestingly, Czech, which preferred no prefix penalty to a small penalty (as presented in Table 4.25) shows preference for the largest prefix penalty. Table 5.10 will show how these penalties can be iteratively learned.

4.5.3 Segmenting the strings

As presented in Section 4.5.1, the transition cost matrix contains costs for transformations of both single letters as well as costs for transformations of clusters of vowels and clusters of consonants.

This is motivated by the fact that the orthographic rendering of underlying phonological vowels is often not a one-to-one mapping. For example, the English long *e* is often rendered as *ee*, as in *keep*, whereas the short vowel *e* is rendered as simply *e*, as in *kept*. In to effectively model the transition from *e* in the inflection to *ee* in the root, the inflection and root must be segmented such that consecutive vowels are considered single segments. Consonant doubling in English past tense presents a rationale for segmenting consecutive consonants as well as vowels.

Table 4.28 presents results for using various segmenting techniques which are defined in Table 4.27. By a small majority (17 out of 32), split method 2, the method where there were *no* clusters, was the preferred method. By default, the split method used in all

	Prefix Penalty			No	Suffix Penalty		
Language	1	0.5	0.25	Penalty	0.25	0.5	1.0
Spanish	**94.13%**	93.88%	93.36%	91.62%	63.38%	69.64%	76.08%
Portuguese	**96.26%**	96.01%	95.38%	93.74%	71.79%	76.24%	80.89%
Catalan	**92.01%**	91.37%	90.51%	88.02%	71.84%	74.96%	79.17%
Occitan	**92.64%**	92.40%	91.98%	88.63%	67.28%	71.21%	75.14%
French	**93.44%**	92.80%	91.78%	88.09%	69.17%	72.05%	76.45%
Italian	**95.26%**	94.86%	94.37%	91.79%	71.58%	75.70%	80.74%
Romanian	**91.14%**	90.48%	89.34%	82.67%	46.08%	51.93%	58.64%
Latin	**85.35%**	84.67%	82.89%	70.60%	21.66%	26.47%	34.02%
English	**93.36%**	92.38%	89.82%	84.80%	46.96%	54.28%	61.76%
Danish	**94.77%**	93.31%	92.69%	90.94%	70.44%	76.80%	81.00%
Norwegian	**94.47%**	93.81%	93.14%	91.40%	72.42%	78.51%	82.96%
Swedish	**90.24%**	88.93%	87.15%	82.74%	42.88%	51.00%	59.83%
Icelandic	**91.33%**	90.95%	90.30%	88.65%	62.38%	67.55%	74.11%
Hindi	**96.88%**	96.48%	**96.88%**	96.48%	87.50%	87.50%	88.67%
Sanskrit	**78.34%**	77.39%	75.75%	67.43%	29.68%	34.41%	41.58%
Estonian	**81.86%**	81.20%	80.70%	78.52%	62.59%	65.71%	69.17%
Tamil	**89.61%**	87.60%	87.27%	83.42%	62.31%	69.01%	74.54%
Finnish	**74.86%**	73.57%	71.88%	62.88%	27.35%	32.43%	39.52%
Turkish	**95.03%**	94.69%	94.09%	89.63%	48.85%	55.83%	64.61%
Uzbek	**84.67%**	84.43%	83.89%	81.02%	51.19%	55.21%	60.40%
Basque	**80.74%**	79.85%	78.69%	73.91%	38.45%	44.21%	49.47%
Czech	76.49%	76.40%	76.42%	**78.26%**	67.13%	70.13%	72.79%
Polish	**93.22%**	93.02%	92.78%	91.71%	68.54%	73.57%	78.83%
Russian	**84.73%**	83.41%	82.12%	80.87%	66.33%	69.59%	72.96%
Greek	**97.92%**	97.40%	97.40%	96.35%	69.27%	73.44%	81.25%
German	**91.58%**	91.53%	91.39%	91.44%	82.02%	84.04%	86.34%
Dutch	**80.49%**	80.11%	80.22%	78.08%	69.56%	71.59%	73.79%
Irish	**92.89%**	**92.89%**	92.21%	87.75%	48.71%	54.16%	62.18%
Welsh	**85.99%**	84.81%	83.52%	76.50%	35.41%	42.08%	50.69%
Tagalog	20.27%	23.80%	28.35%	61.03%	72.24%	**72.32%**	71.40%
Swahili	29.83%	34.94%	41.69%	68.38%	**79.95%**	79.35%	78.31%
Klingon	24.84%	27.34%	30.46%	98.74%	**99.55%**	99.17%	99.08%

Table 4.25: Levenshtein performance using suffix and prefix penalties

Language	Prefix Penalty							No Penalty
	20	10	7.5	5	3	2	1	
Spanish	94.21%	94.22%	**94.22%**	94.17%	94.17%	94.13%	94.13%	91.62%
Portuguese	96.29%	96.31%	96.32%	96.33%	**96.35%**	96.29%	96.26%	93.74%
Catalan	90.85%	91.98%	92.06%	**92.13%**	92.06%	92.11%	92.01%	88.02%
Occitan	92.67%	92.65%	92.68%	92.72%	**92.73%**	**92.73%**	92.64%	88.63%
French	93.84%	**93.84%**	93.80%	93.83%	93.74%	93.67%	93.44%	88.09%
Italian	95.39%	95.39%	95.37%	95.46%	**95.49%**	95.35%	95.26%	91.79%
Romanian	**91.76%**	91.75%	91.74%	91.72%	91.67%	91.47%	91.14%	82.67%
Latin	**86.12%**	**86.12%**	86.08%	86.00%	85.92%	85.74%	85.35%	70.60%
English	**94.44%**	**94.44%**	**94.44%**	94.41%	94.33%	94.19%	93.36%	84.80%
Danish	**94.98%**	**94.98%**	94.96%	94.96%	94.74%	94.70%	94.77%	90.94%
Norwegian	**94.83%**	**94.83%**	**94.83%**	**94.83%**	94.47%	94.47%	94.47%	91.40%
Swedish	**91.13%**	**91.13%**	91.11%	91.02%	90.78%	90.75%	90.24%	82.74%
Icelandic	91.33%	91.33%	91.33%	91.33%	**91.47%**	**91.47%**	91.33%	88.65%
Hindi	**96.88%**	**96.88%**	**96.88%**	**96.88%**	**96.88%**	**96.88%**	**96.88%**	96.48%
Sanskrit	**79.13%**	**79.13%**	79.03%	78.98%	78.93%	78.83%	78.34%	67.43%
Estonian	81.71%	81.69%	81.68%	81.84%	81.78%	81.76%	**81.86%**	78.52%
Tamil	**89.61%**	**89.61%**	**89.61%**	**89.61%**	**89.61%**	**89.61%**	**89.61%**	83.42%
Finnish	**75.66%**	75.65%	75.63%	75.59%	75.39%	75.29%	74.86%	62.88%
Turkish	95.20%	95.19%	95.16%	95.14%	**95.31%**	95.10%	95.03%	89.63%
Uzbek	83.88%	83.90%	83.92%	83.99%	84.17%	84.36%	**84.67%**	81.02%
Basque	79.19%	**83.74%**	82.69%	81.82%	81.32%	81.07%	80.74%	73.91%
Czech	**78.73%**	76.23%	76.46%	76.43%	76.55%	76.54%	76.49%	78.26%
Polish	93.10%	93.12%	93.13%	93.12%	93.16%	93.18%	**93.22%**	91.71%
Russian	**85.39%**	85.35%	85.35%	85.18%	85.07%	85.07%	84.73%	80.87%
Greek	**97.92%**	**97.92%**	**97.92%**	**97.92%**	**97.92%**	**97.92%**	**97.92%**	96.35%
German	91.34%	91.34%	91.34%	91.38%	91.38%	91.41%	**91.58%**	91.44%
Dutch	81.14%	81.14%	81.18%	**81.20%**	80.91%	80.85%	80.49%	78.08%
Irish	92.74%	92.74%	92.74%	92.59%	92.44%	92.66%	**92.89%**	87.75%
Welsh	**87.22%**	87.21%	87.20%	87.18%	86.79%	86.69%	85.99%	76.50%
Tagalog	16.52%	16.62%	16.67%	17.35%	17.69%	18.25%	20.27%	**61.03%**
Swahili	24.03%	24.24%	24.38%	24.80%	25.78%	26.75%	29.83%	**68.38%**
Klingon	11.99%	12.02%	12.02%	13.07%	13.32%	14.22%	24.84%	**98.74%**

Table 4.26: Levenshtein performance using prefix penalties

Split 0	Cluster consecutive vowels, but not consecutive consonants Example: d / o / p / ou / š / t / ě / l / i
Split 1	Cluster consecutive vowels and consonants Example: d / o / p / ou / št / ě / l / i
Split 2	Do not cluster consecutive vowels or consonants Example: d / o / p / o / u / š / t / ě / l / i
Split 3	Split 0, except with accents separated from the underlying letter Example: d / o / p / ou / s$^{\check{}}$ / t / e$^{\check{}}$ / l / i
Split 4	Split 1, except with accents separated Example: d / o / p / ou / s$^{\check{}}$t / e$^{\check{}}$ / l / i
Split 5	Split 2, except with accents separated Example: d / o / p / o / u / s$^{\check{}}$ / t / e$^{\check{}}$ / l / i

Table 4.27: Definitions of the 6 split methods presented in Table 4.28

of the experiments, unless otherwise stated, was split method 3, based on initial intuitions about the potential success of clustering vowels. Table 5.9 will present results on iteratively learning this parameter.

4.6 Translingual Bridge Similarity

4.6.1 Introduction

All of the previously presented similarity measures are effective as unsupervised alignment models. These similarity measures produce significant noise; yet, all are capable of inducing inflection-root mappings using no training data. This is particularly important for resource-poor languages where training data is not available. For all of the languages presented in this thesis, however, significant amounts of training data are available. While these languages were chosen for their broad range of morphological phenomenon, they were also chosen because the available training data allowed for a thorough evaluation of all the

	Accuracy					
Language	Split 0	Split 1	Split 2	Split 3	Split 4	Split 5
Spanish	89.98%	90.34%	91.47%	90.14%	90.57%	**91.58%**
Portuguese	92.78%	93.53%	94.24%	93.06%	93.68%	**94.54%**
Catalan	86.46%	86.86%	85.35%	85.71%	**87.20%**	85.06%
Occitan	87.90%	87.70%	**89.61%**	86.87%	86.35%	87.77%
French	88.08%	89.20%	92.88%	88.39%	89.40%	**93.11%**
Italian	92.24%	**92.81%**	0.06%	0.07%	55.39%	0.06%
Romanian	87.68%	**88.34%**	87.58%	80.00%	80.86%	78.30%
Latin	75.51%	76.94%	75.52%	77.05%	**80.59%**	75.91%
English	82.60%	83.45%	**92.18%**	82.60%	83.45%	**92.18%**
German	90.54%	**91.98%**	73.15%	90.32%	91.75%	72.70%
Dutch	73.86%	**79.43%**	49.62%	73.88%	**79.43%**	49.62%
Danish	92.33%	**92.42%**	91.56%	91.45%	91.54%	90.60%
Norwegian	90.01%	**90.32%**	89.63%	89.28%	89.59%	88.34%
Swedish	86.31%	86.41%	**90.05%**	83.00%	83.07%	88.22%
Icelandic	87.92%	**88.22%**	84.03%	87.49%	87.70%	83.59%
Czech	**78.53%**	77.19%	76.22%	77.56%	76.62%	75.84%
Polish	88.27%	**89.22%**	88.07%	68.25%	58.38%	57.50%
Russian	**90.42%**	**90.42%**	**90.42%**	89.34%	89.34%	89.34%
Irish	83.05%	84.56%	80.04%	**86.73%**	86.49%	79.92%
Welsh	79.64%	**82.21%**	81.54%	79.51%	82.20%	81.46%
Greek	92.71%	92.71%	92.71%	**93.23%**	**93.23%**	**93.23%**
Hindi	**93.52%**	**93.52%**	90.99%	91.55%	91.83%	86.76%
Sanskrit	74.27%	73.05%	75.01%	75.20%	73.31%	**75.62%**
Estonian	78.34%	**80.46%**	74.57%	78.09%	79.83%	74.57%
Finnish	65.25%	**70.47%**	64.68%	61.07%	68.28%	60.81%
Turkish	90.15%	**90.20%**	81.78%	85.60%	85.87%	75.43%
Uzbek	81.09%	**84.28%**	73.46%	81.09%	**84.28%**	73.46%
Tamil	92.82%	92.82%	**93.32%**	90.48%	90.48%	90.32%
Basque	72.48%	72.88%	**75.12%**	72.48%	72.88%	**75.12%**
Tagalog	51.62%	53.84%	**54.31%**	51.62%	53.84%	**54.31%**
Swahili	64.47%	**66.59%**	64.36%	64.47%	**66.59%**	64.36%
Klingon	84.02%	**84.17%**	82.73%	84.02%	**84.17%**	82.73%

Table 4.28: Stand-alone Levenshtein performances. Split 0 has clusters of vowels, split 1 has clusters of vowels and consonants, split 2 has no clusters. Splits 3, 4, 5 are the same as 0, 1, 2 respectively, except that accents are separated from the letters.

measures presented.

For the large majority of the world's 200+ major languages, there is a shortage or absence of annotated training data; however, there is small subset of widely spoken languages with extremely rich resources, including English and French. The Translingual Bridge similarity function aims to leverage the major investments in annotated data and tools for these resource-rich languages to overcome the annotated resource shortage in other languages.

In this section, existing tools for English, including morphological analyzers and part of speech taggers, are applied to word-aligned bilingual text corpora and their output projected onto the second language. Simple, direct projection is quite noisy, however, even with optimal alignments.

As will be seen in Chapter 5, the noisy and incomplete initial inflection-root pairs produced by this projection can be used to accurately bootstrap the supervised morphological analyzer: the induced morphological analyzer achieves over 99% lemmatization accuracy on the complete, highly inflected French verbal system.

While this projection work does rely heavily on existing tools and resources in English, it requires no hand-annotated training data in the target language, no language-specific knowledge and no resources beyond raw text.

In addition to its success in inducing morphological analyzers, Yarowsky et al. [2001] uses this approach to induce part-of-speech taggers, base noun phrase bracketers, and named entity taggers.

4.6.2 Background

Previous research on the word alignment of parallel corpora has tended to focus on their use in translation model training for MT rather on than monolingual applications. One exception is bilingual parsing. Wu [1995], Wu [1997] investigated the use of concurrent parsing of parallel corpora in a transduction inversion framework, helping to resolve attachment ambiguities in one language by the coupled parsing state in the second language. Jones and Havrilla [1998] utilized similar joint parsing techniques (twisted-pair grammars) for word reordering in target language generation.

However, with these exceptions in the field of parsing, no one has previously used knowledge projection via aligned bilingual corpora to induce traditional stand-alone monolingual text analyzers in other languages. Thus, the proposed projection and induction methods, and their application to morphological analysis induction, appears to be highly novel.

4.6.3 Data Resources

The data sets used in these experiments included the English-French Canadian Hansards, the parallel Czech-English Reader's Digest collection, and multiple versions of the Bible including the French Douay-Rheims Bible, Spanish Reina Valera Bible, and three English Bible Versions (King James, New International and Revised Standard). All corpora were automatically word-aligned by the publicly available EGYPT system Al-Onaizan et al. [1999] which is based on IBM's Model 3 statistical MT formalism Brown et al. [1990]. The word-alignment utilized a strictly raw-word-based model with *no* use of morphological

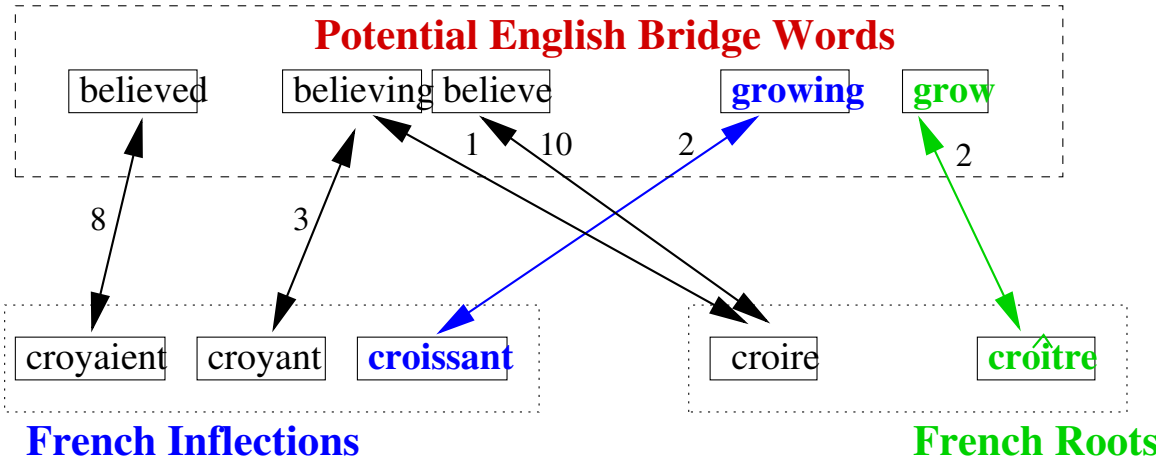


Figure 4.12: Direct morphological alignment between French and English

analysis, or other dictionary resources.

4.6.4 Morphological Analysis Induction

As illustrated in Figure 4.12, the association between a French verbal inflection (*croyant*) and its correct root (*croire*), rather than a similar competitor (*croître*), can be identified by a single-step transitive association via an English bridge word (*believing*). This direct alignment requires only that the corpora are word aligned and does not make any assumptions about the availability of NLP resources of, in this case, English.

However, such direct associations are relatively rare given that inflections in a one language tend to associate with similar inflections in the second language. So, while the infinitive forms tend to associate with analogous infinitive forms, and the present tense forms tend to associate with analogous present tense forms, there are few examples where infinitive forms have been aligned directly with present tense forms. Thus *croyaient* (*believed*) and its root *croire* have no direct English link in the aligned corpus.

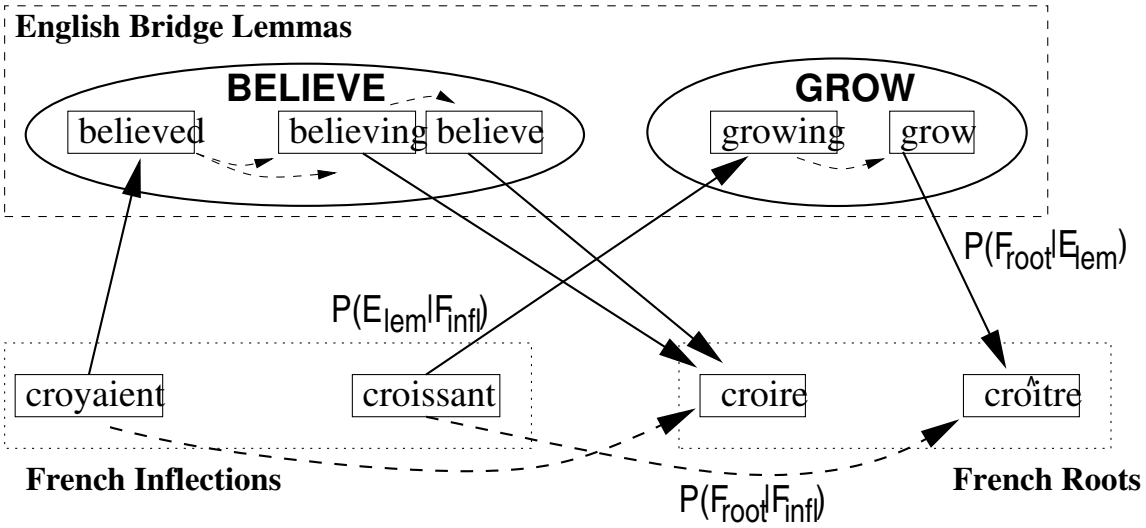


Figure 4.13: French morphological analysis via English

However, Figure 4.13 illustrates that an existing investment in a lemmatizer for English can help bridge this gap by creating the multi-step transitive association *croyaient* → *believed* → *believe* → *croire*. Figure 4.14 illustrates how this transitive linkage via English lemmatization can be potentially utilized for all other English lemmas (such as THINK) with which *croyaient* and *croire* also associate.

Formally, these multiple transitive linkages can be modeled as shown in (4.3), by summing over all English lemmas (E_{lem_i}) with which either a candidate French inflection (F_{infl}) or its root (F_{root}) exhibit an alignment in the parallel corpus.

$$P_{morph-proj}(F_{root}|F_{infl}) = \sum_i P_{alignment}(F_{root}|E_{lem_i})\, P_{alignment}(E_{lem_i}|F_{infl}) \qquad (4.3)$$

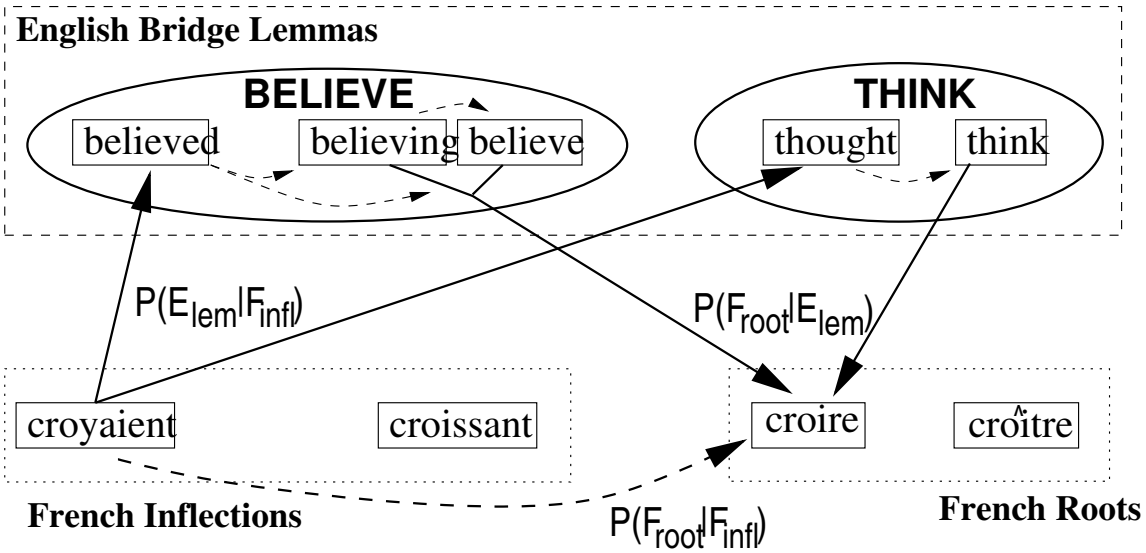


Figure 4.14: Multi-bridge French infl/root alignment

For example, as shown in Figure 4.14:

$$
\begin{aligned}
&P_{mp}(\text{croire}|\text{croyaient}) = \\
&\quad P_a(\text{croire}|\textsc{believe})\, P_a(\textsc{believe}|\text{croyaient}) + \\
&\quad P_a(\text{croire}|\textsc{think})\, P_a(\textsc{think}|\text{croyaient}) + ...
\end{aligned}
\tag{4.4}
$$

This projection-based similarity measure $P_{mp}(F_{root} \mid F_{infl})$ can be quite effective on its own, as shown in the *MProj only* entries in Tables 5.19, 5.17, and 5.18 (for multiple parallel corpora in 3 different languages), especially when restricted to the highest-confidence subset of the vocabulary (5.2% to 77.9% in these data) for which the association exceeds simple fixed probability and frequency thresholds. When estimated from a 1.2 million word subset of the French Hansards, for example, the MProj measure alone achieves 98.5% precision on 32.7% of the inflected French verbs in the corpus (constituting 97.6% of the tokens in the corpus). Unlike traditional string-transduction-based morphology induction meth-

ods where irregular verbs pose the greatest challenges, these typically high-frequency words are often the *best* modeled data in the vocabulary making these multilingual projection techniques a natural complement to existing models.

The high precision on the MProj-covered subset also make these partial pairings effective training data for robust supervised algorithms that can generalize the string transformation behavior to the remaining uncovered vocabulary. These pairings are used in Chapter 5 to bootstrap the supervised systems presented in Chapter 3.

As shown in Table 5.19, by using the projection-based MProj and trie-based supervised models together (with the latter extending coverage to words that may not even appear in the parallel corpus), full verb lemmatization precision on the 1.2M word Hansard subset exceeds 99.5% (by type) and 99.9% (by token) with 95.8% coverage by type and 99.8% coverage by token. The relatively small percentage of cases where MProj and the supervised methods (MTrie) together are not sufficiently confident, using the Levenshtein, context and frequency models can be used as a backoff, bringing the system coverage to 100% with a small drop in precision to 97.9% (by type) and 99.8% (by token) on the unrestricted space of inflected verbs observed in the full French Hansards. As shown in Figure 4.16, performance is strongly correlated with size of the initial aligned bilingual corpus, with a larger Hansard subset of 12M words yielding 99.4% precision (by type) and 99.9% precision (by token).

Morphology Induction via Aligned Bibles

Performance using even small parallel corpora (e.g. a 120K subset of the French Hansards) still yields a respectable 93.2% (type) and 98.9% (token) precision on the verb-

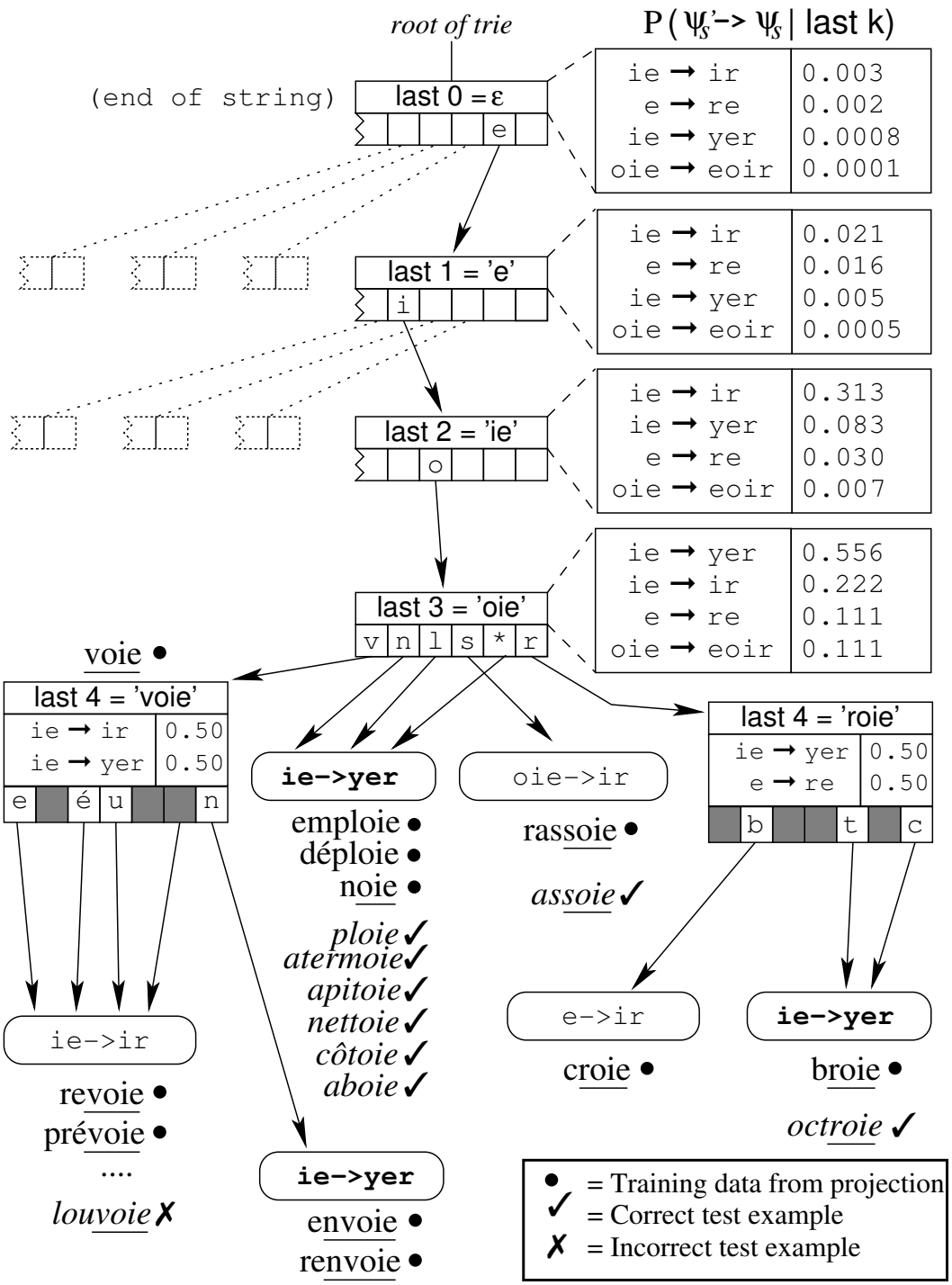


Figure 4.15: The trie data structure, as presented in Chapter 3, trained using pairs derived from the Translingual Bridge algorithm, and tested on unseen pairs. Several similar nodes have been graphically joined to show more detail (such as the children "...évoie" and "...evoie" of "...voie"). However, nodes in the trie are never grouped.

Corpus	Precision		Coverage	
	Typ	Tok	Typ	Tok
French Verbal Morphology Induction				
French Hansards				
· 12M words	.992	.999	.779	.994
· 1.2M words	.985	.998	.327	.976
· 120K words	.962	.931	.095	.901
French Bible (300K words)				
· aligned with 1 English Bible	1.00	1.00	.052	.747
· aligned with 3 English Bibles	.928	.975	.100	.820
Czech Verbal Morphology Induction				
Czech Readers Digest				
· 500K words	.915	.993	.152	.805
SPANISH Verbal Morphology Induction				
Spanish Bible (300K words)				
· aligned with 1 English Bible	.973	.935	.264	.351
· aligned with 1 French Bible	.980	.935	.722	.765
· aligned with 3 English Bibles	.964	.948	.468	.551

Table 4.29: Performance of morphological projection by type and token

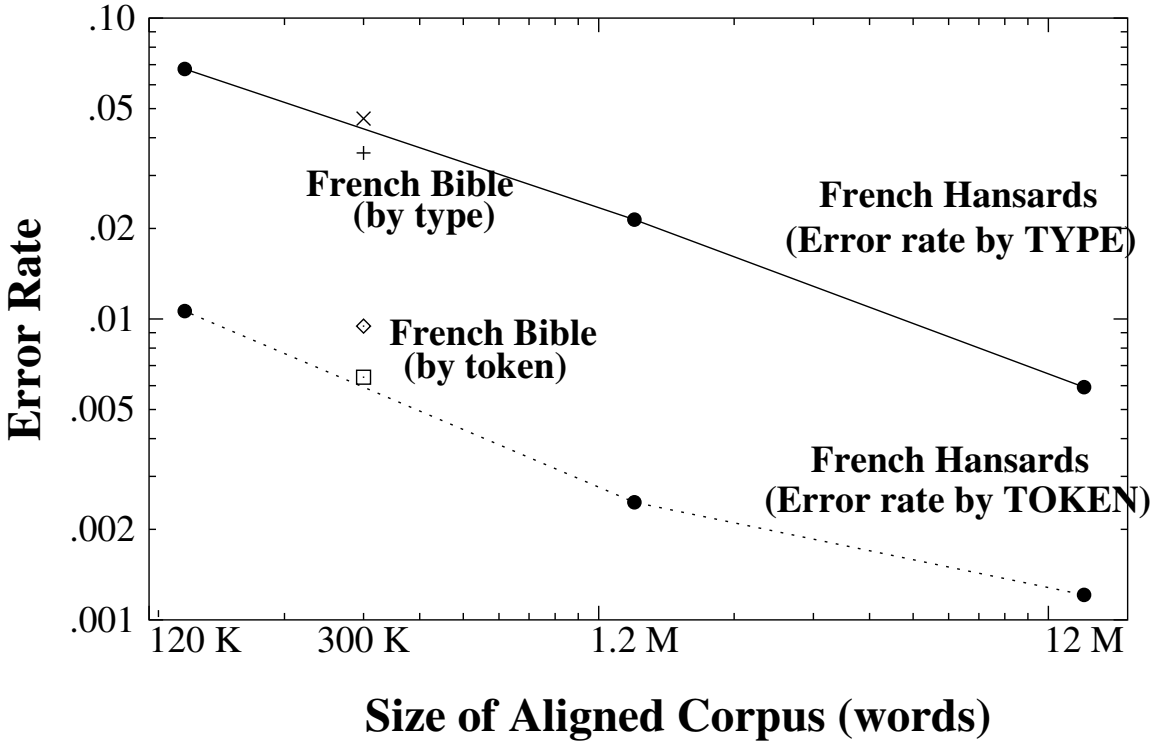


Figure 4.16: Learning Curves for French Morphology

lemmatization test set for the full Hansards. Given that the Bible is actually larger (approximately 300K words, depending on version and language) and available on-line or via OCR for virtually all languages Resnik et al. [2000], results for several experiments on Bible-based morphology induction are also included in Tables 5.19, 5.17, and 5.18.

Boosting Performance via Multiple Parallel Translations

Although one may only have one version of the Bible in a given foreign language, numerous English editions exist and a performance increase can be achieved by simultaneously utilizing alignments to each English version. As illustrated in Figure 9, a single aligned Bible-pair may not exhibit certain transitive bridging links for a given word (due

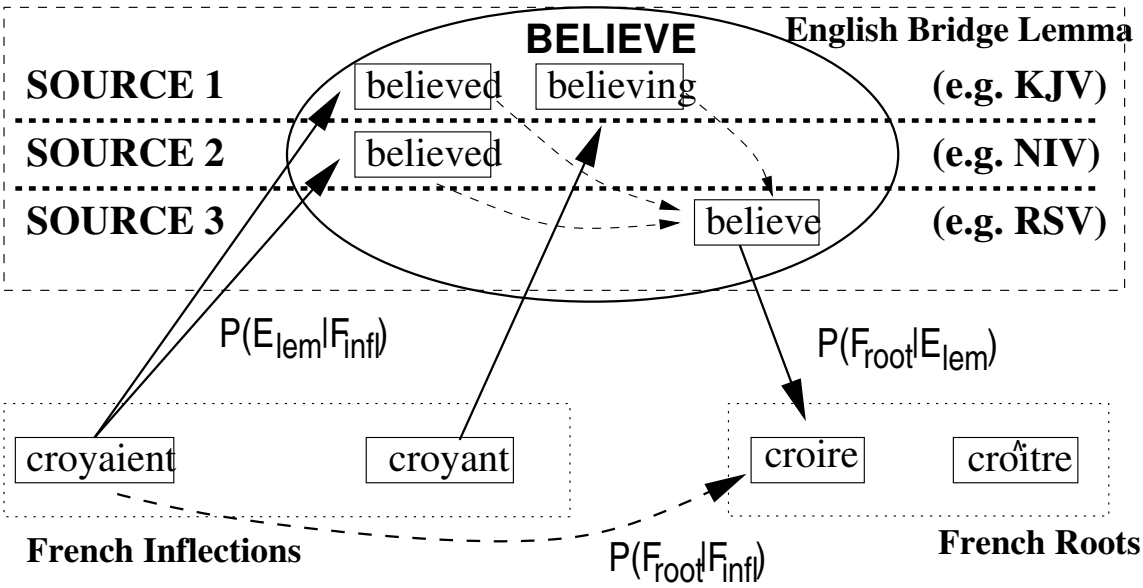


Figure 4.17: Use of multiple parallel Bible translations

both to different lexical usage and poor textual parallelism in some text-regions or version pairs). However, $P_a(F_{root}|E_{lem_i})$ and $P_a(E_{lem_i}|F_{infl})$ need not be estimated from the same Bible pair. Even if one has only 1 Bible in a given source language, each alignment with a distinct English version gives new bridging opportunities with no additional resources on the source language side. The baseline approach (used here) is simply to concatenate the different aligned versions together. While word-pair instances translated the same way in each version will be repeated, this somewhat reasonably reflects the increased confidence in this particular alignment. An alternate model would weight version pairs differently based on the otherwise-measured translation faithfulness and alignment quality between the version pairs. Doing so would help decrease noise. Increasing from 1 to 3 English versions reduces the type error rate (at full coverage) by 22% on French and 28% on Spanish with no increase in the source language resources.

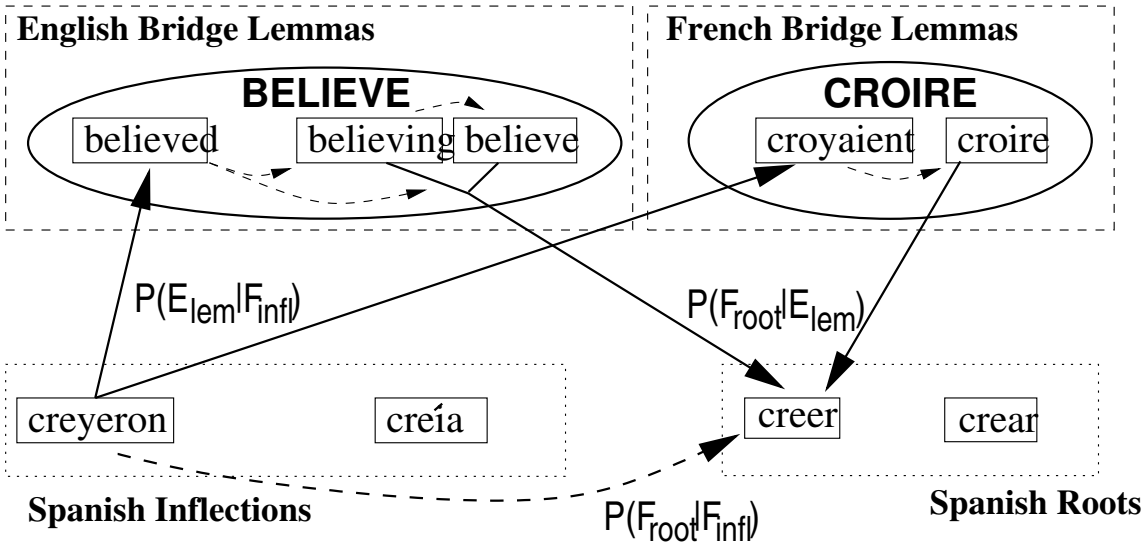


Figure 4.18: Use of bridges in multiple languages.

Boosting Performance via Multiple Bridge Languages

Once lemmatization capabilities have been successfully projected to a new language (such as French), this language can then serve as an additional bridging source for morphology induction in a third language (such as Spanish), as illustrated in Figure 10. This can be particularly effective if the two languages are very similar (as in Spanish-French) or if their available Bible versions are a close translation of a common source (e.g. the Latin Vulgate Bible). As shown in Table 5.19, using the previously analyzed French Bible as a bridge for Spanish achieves performance (97.4% precision) comparable to the use of 3 parallel English Bible versions.

Induced Morphological Analyses for CZECH

		Analysis		
Inflection	Root Out	Point of Suffixation Change	Suffix	TopBridge
bral	brát	át→a	+l	marry
brala	brát	át→a	+la	accept
brali	brát	át→a	+li	marry
byl	být	ýt→y	+l	be
byli	být	ýt→y	+li	be
bylo	být	ýt→y	+lo	be
byly	být	ýt→y	+ly	be
chovala	chovat	t→ ϵ	+la	behave
chová	chovat	at→ ϵ	+á	behave
chováme	chovat	at→ ϵ	+áme	behave
chovají	chovat	t→j	+í	behave
chodila	chodit	t→ ϵ	+la	walk
chodí	chodit	it→ ϵ	+í	walk
choďte	chodit	dit→ďt	+e	swim
chránila	chránit	t→ ϵ	+la	protect
chráněn	chránit	it→ ěn	+ϵ	protect
chrání	chránit	it→ ϵ	+í	protect
couvají	couvat	t→j	+í	back
couval	couvat	t→ ϵ	+l	back
chce	chtít	tít→c	+e	want
chceme	chtít	tít→c	+eme	want
chcete	chtít	tít→c	+ete	want
chceš	chtít	tít→c	+eš	want
chci	chtít	tít→c	+i	want
chtıjř	chtít	ít→tıjř	+ϵ	want
chtìl	chtít	ít→tı	+l	want
chtìli	chtít	ít→tı	+li	want
chtějí	chtít	ít→ěj	+í	want
chtěl	chtít	ít→ě	+l	want
chtěli	chtít	ít→ě	+li	want
chtělo	chtít	ít→ě	+lo	want

Table 4.30: Sample of induced morphological analyses in Czech

Induced Morphological Analyses for SPANISH

		Analysis		
Inflection	Root Out	Point of Suffixation Change	Suffix	TopBridge
aborrecen	aborrecer	er→ ϵ	+en	hate
aborreció	aborrecer	er→ ϵ	+ió	hate
aborrecí	aborrecer	er→ ϵ	+í	hate
aborrecía	aborrecer	er→ ϵ	+ía	hate
aborrezco	aborrecer	cer→zc	+o	hate
abrace	abrazar	zar→c	+e	embrace
abrazado	abrazar	ar→ ϵ	+ado	embrace
adquiere	adquirir	rir→er	+e	get
adquirido	adquirir	ir→ ϵ	+ido	get
andamos	andar	ar→ ϵ	+amos	walk
andando	andar	ar→ ϵ	+ando	walk
andarán	andar	ar→ ϵ	+arán	wander
andarás	andar	ar→ ϵ	+arás	wander
andaré	andar	ar→ ϵ	+aré	walk
andemos	andar	ar→ ϵ	+emos	walk
anden	andar	ar→ ϵ	+en	walk
andes	andar	ar→ ϵ	+es	walk
anduvo	andar	ar→uv	+o	walk
bendecid	bendecir	ir→ ϵ	+id	bless
bendecido	bendecir	ir→ ϵ	+ido	bless
bendecimos	bendecir	ir→ ϵ	+imos	bless
bendice	bendecir	ecir→ ϵ	+ice	bless
bendiciendo	bendecir	ecir→iciend	+o	bless
bendición	bendecir	ecir→ición	+ϵ	bless
bendiga	bendecir	ecir→ig	+a	bless
bendigan	bendecir	ecir→ig	+an	bless
bendijeren	bendecir	ecir→ijer	+en	bless
bendijo	bendecir	ecir→ij	+o	bless
buscáis	buscar	ar→ ϵ	+áis	seek
buscó	buscar	ar→ ϵ	+ó	seek
busque	buscar	car→qu	+e	seek
busqué	buscar	car→qu	+é	seek

Table 4.31: Sample of induced morphological analyses in Spanish

Morphology Induction Performance

Some additional general observations are in order regarding the morphology performance shown in Tables 5.19, 5.17 and 5.18:

- Performance induction using the French Bible as the bridge source is evaluated using the full test verb set from the French Hansards. The strong performance illustrates that even a small single text in a very different genre can provide effective transfer to modern (conversational) French. While the observed genre and topic-sensitive vocabulary differs substantially between the Bible and Hansards, the observed inventories of stem changes and suffixation actually have large overlap, as do the set of observed high-frequency irregular verbs. Thus the inventory of morphological phenomena seem to translate better across genre than do lexical choice and collocation models.

- Over 60% of errors are due to gaps in the candidate rootlists. Currently the candidate rootlists are derived automatically by applying the projected POS models and selecting any word with uninflected verb probability greater than a generous threshold and ending in a canonical verb suffix. False positives are easily tolerated (less than 5% of errors are due to spurious non-root competitors), but with missing roots the algorithms are forced either to propose previously unseen roots or align to the closest previously observed root candidate. Thus while *no* non-English dictionary was used in the computation of these results, it would substantially improve performance to have a dictionary-based inventory of potential roots to map into.

- Performance in all languages has been significantly hindered by low-accuracy parallel-

corpus word-alignments using the original Model 3 GIZA system. Use of Franz Joseph Och's recently released GIZA++ [Al-Onaizan et al., 1999] word-alignment models should improve performance for all of the applications studied in this paper, as would iterative re-alignments using richer alignment features (including lemma and part-of-speech) derived from this research.

- The current somewhat lower performance on Czech is due to several factors. They include (a) very low accuracy initial word-alignments due to often non-parallel translations of the Reader's Digest sample and the failure of the initial word-alignment models to handle the highly inflected Czech morphology. (b) the small size of the Czech parallel corpus (less than twice the length of the Bible). (c) the common occurrence in Czech of two very similar perfective and non-perfective root variants (e.g. *odolávat* and *odolat*, both of which mean *to resist*). A simple monolingual dictionary-derived list of canonical roots would resolve ambiguity regarding which is the appropriate target.

- Many of the errors are due to all (or most) inflections of a single verb mapping to the same (wrong) root. But for many applications where the function of lemmatization is to cluster equivalent words (e.g. stemming for information retrieval), the choice of label for the lemma is less important than correctly linking the members of the lemma.

- The learning curves in Figure 11 shows the strong correlation between performance and size of the aligned corpus. Given that large quantities of parallel text currently exist in translation bureau archives and OCR-able books, not to mention the increas-

ing online availability of bitext on the web, the natural growth of available bitext quantities should continue to support performance improvement.

- The system analysis examples shown in Tables 4.30 and 4.31 are representative of system performance and selected to illustrate the range of encountered phenomena. All system evaluation is based on the task of selecting the correct root for a given inflection (which has a long lexicography-based consensus regarding the "truth"). In contrast, the descriptive analysis of any such pairing is very theory dependent without standard consensus. Thus the given decomposition into stem-change and affix(es) is somewhat arbitrary and is provided to show insight into system performance. The "TopBridge" column shows the strongest English bridge lemma utilized in mapping (typically one of many). When no entry is given, no above-threshold bridge was detected, and the root was selected using the subsequently derived MTrie analysis.

These results are quite impressive in that they are based on essentially no language-specific knowledge of French, Spanish or Czech. The multilingual bridge algorithm is surface-form independent, and can just as readily handle obscure infixational or reduplicative morphological processes.

The Translingual Bridge similarity measure is able to align complex inflected word forms with their root forms, even when their surface similarity is quite different or highly irregular. Previous major investments in English annotated corpora and tool development can be maximally leveraged across languages, achieving accurate stand-alone tool development in other languages without comparable human annotation efforts.

Chapter 5

Model Combination

5.1 Overview

As shown empirically in Chapter 4, none of the similarity models is sufficiently effective on its own. However, traditional classifier combination techniques can be applied to merge these models. To improve performance further, the output of the merged system can be used to iteratively retrain the parameters of each of its four component models. The eventual output can be used as noisy training data for the supervised algorithms presented in Chapter 3.

Table 5.1 illustrates an overview of the combined measures in action, showing the relative rankings of candidate roots for the English inflection *gave*. The initial Context model did not rank *give* at or near the top of the highest ranked candidate roots, and the combination of the Context and Levenshtein models only ranked *give* as the second most likely root candidate. However, the top ranked candidates from the combination of Context and Levenshtein were used to train a combination of supervised models which misanalyzed

Candidate Roots for the English inflection **GAVE**:

Iteration 0(b)												Iteration 5(c)		
Context Similarity			Levenshtein Similarity			CS + LS Similarity			Supervised Models			Supervised Models		
1	while	0.636194	1	**give**	0.2	1	have	4	1	love	0.046380	1	**give**	0.318705
2	work	0.588139	2	have	1	2	**give**	8.4	2	wave	0.007600	2	gave	0.000581
3	run	0.581005	3	wave	1	3	move	17.6	3	**give**	0.004680	3	gavee	0.000006
4	make	0.578019	4	glove	1.2	4	live	21.4	4	move	0.000397	4	gav	0.000001
5	position	0.575599	5	live	1.2	5	while	226.2	5	live	0.000374			
6	play	0.575243	6	love	1.2	6	work	226.4						
	...			...			...							
38	**give**	0.512497												

Table 5.1: Unsupervised bootstrapping with iterative retraining used in the experiments presented in this chapter.

gave as an inflection of *love*. However, after 5 iterations involving both supervised and unsupervised retraining, the best analysis given by a combinination of supervised models correctly (and confidently) idenitifies the root as *give*, with a score of 0.319 which is much greater than the 2nd place candidate, *gave* with a score of 0.00058.

Figure 5.1 shows the overall architecture of the iterative retraining process. First, the initial model parameters for the similarity measures are used to create initial analyses in each of the models. The analyses from each of the similarity measures are combined and used as training data for the supervised models. These models can then be used to estimate better initial model parameters for the similarity measures. The final analysis is the result of combining the iteratively retrained similarity measures with the output of the supervised models.

Table 5.2 shows the actual iterations used for the experiments presented in the following sections.

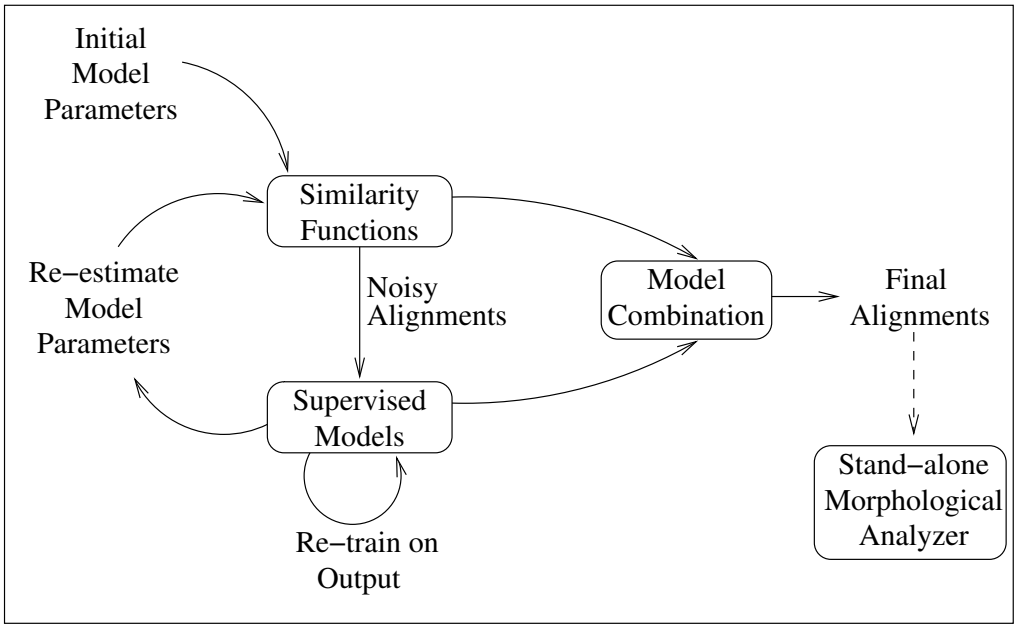


Figure 5.1: The iterative re-training pipeline.

5.2 Model Combination and Selection

The major problem with combining the unsupervised similarity models is that the dynamic range of the various measures is quite different and not amenable to direct combination (whether it be multiplicative or additive). For example, the Levenshtein distance is a value greater than or equal to zero such that the most similar string is the one with the smallest distance. On the other hand, the cosine similarity measure used by the context model returns scores between 0 and 1 where the most similar vector has the highest score.

To address the problem, the raw scores returned by the models are not used in combination; rather, combination of models is done by rank. The final analysis score for an inflection and a proposed root is determined by a weighted average of the rank of that root in each model. Some models do not provide a ranking for every inflection-root analysis. For

Iteration 0	(a) Initialize the parameters of the Levenshtein and Context similarity models to default values (as presented in Sections 4.4 and 4.5) and get the 1-best consensus output for each inflection. (b) Use the output of 0(a) to train the 4 supervised models and get the 1-best consensus output for each inflection.
Iteration 1	(a) Use the output of 0(b) to train the 2 unsupervised models and get the 1-best consensus output for each inflection. (b) Use the output of 1(a) to train the 4 supervised models and get the 1-best consensus output for each inflection.
Iteration 2	(a) [nothing] (b) Use the output of 1(b) to train the 4 supervised models and get the 1-best consensus output for each inflection.
Iteration 3	(a) Use the output of 2(b) to train the 2 unsupervised models and get the 2-best consensus output for each inflection. (b) Use the output of 3(a) to train the 4 supervised models and get the 1-best consensus output for each inflection.
Iteration 4	(a) [nothing] (b) Use the output of 3(b) to train the 4 supervised models and get the 1-best consensus output for each inflection.
Iteration 5	(a) Use the output of 4(b) to train the 2 unsupervised models and keep *all* consensus outputs for each inflection. (b) [nothing] (c) Build a combined model from 4(b) and 5(a) using the decision tree in Figure 5.2 (on page 186).

Table 5.2: The iterative retraining pipeline used in the presented experiments. The unsupervised models are combined by a weighted average of rank, and the supervised models by equal average of probabilities (Section 3.6.2).

example, the context similarity function does not return a ranking for inflections that were not present in the corpus. If a model ranks only k possible analyses of a given inflection, all other roots are considered to have rank $k + 1$.

Sections 5.4.1 and 5.4.2 evaluate some of the variations on the procedure in Table 5.2: using 1-best consensus throughout, and using an unequal average of supervised models.

5.3 Iterating with unsupervised models

5.3.1 Re-estimating parameters

Since no supervised training pairs are available to train the supervised model of Chapter 3, the unsupervised similarity methods are used to provide a noisy list of inflection-root training pairs. However, as was shown extensively throughout Chapter 4, the performance of the Context and Levenshtein similarity measures is determined in large part by the initial parameters used in each model. Although these parameters are not known ahead of time, and no training data is available to determine them, they can be effectively re-estimated during the iteration of Table 5.2.

The process of re-estimating parameters is identical for each of the similarity models. First, an initial analysis provided by the similarity models is used as training data for the supervised morphological transformation models of Chapter 3. The output of the supervised models is used as an approximation of the true analysis to evaluate each of the parameters to the similarity models. The similarity models are then re-run using these new parameters and this process iterates until iteration 5.[1]

For example, the re-estimation of the window size works as follows. Recall that Table 4.11 presented the context similarity model's sensitivity to initial window size for each language. Using the iterative parameter re-estimation method, a good window size can be learned, as shown in Table 5.3. To retain the Context model's window size at each iteration, the window size is chosen for which the Context model's output best matches the output of the supervised model from the previous (b) iteration. Table 5.3 shows the accuracy of each

[1]Though not done here, this could be run until the parameters converge.

UNWEIGHTED WINDOW SIZE PARAMETERS FOR CONTEXT SIMILARITY

Language	System	1x1	2x2	3x3	4x4	5x5	10x10	Diff.
Portuguese	Iter. 1(a)	13.2%	22.1%	24.3%	**24.6%**	23.4%	19.7%	-0.7%
	Iter. 3(a)	15.9%	25.2%	**26.8%**	26.2%	25.2%	20.3%	0.0%
	Iter. 5(a)	16.1%	25.6%	**27.3%**	26.8%	25.7%	20.9%	**0.0%**
	Truth	17.3%	27.1%	**28.6%**	27.9%	26.6%	21.4%	-
Estonian	Iter. 1(a)	28.2%	33.4%	35.3%	35.1%	**35.6%**	34.2%	-1.5%
	Iter. 3(a)	33.4%	37.1%	**39.3%**	38.5%	38.4%	36.6%	0.0%
	Iter. 5(a)	33.0%	36.7%	**38.8%**	37.6%	37.4%	35.7%	**0.0%**
	Truth	39.9%	42.7%	**44.4%**	42.9%	42.2%	39.6%	-
Russian	Iter. 1(a)	33.7%	40.0%	**41.9%**	39.8%	37.9%	30.4%	0.0%
	Iter. 3(a)	33.1%	37.6%	**38.6%**	37.0%	34.2%	26.9%	0.0%
	Iter. 5(a)	35.1%	39.4%	**40.7%**	39.3%	36.3%	28.4%	**0.0%**
	Truth	40.3%	46.2%	**46.4%**	44.7%	40.9%	32.2%	-
German	Iter. 1(a)	7.4%	10.6%	11.7%	12.4%	**12.5%**	11.9%	0.0%
	Iter. 3(a)	9.0%	11.5%	12.8%	**13.7%**	13.6%	12.7%	-0.1%
	Iter. 5(a)	9.2%	11.6%	12.8%	**13.6%**	**13.6%**	12.7%	**-0.1%**
	Truth	9.6%	12.4%	13.9%	14.7%	**14.8%**	13.9%	-
Turkish	Iter. 1(a)	34.4%	38.2%	41.2%	**43.1%**	42.9%	36.2%	-1.4%
	Iter. 3(a)	43.9%	44.9%	**45.4%**	43.9%	43.8%	39.8%	0.0%
	Iter. 5(a)	44.4%	45.0%	**45.5%**	43.8%	43.4%	39.8%	**0.0%**
	Truth	45.8%	46.3%	**46.5%**	45.1%	44.4%	40.8%	-
Basque	Iter. 1(a)	9.3%	16.2%	19.1%	21.6%	**23.6%**	20.0%	-0.9%
	Iter. 3(a)	13.6%	**17.8%**	17.4%	16.7%	16.6%	14.9%	0.0%
	Iter. 5(a)	14.2%	18.6%	**18.7%**	17.8%	17.9%	16.2%	**-0.3%**
	Truth	16.0%	**20.7%**	*20.4%*	19.7%	19.8%	18.0%	-
English	Iter. 1(a)	11.6%	16.5%	**17.8%**	17.6%	16.3%	13.5%	0.0%
	Iter. 3(a)	16.6%	22.6%	**23.5%**	22.6%	21.2%	17.5%	0.0%
	Iter. 5(a)	16.6%	22.6%	**23.2%**	22.3%	20.7%	17.1%	**0.0%**
	Truth	17.2%	23.5%	**24.2%**	23.3%	22.0%	18.2%	-

Table 5.3: Re-estimation of the initial window size in Context Similarity. For each language, the first row (labeled as Truth) is the true accuracy of the alignment provided under each of the window size configurations. At each of the iterations where the window size is re-estimated, the configurations are graded against the output of the supervised morphological similarity models; hence, the performance listed at each iteration is the agreement rate between the supervised models and the particular parameterization of the Context model. The column labeled *Diff.* shows the difference in performance of the selected parameter from the most effective parameter, as graded by the truth. This tends to improve slightly over time, at least from Iteration 1(a) to Iteration 3(a).

window size when scored in this way against the previous iteration's output; the chosen top-scoring window size is shown in boldface.

Ideally, the window size that scored best against the truth should be chosen, as show in the last row for each language, but this unsupervised approximation scores almost as well against the truth as shown by the "Diff." column.

In three of the six cases where the most effective window size is selected in the final iteration (Portuguese, Estonian, and Turkish), the parameter selected at Iteration 1(a) is suboptimal, indicating that the iteration and re-estimation process successfully recovers the most effective configuration.

It is possible for the estimated accuracy of the context models to outperform their true accuracy. For example, in the 5x5 window size configuration for Basque at Iteration 1(a), the performance of the context model is listed as 23.6%, while the performance on the truth in only 19.8%. This disparity arises because the output of the Context model can better reproduce the output of the combined supervised models than it can reproduce the truth. That should not be surprising since the combined supervised models were themselves partially trained on the output of the Context model.

Of the remaining context model parameters – the window size when using weighted positioning (Table 5.4), whether or not to use tf-idf (Table 5.5), and deciding which stop-words should be removed from the model (Table 5.6) (originally presented in Tables 4.12, 4.13, and 4.14) – are all chosen optimally after just 2 iterations. The grammar-sensitive window position (originally presented in Tables 4.15, and 4.16) converges to the optimal parameter for only 4 of the 7 presented languages. The difference between performance of

WEIGHTED WINDOW SIZE PARAMETERS FOR CONTEXT SIMILARITY

Language	System	Weighted 2x2	Weighted 3x3	Weighted 4x4	Weighted 5x5	Weighted 6x6	Weighted 7x7
Portuguese	Iter. 1(a)	19.4%	22.6%	24.6%	25.5%	**26.0%**	25.8%
	Iter. 3(a)	22.7%	25.9%	27.6%	**28.3%**	28.3%	28.1%
	Iter. 5(a)	23.1%	26.3%	28.1%	**28.9%**	28.8%	28.7%
	Truth	24.6%	28.0%	29.7%	**30.3%**	30.2%	29.9%
Estonian	Iter. 1(a)	32.5%	35.3%	35.4%	**35.8%**	35.5%	35.7%
	Iter. 3(a)	37.5%	39.5%	40.2%	**40.7%**	40.0%	40.0%
	Iter. 5(a)	37.2%	39.1%	39.6%	**40.1%**	39.5%	39.4%
	Truth	43.7%	45.1%	45.3%	**45.7%**	45.1%	44.6%
Russian	Iter. 1(a)	38.9%	42.0%	**43.4%**	42.8%	42.1%	40.3%
	Iter. 3(a)	37.3%	38.6%	**40.1%**	39.4%	38.6%	37.4%
	Iter. 5(a)	39.7%	40.9%	**42.0%**	41.3%	40.6%	39.4%
	Truth	46.2%	47.4%	**48.1%**	47.4%	46.5%	45.3%
German	Iter. 1(a)	-	11.1%	-	**12.4%**	-	-
	Iter. 3(a)	-	12.4%	-	**13.4%**	-	-
	Iter. 5(a)	-	12.5%	-	**13.6%**	-	-
	Truth	-	13.3%	-	**14.6%**	-	-
Turkish	Iter. 1(a)	38.7%	41.5%	43.4%	45.0%	**45.4%**	45.1%
	Iter. 3(a)	46.4%	**46.9%**	46.5%	46.8%	46.6%	46.1%
	Iter. 5(a)	46.8%	**47.2%**	46.7%	46.8%	46.5%	45.8%
	Truth	48.0%	**48.4%**	47.8%	47.9%	47.7%	47.2%
Basque	Iter. 1(a)	-	18.2%	-	**22.9%**	-	-
	Iter. 3(a)	-	19.0%	-	**19.8%**	-	-
	Iter. 5(a)	-	20.1%	-	**21.0%**	-	-
	Truth	-	22.2%	-	**23.1%**	-	-
English	Iter. 1(a)	-	18.0%	-	**19.5%**	-	-
	Iter. 3(a)	-	24.3%	-	**26.1%**	-	-
	Iter. 5(a)	-	24.2%	-	**25.8%**	-	-
	Truth	-	25.2%	-	**26.9%**	-	-

Table 5.4: Re-estimation of optimal weighted window size in Context Similarity. Gaps in the table indicate that accuracy was not evaluated at these positions.

TF-IDF PARAMETERS FOR CONTEXT SIMILARITY

Language	System	with TF-IDF	without TF-IDF
Portuguese	Iter. 1(a)	**23.4%**	13.7%
	Iter. 3(a)	**25.2%**	15.2%
	Iter. 5(a)	**25.7%**	15.5%
	Truth	**26.6%**	16.4%
Estonian	Iter. 1(a)	**35.6%**	27.5%
	Iter. 3(a)	**38.4%**	26.4%
	Iter. 5(a)	**37.4%**	24.9%
	Truth	**42.2%**	28.2%
Russian	Iter. 1(a)	**37.9%**	28.7%
	Iter. 3(a)	**34.2%**	26.0%
	Iter. 5(a)	**36.3%**	27.7%
	Truth	**40.9%**	31.9%
German	Iter. 1(a)	**12.5%**	9.4%
	Iter. 3(a)	**13.6%**	10.4%
	Iter. 5(a)	**13.6%**	10.4%
	Truth	**14.8%**	11.1%
Turkish	Iter. 1(a)	**42.9%**	27.7%
	Iter. 3(a)	**43.8%**	29.8%
	Iter. 5(a)	**43.4%**	29.5%
	Truth	**44.4%**	30.3%
Basque	Iter. 1(a)	**23.6%**	10.7%
	Iter. 3(a)	**16.6%**	7.9%
	Iter. 5(a)	**17.9%**	8.3%
	Truth	**19.8%**	9.1%
English	Iter. 1(a)	**16.3%**	7.0%
	Iter. 3(a)	**21.2%**	9.1%
	Iter. 5(a)	**20.7%**	8.8%
	Truth	**22.0%**	9.3%

Table 5.5: Re-estimation of the decision to use tf-idf for the context similarity model

STOPWORD LIST PARAMETERS FOR CONTEXT SIMILARITY

Language	System	Remove Nothing	Remove Punctuation	Remove 100 Most Freq	Remove Function Words
Portuguese	Iter. 1(a)	23.4%	**23.7%**	23.6%	22.6%
	Iter. 3(a)	25.2%	25.4%	**26.7%**	25.9%
	Iter. 5(a)	25.7%	26.0%	**27.1%**	26.3%
	Truth	26.6%	26.9%	**28.3%**	27.5%
Estonian	Iter. 1(a)	**35.6%**	34.3%	33.5%	-
	Iter. 3(a)	38.4%	39.6%	**40.5%**	-
	Iter. 5(a)	37.4%	39.0%	**39.8%**	-
	Truth	42.2%	44.2%	**45.5%**	-
Russian	Iter. 1(a)	37.9%	**40.8%**	38.1%	30.7%
	Iter. 3(a)	34.2%	**36.7%**	35.3%	27.7%
	Iter. 5(a)	36.3%	**39.0%**	37.2%	28.9%
	Truth	40.9%	**44.3%**	43.2%	33.9%
German	Iter. 1(a)	**12.5%**	**12.5%**	12.2%	11.8%
	Iter. 3(a)	13.6%	13.6%	**14.2%**	13.2%
	Iter. 5(a)	13.6%	13.6%	**14.1%**	13.1%
	Truth	14.8%	14.8%	**15.3%**	14.1%
Turkish	Iter. 1(a)	42.9%	**44.6%**	42.2%	38.0%
	Iter. 3(a)	43.8%	**43.9%**	41.8%	39.9%
	Iter. 5(a)	43.4%	**43.6%**	41.9%	40.5%
	Truth	44.4%	**44.9%**	43.6%	41.9%
Basque	Iter. 1(a)	**23.6%**	23.4%	22.8%	23.4%
	Iter. 3(a)	16.6%	17.4%	**19.2%**	17.4%
	Iter. 5(a)	17.9%	18.5%	**20.3%**	18.5%
	Truth	19.8%	20.5%	**22.6%**	20.5%
English	Iter. 1(a)	16.3%	17.5%	**20.1%**	19.0%
	Iter. 3(a)	21.2%	22.8%	**26.9%**	26.0%
	Iter. 5(a)	20.7%	22.2%	**26.5%**	25.8%
	Truth	22.0%	23.5%	**27.9%**	27.1%

Table 5.6: Choosing stop words in the Context Similarity model

WINDOW POSITION FOR CONTEXT SIMILARITY

Language	System	6 x 0	5 x 1	4 x 2	3 x 3	2 x 4	1 x 5	0 x 6	Diff.
Portuguese	Iter. 1(a)	11.1%	17.5%	22.6%	24.3%	23.7%	23.4%	**26.6%**	0.0%
	Iter. 3(a)	11.2%	19.7%	25.5%	26.8%	26.9%	26.9%	**30.7%**	0.0%
	Iter. 5(a)	11.6%	20.0%	26.0%	27.3%	27.4%	27.3%	**30.9%**	**0.0%**
	Truth	12.0%	21.4%	27.5%	28.6%	29.0%	29.0%	**32.9%**	-
Estonian	Iter. 1(a)	26.7%	31.6%	34.2%	**35.3%**	34.6%	34.9%	27.1%	-0.3%
	Iter. 3(a)	29.2%	35.9%	38.5%	**39.3%**	38.8%	38.8%	29.8%	-0.3%
	Iter. 5(a)	28.9%	35.5%	38.2%	**38.8%**	38.0%	38.3%	29.2%	**-0.3%**
	Truth	31.9%	41.4%	44.0%	*44.4%*	44.3%	**44.7%**	32.2%	-
Russian	Iter. 1(a)	17.1%	34.8%	41.2%	41.9%	**42.9%**	42.6%	41.8%	0.0%
	Iter. 3(a)	15.4%	33.7%	38.6%	38.6%	39.6%	39.4%	**39.7%**	-0.6%
	Iter. 5(a)	16.5%	35.7%	40.7%	40.7%	41.4%	41.2%	**41.6%**	**-0.6%**
	Truth	19.9%	41.1%	47.3%	46.4%	**47.9%**	47.4%	*47.3%*	-
German	Iter. 1(a)	8.1%	9.3%	11.0%	**11.7%**	11.6%	10.8%	8.5%	0.0%
	Iter. 3(a)	9.3%	10.4%	12.2%	**12.8%**	12.7%	12.4%	9.2%	0.0%
	Iter. 5(a)	9.4%	10.7%	12.3%	12.8%	**12.9%**	12.6%	9.4%	**-0.1%**
	Truth	10.0%	11.3%	13.1%	**13.9%**	*13.8%*	13.3%	10.0%	-
Turkish	Iter. 1(a)	**42.3%**	42.2%	41.6%	41.2%	40.3%	38.5%	25.8%	0.0%
	Iter. 3(a)	**50.5%**	47.7%	46.9%	45.4%	45.4%	44.2%	25.3%	0.0%
	Iter. 5(a)	**51.1%**	47.8%	46.9%	45.5%	45.7%	44.3%	25.0%	**0.0%**
	Truth	**52.7%**	49.1%	48.1%	46.5%	47.0%	45.4%	25.3%	-
Basque	Iter. 1(a)	18.2%	17.2%	18.7%	**19.1%**	18.2%	16.1%	8.5%	-5.5%
	Iter. 3(a)	**22.2%**	18.3%	18.8%	17.4%	18.6%	16.9%	5.4%	0.0%
	Iter. 5(a)	**23.4%**	19.3%	19.8%	18.7%	19.6%	17.8%	6.0%	**0.0%**
	Truth	**25.9%**	21.4%	21.9%	20.4%	21.7%	19.8%	6.4%	-
English	Iter. 1(a)	9.4%	-	-	17.8%	-	-	**18.3%**	0.0%
	Iter. 3(a)	12.9%	-	-	23.5%	-	-	**24.8%**	0.0%
	Iter. 5(a)	12.9%	-	-	23.2%	-	-	**24.3%**	**0.0%**
	Truth	13.3%	-	-	24.2%	-	-	**25.7%**	-

Table 5.7: Re-estimation of optimal window position for the Context Similarity model. The column labeled "Diff." refers to the difference in performance between the parameter selected and the most effective parameter as selected by the truth. A score of 0.0% in this column indicates the most effective parameters were selected.

the selected window position and the most effective window position is, on average, less than 0.15%; however, choosing randomly among the window sizes yields an average performance difference of 4.77%.

The re-estimation of parameters for the Levenshtein model[2] is performed similarly but less successfully. The ability to choose an effective transition cost matrix (Table 5.8) from a set of matrices was successful in only three of the seven languages. In the remaining four, three chose the second-best performing matrix (as graded against the truth). In the remaining language, Russian, parameter re-estimation chose the fourth-best performing matrix.

When choosing an effective split configuration for Levenshtein similarity (Table 5.9), the re-estimation lead to the optimal split for only two languages. In Russian, three models were tied as the best performing, but choosing between them would have to be done at random. In English, the choice of the suboptimal split method 2 results in nearly a 75% increase in error rate relative to the optimal split method 3.

For the prefix penalty parameter (Table 5.10), only Basque had its optimal penalty chosen. However, the difference in performance between the optimal parameter and the selected parameter was quite low for many of the languages. Only Russian showed a decrease of more than 1.5% from the optimal parameter, and all languages selected a parameter which performed at or above the initial baseline parameterization.

[2]Originally presented in Tables 4.21, 4.26, 4.25, 4.28.

TRANSITION COST MATRIX PARAMETERS FOR LEVENSHTEIN SIMILARITY

Language	System	baseline	acl-2000	2Δ	equal	$\frac{1}{2}\Delta$	swap	Diff.
Portuguese	Iter. 1(a)	91.9%	91.2%	91.7%	**93.4%**	85.3%	79.2%	0.0%
	Iter. 3(a)	98.2%	97.7%	98.2%	**98.4%**	94.1%	94.6%	0.0%
	Iter. 5(a)	99.1%	98.6%	99.2%	**99.6%**	94.7%	95.1%	**0.0%**
	Truth	94.0%	92.7%	97.2%	**98.1%**	83.8%	78.7%	-
Estonian	Iter. 1(a)	88.6%	86.2%	89.5%	**90.0%**	80.1%	74.5%	0.0%
	Iter. 3(a)	96.0%	95.0%	95.3%	**96.3%**	92.3%	91.2%	0.0%
	Iter. 5(a)	96.4%	95.5%	96.1%	**97.0%**	92.6%	91.4%	**0.0%**
	Truth	78.5%	81.0%	90.3%	**92.1%**	78.2%	74.9%	-
Russian	Iter. 1(a)	70.1%	69.6%	71.2%	**73.1%**	65.0%	56.2%	0.0%
	Iter. 3(a)	**90.3%**	90.1%	90.2%	90.2%	87.5%	85.8%	-8.8%
	Iter. 5(a)	**93.8%**	93.5%	93.2%	93.6%	90.8%	87.0%	**-8.8%**
	Truth	*81.0%*	83.6%	86.0%	**89.8%**	78.2%	68.1%	-
German	Iter. 1(a)	**97.3%**	96.3%	96.4%	97.2%	88.3%	72.7%	-2.9%
	Iter. 3(a)	**97.4%**	97.2%	97.3%	97.3%	93.6%	84.2%	-2.9%
	Iter. 5(a)	98.5%	98.3%	98.1%	**98.5%**	94.5%	84.9%	**0.0%**
	Truth	91.9%	91.8%	92.5%	**94.8%**	85.7%	73.7%	-
Turkish	Iter. 1(a)	84.2%	84.4%	80.3%	**87.1%**	83.6%	75.0%	-1.9%
	Iter. 3(a)	99.2%	98.9%	99.3%	**99.4%**	90.5%	93.9%	-1.9%
	Iter. 5(a)	99.5%	99.0%	99.7%	**99.7%**	90.2%	94.2%	**-1.9%**
	Truth	89.6%	85.5%	**96.8%**	*94.9%*	58.2%	58.3%	-
Basque	Iter. 1(a)	62.2%	56.6%	56.4%	**66.5%**	38.7%	32.8%	-0.2%
	Iter. 3(a)	89.3%	87.0%	92.8%	**93.1%**	78.5%	88.5%	-0.2%
	Iter. 5(a)	91.2%	88.9%	96.5%	**97.8%**	80.2%	90.9%	**-0.2%**
	Truth	73.9%	71.4%	**92.5%**	*92.3%*	62.8%	73.9%	-
English	Iter. 1(a)	83.4%	83.1%	84.1%	**87.9%**	68.5%	61.2%	-0.1%
	Iter. 3(a)	98.0%	97.8%	98.3%	**98.9%**	85.2%	82.1%	-0.1%
	Iter. 5(a)	98.8%	98.5%	99.5%	**99.5%**	86.4%	82.8%	**-0.1%**
	Truth	84.9%	85.3%	**96.0%**	*95.9%*	70.1%	65.2%	-

Table 5.8: Estimating the most effective transition cost matrix for the Levenshtein similarity models. For six of the seven languages, the first or second most effective matrix was chosen. The *Diff.* column shows the performance decrease in the model using the highest ranked parameterization, as graded against the true alignment.

SPLIT METHOD PARAMETERS FOR LEVENSHTEIN SIMILARITY

Language	System	Split 0	Split 1	Split 2	Split 3	Split 4	Split 5	Diff
Portuguese	Iter. 1(a)	91.6%	91.7%	88.0%	91.9%	**91.9%**	88.5%	-1.4%
	Iter. 3(a)	98.0%	98.2%	97.9%	98.2%	**98.2%**	98.1%	-1.4%
	Iter. 5(a)	99.0%	99.1%	98.8%	99.1%	**99.2%**	99.0%	**-1.4%**
	Truth	93.6%	94.2%	95.4%	94.0%	*94.5%*	**95.9%**	-
Estonian	Iter. 1(a)	88.5%	**88.8%**	80.1%	88.6%	88.6%	80.7%	-0.5%
	Iter. 3(a)	95.4%	**96.2%**	88.8%	96.0%	96.1%	90.0%	-0.5%
	Iter. 5(a)	95.7%	96.9%	89.1%	96.4%	**97.0%**	90.4%	**0.0%**
	Truth	76.9%	79.6%	74.8%	78.5%	**80.1%**	75.4%	-
Russian	Iter. 1(a)	69.7%	69.2%	64.4%	**70.1%**	69.5%	64.8%	0.0%
	Iter. 3(a)	**90.3%**	**90.3%**	85.2%	**90.3%**	**90.3%**	85.6%	-0.4%
	Iter. 5(a)	**93.8%**	93.7%	88.3%	**93.8%**	**93.8%**	88.4%	**-0.3%**
	Truth	80.6%	80.3%	78.2%	**81.0%**	80.5%	78.3%	-
German	Iter. 1(a)	97.3%	97.6%	82.1%	97.3%	**97.6%**	82.4%	-0.2%
	Iter. 3(a)	97.3%	97.4%	86.4%	97.4%	**97.4%**	86.7%	-0.2%
	Iter. 5(a)	98.5%	98.6%	87.3%	98.5%	**98.6%**	87.7%	**-0.2%**
	Truth	92.1%	**92.9%**	75.6%	91.9%	92.7%	75.3%	-
Turkish	Iter. 1(a)	81.7%	81.7%	77.2%	84.2%	**84.2%**	82.0%	-0.4%
	Iter. 3(a)	99.1%	99.1%	97.0%	**99.2%**	**99.2%**	97.7%	-0.5%
	Iter. 5(a)	99.4%	99.4%	97.2%	**99.5%**	**99.5%**	97.8%	**-0.5%**
	Truth	90.0%	**90.1%**	82.1%	89.6%	89.7%	82.2%	-
Basque	Iter. 1(a)	**62.2%**	59.5%	50.3%	**62.2%**	59.5%	50.3%	-2.7%
	Iter. 3(a)	89.3%	89.4%	**89.6%**	89.3%	89.4%	**89.6%**	0.0%
	Iter. 5(a)	91.2%	91.1%	**92.6%**	91.2%	91.1%	**92.6%**	**0.0%**
	Truth	73.9%	74.3%	**76.6%**	73.9%	74.3%	**76.6%**	-
English	Iter. 1(a)	**83.4%**	83.0%	80.5%	**83.4%**	83.0%	80.5%	-7.0%
	Iter. 3(a)	**98.0%**	**98.0%**	96.5%	**98.0%**	**98.0%**	96.5%	-6.5%
	Iter. 5(a)	98.8%	**98.8%**	97.7%	98.8%	**98.8%**	97.7%	**-5.6%**
	Truth	84.9%	86.0%	**91.9%**	84.9%	86.0%	**91.9%**	-

Table 5.9: Re-estimating the best performing split method in Levenshtein similarity. When multiple parameterizations are ranked equally, the performance difference listed is the average decrease in the performance across the equally ranked parameterizations.

PREFIX PENALTY PARAMETERS FOR LEVENSHTEIN SIMILARITY

Language	System	Prefix Penalty 20	10	5	3	1	0.25	0	Diff.
Portuguese	Iter. 1(a)	89.3%	89.4%	89.6%	89.8%	90.3%	91.0%	**91.9%**	-2.7%
	Iter. 3(a)	98.0%	98.0%	98.0%	98.0%	98.1%	**98.2%**	98.2%	-1.0%
	Iter. 5(a)	99.0%	99.0%	99.0%	99.0%	99.1%	**99.2%**	99.1%	**-1.0%**
	Truth	96.6%	96.6%	96.6%	**96.7%**	96.6%	*95.7%*	94.0%	-
Estonian	Iter. 1(a)	86.5%	86.5%	86.5%	86.8%	87.0%	87.3%	**88.6%**	-3.4%
	Iter. 3(a)	95.1%	95.1%	95.3%	95.4%	95.7%	**96.1%**	96.0%	-1.2%
	Iter. 5(a)	96.2%	96.2%	96.3%	96.5%	96.5%	**96.8%**	96.4%	**-1.2%**
	Truth	81.7%	81.7%	81.9%	81.8%	**81.9%**	*80.7%*	78.5%	-
Russian	Iter. 1(a)	69.3%	69.3%	69.3%	69.5%	69.7%	68.9%	**70.1%**	-4.5%
	Iter. 3(a)	89.9%	89.9%	89.9%	89.9%	90.0%	**90.3%**	90.3%	-3.2%
	Iter. 5(a)	93.3%	93.3%	93.4%	93.4%	93.6%	93.7%	**93.8%**	**-4.5%**
	Truth	**85.5%**	85.5%	85.3%	85.2%	84.9%	82.3%	*81.0%*	-
German	Iter. 1(a)	92.0%	92.1%	92.4%	92.6%	93.6%	95.2%	**97.3%**	-0.2%
	Iter. 3(a)	96.7%	96.7%	96.9%	96.9%	97.1%	97.3%	**97.4%**	-0.2%
	Iter. 5(a)	97.6%	97.6%	97.8%	97.9%	98.2%	98.4%	**98.5%**	**-0.2%**
	Truth	91.8%	91.8%	91.9%	91.9%	**92.1%**	91.9%	*91.9%*	-
Turkish	Iter. 1(a)	82.8%	82.9%	82.9%	83.1%	82.8%	82.5%	**84.2%**	-5.7%
	Iter. 3(a)	99.4%	99.4%	99.4%	**99.4%**	**99.4%**	99.4%	99.2%	-0.3%
	Iter. 5(a)	99.8%	99.8%	99.8%	99.8%	**99.8%**	99.8%	99.5%	**-0.3%**
	Truth	95.2%	95.2%	95.1%	**95.3%**	*95.0%*	94.1%	89.6%	-
Basque	Iter. 1(a)	**66.2%**	57.2%	53.5%	53.9%	55.2%	56.5%	62.2%	-4.6%
	Iter. 3(a)	91.1%	91.8%	91.9%	**91.9%**	91.8%	91.6%	89.3%	-2.5%
	Iter. 5(a)	94.5%	**94.8%**	94.3%	94.4%	94.3%	93.9%	91.2%	**0.0%**
	Truth	79.2%	**83.8%**	81.8%	81.3%	80.8%	78.7%	73.9%	-
English	Iter. 1(a)	80.9%	80.9%	80.9%	81.2%	81.6%	82.3%	**83.4%**	-9.5%
	Iter. 3(a)	98.2%	**98.3%**	98.3%	98.3%	**98.3%**	98.2%	98.0%	0.0%
	Iter. 5(a)	99.0%	99.0%	99.0%	99.0%	**99.1%**	99.0%	98.8%	**-1.1%**
	Truth	**94.6%**	**94.6%**	94.6%	94.5%	*93.5%*	90.0%	84.9%	-

Table 5.10: Estimation of the most effective prefix penalty for Levenshtein similarity

5.4 Retraining the supervised models

The primary goal of iterative retraining is to refine the core morphological transformation model of Chapter 3, which not only serves as one of the four similarity models, but is also a primary, final, stand-alone deliverable of the learning process.

As subsequent iterations proceed, the supervised probability models are retrained on the output of the prior iteration and filtering out analyses without a minimum level of support[3] from training data to help reduce noise. The final analysis probabilities are then interpolated with the retrained similarity models from Chapter 4.

5.4.1 Choosing the training data

Once the parameters to the unsupervised models have been selected as above and the candidate rankings of each model are combined by rank, the inflection-root pairs from the unsupervised models must be selected for use as training data for the supervised models. Two methods for selecting these data were evaluated. In the first method, only the root which was the highest ranked candidate for each inflection was chosen. In the second method, the top two candidates were used.

The motivation behind using the top two candidate roots instead of only the single best candidate was two-fold. First, since the analysis methods are often noisy, the best single candidate is not always correct. Using the top two candidates instead of only the single best one allows more potentially correct examples (but also more incorrect examples) to be included in the training data. Second, having competing training pairs used in the

[3]This minimum is a threshold set at conservative 0.05

supervised methods alleviates problems of over-fitting to the potentially misaligned forms. By providing a competitor for the same inflection, there is less tendency to memorize the singleton choice, and greater tendency to favor the analysis most consistent with the overall current trie models.

In nearly every language, based on tests after both Iteration 3(b) and Iteration 4(b), training on the top 2 inflection-root candidate pairs from the unsupervised model improved performance for the supervised Base model but hurt performance for the WFBase and WFAffix models. The Affix model had mixed performance at Iteration 3(b), but showed a strong preference for using the top 2 analyses at Iteration 4(b). (Table 5.11)

Because the point-of-affixation changes are stored in a smoothed trie, both the Base and the Affix models are quite robust in the presence of noisy training data. While the WFBase and WFAffix models also store their point-of-affixation changes in a trie, they do not store the probabilities of the allowable internal vowel change based on context. As mentioned in Section 3.5.1, the probability for each vowel changes is derived from the normalized *but unsmoothed* counts of the vowel changes seen in training data. In addition, the vowel change probabilities are not sensitive to the contexts in which they were found. For this reason, the amount of noise introduced when including the second most likely analysis plays havoc with the vowel change probabilities which leads to substantially reduced performance for the WFBase and WFAffix models.

5.4.2 Weighted combination the supervised models

As each step (b) in the iteration, all four supervised models are run on the training data provided by the unsupervised similarity models. As previously shown in Table 3.35,

	Base Model		Affix Model		WFBase Model		WFAffix Model	
Iteration 3(b)	Top 1	Top 2	Top 1	Top 2	Top 1	Top 2	Top 1	Top 2
English	**97.8%**	97.6%	**92.3%**	85.9%	**91.3%**	82.3%	**85.7%**	75.9%
Russian	**75.8%**	75.7%	**75.8%**	75.7%	**73.3%**	65.5%	**73.3%**	65.5%
Estonian	80.7%	**81.6%**	**84.1%**	81.6%	**72.2%**	61.0%	**70.4%**	61.2%
German	88.9%	**89.2%**	88.9%	**89.2%**	**85.8%**	74.6%	**86.5%**	75.5%
Turkish	98.9%	**99.1%**	99.0%	**99.1%**	**96.0%**	89.5%	**96.6%**	82.2%
Portuguese	96.7%	**97.3%**	96.8%	**97.1%**	**83.9%**	76.4%	**94.4%**	87.2%
Basque	89.9%	**91.2%**	**89.4%**	88.9%	55.2%	**64.5%**	**75.0%**	55.0%
Iteration 4(b)	Top 1	Top 2	Top 1	Top 2	Top 1	Top 2	Top 1	Top 2
English	97.8%	**97.9%**	94.0%	**97.5%**	**95.7%**	90.6%	**89.8%**	86.5%
Russian	75.3%	**77.8%**	75.3%	**77.8%**	**72.7%**	68.4%	**72.7%**	68.4%
Estonian	82.3%	**83.1%**	**87.5%**	87.4%	**76.1%**	64.1%	**75.6%**	62.0%
German	89.5%	**90.5%**	89.5%	**90.5%**	**89.6%**	78.8%	**90.0%**	79.8%
Turkish	99.1%	**99.2%**	99.1%	**99.2%**	**98.8%**	91.2%	**98.8%**	93.6%
Portuguese	97.4%	**97.5%**	97.3%	**97.4%**	**96.8%**	86.5%	**97.3%**	75.7%
Basque	92.6%	**93.2%**	91.7%	**92.4%**	**91.6%**	87.0%	**89.6%**	79.5%

Table 5.11: Sensitivity of supervised models to training data selection. These results show the performance difference between running the various supervised models on either the single best output of the unsupervised models (Top 1) or using the top two choices (Top 2).

combining the models trained from clean exemplars (the fully supervised case) results in an increase in performance relative to choosing the single best model.

By contrast, when bootstrapping in the absence of supervised training data, according to Table 5.2, only four of the seven languages showed an increased performance over the single best model when weighting each of the models equally. Of the remaining three languages, Turkish showed a slight preference for the Base model and Estonian showed a slight preference for the Affix model when using the Top 2 data set for training. In English, the Base model outperformed the combined models for both the Top1 and Top2 training sets. This is due in large part to the fact that modeling internal vowel shifts is not required for a large portion of English morphology and the noise problems of the WFBase

Language	Data	Base	Affix	WFBase	WFAffix	Equally Combined	Diff.
English	Top 1	**97.8%**	94.0%	95.7%	89.8%	95.3%	-2.5%
	Top 2	**97.9%**	97.5%	90.6%	86.5%	97.5%	**-0.4%**
Estonian	Top 1	82.3%	**87.5%**	76.1%	75.6%	87.1%	-0.4%
	Top 2	83.1%	**87.4%**	64.1%	62.0%	86.4%	**-1.0%**
Russian	Top 1	75.3%	75.3%	72.7%	72.7%	**78.8%**	0.0%
	Top 2	77.8%	77.8%	68.4%	68.4%	**78.9%**	**0.0%**
German	Top 1	89.5%	89.5%	89.6%	90.0%	**93.6%**	0.0%
	Top 2	90.5%	90.5%	78.8%	79.8%	**93.3%**	**0.0%**
Turkish	Top 1	**99.1%**	**99.1%**	98.8%	98.8%	98.9%	-0.2%
	Top 2	**99.2%**	**99.2%**	91.2%	93.6%	99.1%	**-0.1%**
Portuguese	Top 1	97.4%	97.3%	96.8%	97.3%	**97.6%**	0.0%
	Top 2	97.5%	97.4%	86.5%	75.7%	**97.6%**	**0.0%**
Basque	Top 1	**92.6%**	91.7%	91.6%	89.6%	**92.6%**	0.0%
	Top 2	93.2%	92.4%	87.0%	79.5%	**93.8%**	**0.0%**

Table 5.12: Accuracies of each the models on their own, and combined using an unweighted average of the scores of each model. Accuracies were taken from Iteration 4(b) and tested on both the models trained from the Top 1 and Top 2 analyses derived from Iteration 3(b).

and WFAffix models hurt performance too much in the combined model.

This raises the question of whether or not the weight given to each of the various supervised models when combining them can be learned in the same way unequal weighting of that the parameter space for the context and Levenshtein similarities were learned.

In other words, the weights for the supervised models on each iteration (b) were chosen to maximize the model's success at reproducing the outputs of the unsupervised models at the previous (a) iteration.

To test this, the context similarity model was chosen as the estimate for the truth. Although the Levenshtein similarity measure is, by itself, more accurate than the context similarity model,[4] Levenshtein was not chosen as the estimate for the truth when selecting

[4]See Chapter 4 for more details.

the supervised model candidates. Because both the Levenshtein models and the supervised models use orthographic information in determining analysis probabilities, there was not sufficient independence between these measures for Levenshtein to be used to more effectively estimate the goodness of the supervised methods.

Because the context similarity model does not often choose the correct candidate root as its single highest ranked analysis, accuracy was not used to gauge the goodness of the model. Instead, the model was chosen that had the highest score defined by:

$$\begin{aligned} score_{combined}(model) = & \frac{1}{8} score_{rank}(model) + \frac{1}{8} score_{logrank}(model) + \\ & \frac{1}{8} score_{modRank}(model) + \frac{1}{8} score_{score}(model) + \\ & \frac{1}{2} score_{correct}(model) \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} score_{rank}(model) &= \sum_a \frac{1}{rank(a)} \\ score_{logrank}(model) &= \sum_a \frac{1}{log(rank(a))} \\ score_{modRank}(model) &= \sum_a \frac{1}{modRank(a)} \\ score_{score}(model) &= \sum_a score(a) \\ score_{correct}(model) &= \sum_a correct(a) \end{aligned} \tag{5.2}$$

where a is the most likely analysis *for each inflection* according to each supervised model, $rank(a)$ is the rank of this first choice candidate from the context similarity model, $score(a)$ is the cosine similarity score of this analysis, and $correct(a)$ is $\frac{1}{k}$ (where k is the number of

supervised models) if the analysis was in the top 10 analyses deemed by context similarity. The function $modRank(a)$ was computed as follows: for each model, compute $rank(a)$, then sort these ranks relative to the other models. The model with the highest $rank(a)$ relative to the other models receives $\frac{1}{1}$ points, the model with the second highest $rank(a)$ receives $\frac{1}{2}$ points, etc. The points are then normalized (so that the total points assigned for each inflection is 1) and assigned to $modRank(a)$.[5]

The combination of scores was chosen without experimentation. As Table 5.14 illustrates, the very high agreement rates between the scoring methods somewhat mitigates any biases in the initial score combination.

The results of this combination can be found in Table 5.13. With only one exception each, the accuracy of the equally weighted combination was always equal or better in Iteration 3(b), and the accuracy of the performance-based weighting was always equal or better in Iteration 4(b). Although a full error analysis has not been done, initial analysis suggests that this difference is due to false matches between the context similarity and supervised methods.

In particular, at Iteration 3(b), the supervised models have been trained directly on the output of the unsupervised models. This means that to a large degree, there will be significant correlation between the supervised models and the unsupervised context model on which it is being graded. Because of this correlation, the scoring methods find many examples of "false positives", or examples where the context similarity and supervised models agree, but are both wrong. This miscoring ends up overweighting the wrong models;

[5]Essentially, the top rated system according to context similarity receives 0.48, the second system receives 0.24, the third system receives 0.16, and the final system receives 0.12.

Language	Iteration	Top 1 Training		Top 2 Training	
		Equally Weighted	Performance Weighted	Equally Weighted	Performance Weighted
English	Iter. 3(b)	**95.2%**	95.1%	93.1%	**93.7%**
	Iter. 4(b)	95.3%	**95.9%**	97.5%	97.5%
Estonian	Iter. 3(b)	**85.4%**	84.8%	**85.9%**	85.6%
	Iter. 4(b)	87.1%	**87.2%**	86.4%	**86.7%**
Russian	Iter. 3(b)	81.2%	81.2%	76.8%	76.8%
	Iter. 4(b)	**78.8%**	78.7%	78.9%	78.9%
German	Iter. 3(b)	**93.6%**	93.1%	92.3%	92.3%
	Iter. 4(b)	93.6%	93.6%	93.3%	93.3%
Turkish	Iter. 3(b)	**98.0%**	97.4%	**98.7%**	98.6%
	Iter. 4(b)	98.9%	98.9%	99.1%	99.1%
Portuguese	Iter. 3(b)	**97.5%**	97.4%	97.5%	97.5%
	Iter. 4(b)	97.6%	97.6%	97.6%	**97.7%**
Basque	Iter. 3(b)	**91.2%**	90.9%	90.5%	90.5%
	Iter. 4(b)	92.6%	92.6%	93.8%	**93.9%**

Table 5.13: Accuracy of the equally weighted model vs. the accuracy of the performance-weighted model taken from Iterations 3(b) and 4(b), and tested on both the models trained from the Top 1 and Top 2 analyses derived from the previous iteration. With only one exception each, the equal combination is as good or better at Iteration 3(b) than the performance-based combination, and the performance-based combination is as good or better at Iteration 4(b) than the equal combination.

hence, the conservative equal weighting combination outperforms the performance-based weighting.

At Iteration 4(b), when the training data is more consistent (since it is trained on the output of the supervised model at Iteration 3(b)), there are fewer "false positives" and so the context models serve well as estimators for the combination weights.

Table 5.14 shows the output of each the individual scoring functions as well as the final combined score for Estonian, English and Portuguese. At Iteration 3(b), the combined weights underweight models with relatively high accuracy (Estonian Base model), overweight models with relatively low accuracy (English Affix model) or fail to distinguish

ESTONIAN

Scoring Method	Iteration 3(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}	Iteration 4(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}
$score_{rank}$	0.25	**0.33**	0.19	0.23	0.27	**0.33**	0.16	0.24
$score_{logrank}$	0.25	**0.29**	0.22	0.24	0.26	**0.29**	0.21	0.24
$score_{rankF}$	0.22	**0.28**	0.25	0.25	0.23	**0.28**	0.23	0.26
$score_{score}$	0.23	**0.32**	0.19	0.26	0.25	**0.33**	0.17	0.25
$score_{correct}$	0.23	**0.35**	0.19	0.23	0.26	**0.35**	0.15	0.25
$score_{combined}$	0.23	**0.32**	0.20	0.24	0.26	**0.33**	0.17	0.25
$score_{combined-truth}$	**0.35**	0.34	0.15	0.16	0.34	**0.37**	0.15	0.13
System Accuracy	81.6%	**81.6%**	61.0%	61.2%	83.1%	**87.4%**	64.1%	62.0%

ENGLISH

Scoring Method	Iteration 3(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}	Iteration 4(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}
$score_{rank}$	0.30	**0.30**	0.22	0.17	0.34	**0.34**	0.20	0.12
$score_{logrank}$	0.27	**0.27**	0.24	0.22	0.29	**0.29**	0.23	0.19
$score_{rankF}$	0.25	**0.26**	0.25	0.24	0.26	**0.27**	0.24	0.23
$score_{score}$	0.22	**0.30**	0.24	0.24	0.29	**0.31**	0.22	0.18
$score_{correct}$	**0.32**	**0.32**	0.21	0.16	0.36	**0.36**	0.17	0.11
$score_{combined}$	0.29	**0.30**	0.22	0.19	0.33	**0.33**	0.20	0.14
$score_{combined-truth}$	**0.41**	0.26	0.21	0.12	**0.37**	0.36	0.19	0.09
System Accuracy	**97.6%**	85.9%	82.3%	75.9%	**97.9%**	97.5%	90.6%	86.5%

PORTUGUESE

Scoring Method	Iteration 3(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}	Iteration 4(b) λ_{Base}	λ_{Affix}	λ_{WFB}	λ_{WFA}
$score_{rank}$	**0.27**	0.27	0.22	0.24	**0.31**	0.31	0.24	0.14
$score_{logrank}$	**0.26**	0.26	0.24	0.25	**0.28**	0.28	0.24	0.20
$score_{rankF}$	0.24	0.24	**0.26**	0.26	**0.26**	0.26	0.25	0.23
$score_{score}$	0.22	0.22	**0.30**	0.27	**0.28**	0.28	0.28	0.16
$score_{correct}$	**0.27**	0.27	0.22	0.25	0.31	**0.31**	0.24	0.13
$score_{combined}$	**0.26**	0.26	0.24	0.25	0.30	**0.30**	0.25	0.16
$score_{combined-truth}$	**0.34**	0.33	0.10	0.22	**0.34**	0.34	0.22	0.10
System Accuracy	**97.3%**	97.1%	76.4%	87.2%	**97.5%**	97.4%	86.5%	75.7%

Table 5.14: Estimating performance-based weights for supervised model combination. Results are presented using the Iteration 3(a) re-estimated context similarity measure to score the Iteration 3(b) and 4(b) supervised models as presented in Equation (5.2). The results of the combination are shown in Table 5.13. Also presented are results for $score_{combined-truth}$ which computes the combined score for these weights as graded on the truth, and the actual system performance graded on the truth. The λ_i's are computed as: $\lambda_i = \frac{score_{combined}(i)}{\sum_j score_{combined}(j)}$

between models with large performance differences (all Portuguese models). At Iteration 4(b), this never happens: the combination weights reflect the relative performance of each of the systems well.

It should be noted that Table 5.2, or any of the previous results in this section, do not utilize the unequal weighting of these models – only the results in this section. This was done because, no development set (evaluation data held out from standard training and/or evaluation) was allocated and, although the performance was improved, there was no way to know this a priori. Therefore, in order to be experimentally honest, the final results (presented in Table 5.16) use *only* the equally weighted combination for all iterations, including Iteration 4(b). With more aggressive inductive bias, the performance-based voting at Iteration 4(b) should be chosen.

5.5 Final Consensus Analysis

As described in Table 5.2, the experiments presented here used a five step iterative pipeline. All of the details of this pipeline have been explained thoroughly except the final combination of supervised and unsupervised systems into the final analysis, which is done using the decision tree shown in Figure 5.2.

At Iteration 5(a), the parameters for the unsupervised models are once again re-estimated and combined using the rank-based combination discussed previously. Then, in Iteration 5(c), for every inflection-root analysis proposed by the supervised methods which is above a threshold,[6] the highest ranked combined choice of the unsupervised methods is

[6]For these experiments, a conservative threshold of 0.05 was used.

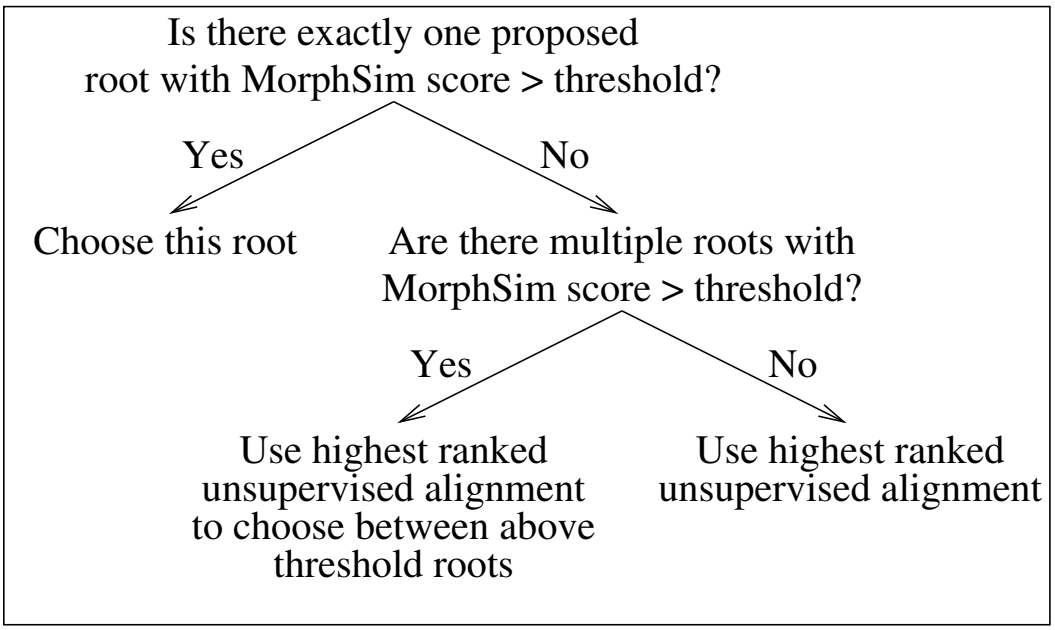


Figure 5.2: Using a decision tree to final combination

chosen. If there is only one such candidate above threshold, it is chosen by default. If there are no candidate analyses from the supervised model compatible with the ___ template, the highest ranked unsupervised consensus candidate is used.

5.5.1 Using a backoff model in fully supervised analyzers

Table 5.16 includes the column "Supervised with Backoff". As was seen first in Table 3.36, the precision of the supervised systems are extremely high. The cost of this precision is that the coverage for some languages (with limited supervised training data or agglunation, for example) is unacceptably low. To alleviate this problem, another model (e.g. by Table 5.2's procedure) can be used as a backoff model in the same way as in Figure 5.2. Including this backoff increases the coverage to 100% and improves overall

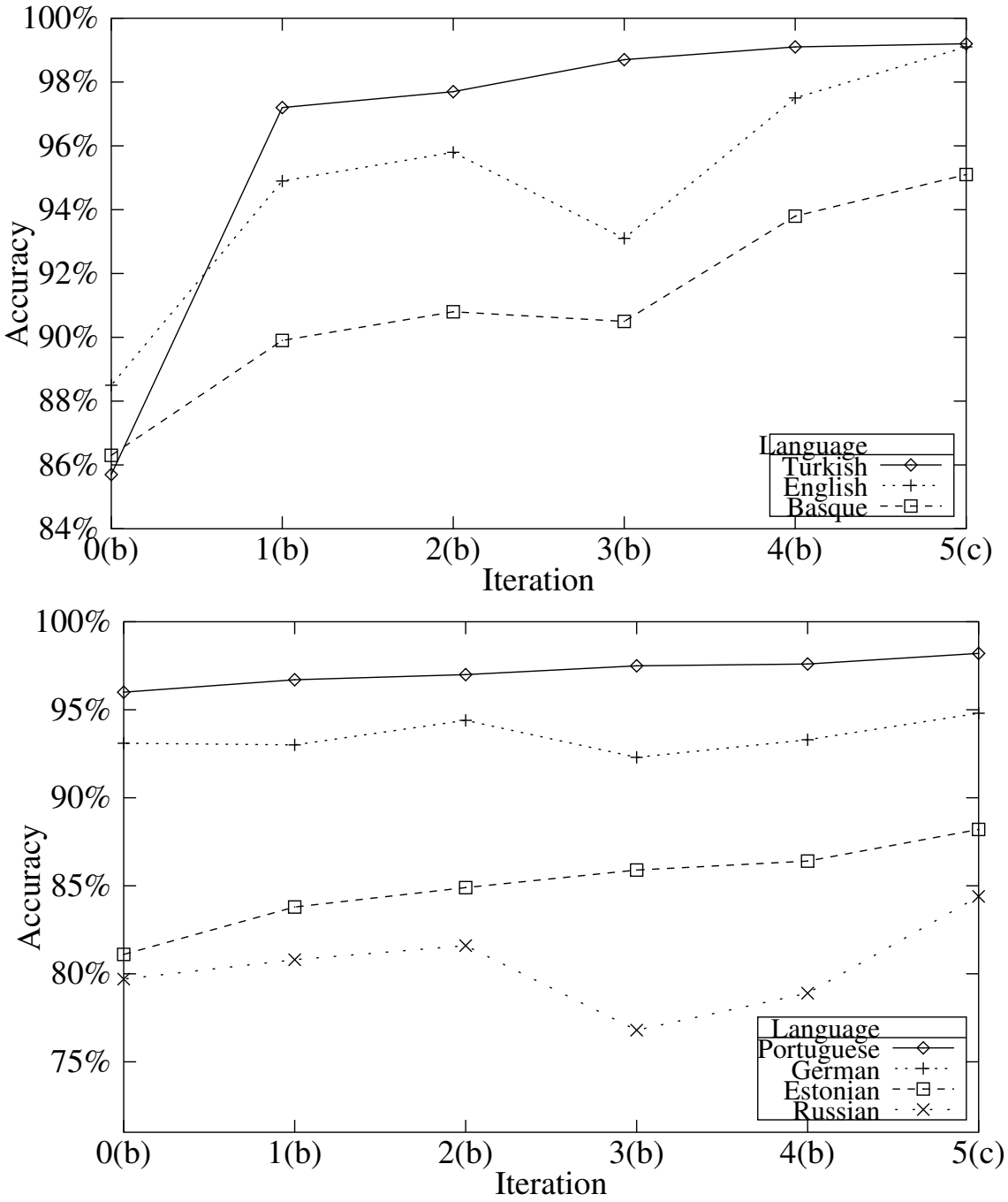


Figure 5.3: Performance increases from Iteration 0(b) to Iteration 5(c). The performance drop seen in many languages at Iteration 3(b) is due to the training on noisier data (the output of 3(a)) than 2(b) or 4(b) which train on the cleaner output of 1(b) and 3(b), respectively. Russian, which poorly estimates the unsupervised parameters (see Tables 5.3 through 5.10) does more poorly at Iteration 3(b) than at 1(b) which carries over into a loss of accuracy from 2(b) to 4(b).

Language	All	Regular	Semi-Regular	Irregular	Obsolete/ Other
English	99.05%	99.41%	99.50%	54.84%	100.0%
Portuguese	98.20%	98.31%	-	83.33%	20.00%
German	94.76%	97.83%	96.31%	75.60%	96.81%
Basque	95.14%	-	-	-	-
Russian	84.42%	-	-	-	-
Estonian	88.18%	-	-	-	-
Turkish	99.19%	99.96%	97.33%	19.35%	-

Table 5.15: Accuracy at Iteration 5(c) on different types of inflections (where classification labels were available).

accuracy as shown in Figure 5.16.

This was backoff was done using the output of 5(a) as the backoff model. There are other ways to do this, too. One way is to back off to 5(c); another way is to get a better model than 5(a) or 5(c) by replacing iteration 0(b) with the limited supervised data so the models gets off to the right start. But even in here, this result is only used for backoff and will not override the supervised model.

5.6 Bootstrapping from BridgeSim

As was seen in Chapter 4, the Multilingual Bridge Similarity measure is quite accurate as a stand-alone analyzer. Unfortunately, on its own, this measure is limited to analyzing only those forms found with sufficient frequency in an aligned bitext. Using methods similar to those described in the Section 5.5, the output of the BridgeSim measure can be used as input for the supervised methods which can then be in conjuction with a backoff model, as in Section 5.5.1.

Tables 5.17, 5.18, and 5.19 show the performance of BridgeSim pipeline which

	UNSUPERVISED MORPHOLOGICAL ANALYSIS							
	Iteration						Fully	Supervised
Language	0(b)	1(b)	2(b)	3(b)	4(b)	5(c)	Supervised	with Backoff
English	88.5%	94.9%	95.8%	93.1%	97.5%	99.1%	99.1%	99.5%
Portuguese	96.0%	96.7%	97.0%	97.5%	97.6%	98.2%	97.9%	98.9%
German	93.1%	93.0%	94.4%	92.3%	93.3%	94.8%	97.9%	98.3%
Basque	86.3%	89.9%	90.8%	90.5%	93.8%	95.1%	96.0%	97.4%
Russian	79.7%	80.8%	81.6%	76.8%	78.9%	84.4%	90.8%	93.3%
Estonian	81.1%	83.8%	84.9%	85.9%	86.4%	88.2%	96.8%	98.3%
Turkish	85.7%	97.2%	97.7%	98.7%	99.1%	99.2%	99.5%	99.8%

Table 5.16: Results of five iterations of the unsupervised pipeline compared against fully supervised methods. The "Fully Supervised" column is equivalent to the final column of Table 3.35. The final column "Supervised with Backoff" uses the decision tree in Figure 5.2 with the supervised model of the previous column and backoff to the re-estimated unsupervised models at Iteration 4(b). Note that for English, Portuguese and, to a weaker extent, Turkish, the fully unsupervised models with multiple similarity measures match or even exceed the performance of the fully supervised models.

starts from only the morphologically projected forms from BridgeSim (MProj), then uses this as seed data to train the supervised trie-based models (MProj+MTrie) and then finally incorporates a backoff model (BKM) for any remaining unanswered forms or ambiguities. No iteration is done in these experiments, but certainly the output of BridgeSim *could* be substituted for 0(b) in Table 5.2.

CZECH Verbal Morphology Induction

Model	Precision		Coverage	
	Typ	Tok	Typ	Tok
Czech Reader's Digest (500K words):				
MProj only	.915	.993	.152	.805
MProj+MTrie	.916	.917	.893	.975
MProj+MTrie+BKM	**.878**	**.913**	1.00	1.00

Table 5.17: Performance of full verbal morphological analysis, including precision/coverage by type/token

SPANISH Verbal Morphology Induction

Model	Precision		Coverage	
	Typ	Tok	Typ	Tok
Spanish Bible (300K words) via 1 English Bible:				
MProj only	.973	.935	.264	.351
MProj+MTrie	.988	.998	.971	.967
MProj+MTrie+BKM	**.966**	**.985**	1.00	1.00
Spanish Bible (300K words) via French Bible:				
MProj only	.980	.935	.722	.765
MProj+MTrie	.983	.974	.986	.993
MProj+MTrie+BKM	**.974**	**.968**	1.00	1.00
Spanish Bible (300K words) via 3 English Bibles:				
MProj only	.964	.948	.468	.551
MProj+MTrie	.990	.998	.978	.987
MProj+MTrie	**.976**	**.987**	1.00	1.00

Table 5.18: Performance on Spanish inflections bootstrapped from the Bridge Similarity model

FRENCH Verbal Morphology Induction

	Precision		Coverage	
Model	Typ	Tok	Typ	Tok
French Hansards (12M words):				
MProj only	.992	.999	.779	.994
MProj+MTrie	.998	.999	.988	.999
MProj+MTrie+BKM	**.994**	**.999**	1.00	1.00
French Hansards (1.2M words):				
MProj only	.985	.998	.327	.976
MProj+MTrie	.995	.999	.958	.998
MProj+MTrie+BKM	**.979**	**.998**	1.00	1.00
French Hansards (120K words):				
MProj only	.962	.931	.095	.901
MProj+MTrie	.984	.993	.916	.994
MProj+MTrie+BKM	**.932**	**.989**	1.00	1.00
French Bible (300K words) via 1 English Bible:				
MProj only	1.00	1.00	.052	.747
MProj+MTrie	.991	.998	.918	.992
MProj+MTrie+BKM	**.954**	**.994**	1.00	1.00
French Bible (300K words) via 3 English Bibles:				
MProj only	.928	.975	.100	.820
MProj+MTrie	.981	.991	.931	.990
MProj+MTrie+BKM	**.964**	**.991**	1.00	1.00

Table 5.19: Performance of full verbal morphological analysis, including precision/coverage by type/token

Chapter 6

Conclusion

6.1 Overview

This dissertation has presented a comprehensive original framework for the supervised and unsupervised machine learning of inflectional computational morphology. These models are evaluated over data sets in 32 languages, and an extensive variety and range of both data dimensions and model parameter dimensions are systematically and contrastively explored. These studies have yielded valuable insights into the nature of inflectional morphological phenomena and the efficacy of diverse models and model parameters for handling this phenomena.

When morphological training data is available, the supervised methods presented in Chapter 3 are able to produce highly accurate lemmatizations of inflections to their respective roots. When training data is not available, the unsupervised similarity methods presented in Chapter 4 are capable of producing noisy alignments which can be used, as in Chapter 5, to bootstrap parameters of the supervised methods. Together, the supervised

and unsupervised learners can be iteratively retrained to create models of lemmatization which can outperform the supervised methods trained on clean training exemplars for some sets of languages (See Table 5.16).

6.1.1 Supervised Models for Morphological Analysis

Four models of supervised morphological analysis were presented in Chapter 3: the Base model, the Affix model, the WFBase model, and the WFAffix model. Each model is successively more complex, capable of modeling more inflectional phenomena than the simpler model before it.

The first model, the Base model (Section 3.3), treats the process by which an inflection is transformed into a root as a word-final string rewrite rule. In this model, these rewrite rules are stored in a smoothed trie such that these rules are sensitive to the final letters of the inflection. Although the Base model cannot handle any prefixation, and is limited in its ability to capture point-of-suffixation changes which are consistent across multiple inflections of the same root, the Base model is remarkably accurate across a broad range of the evaluated languages. (Tables 3.14 and 3.15)

The second model, the Affix model (Section 3.4), makes use of user-supplied lists of prefixes, suffixes and canonical root endings to model suffixation as a separate process from that of point-of-suffixation change. If appropriate prefixes are given in these user-supplied lists, the Affix model also provides support for inflections with purely concatenative prefixation. The Affix model consistently outperformed the Base model in supervised learning, as shown in direct comparison in Table 3.25.

The final two models are based on the introduced notion of a Wordframe which

allows for internal vowel changes, point-of-prefixation, as well as point-of-suffixation chages, and has the ability to run either without user-supplied affix lists (WFBase) or with them (WFAffix). Table 3.32 presents a direct comparison between all four models.

Across the 32 languages over which these models were evaluated, there was no single model which performed better for all, or even most, of the languages. Thus, one successful conservative approach is to perform a direct unweighted average of the inflection-root analysis scores produced by each of the 4 models, or to conduct performance-weighted voting over these models. Doing so typically increases performance over any one component model and over all pairwise model combinations, illustrating the differences in the information captured by each model. Table 3.35 presents results for this combination.

6.1.2 Unsupervised Models for Morphological Alignment

Since morphologically annotated or paired inflection-root training data is not often available for a given language in which morphological analysis is to be done, unsupervised or minimally supervised learning is extremely useful and essential for rapid broad language coverage.

Four novel, independent, orthogonal similarity measures are introduced as the central information sources for this induction. The colletive use of these models for morphological analysis represents a paradigm shift from prior approaches to computational morphology in that they do not focus on string characteristics to determine whether a particular root is a good fit for an inflection.

The Frequency similarity model (Section 4.3) uses word counts derived from a corpus to align inflections with potential roots. The motivation behind using such a measure

was the intuition that inflections which occur with high frequency in a corpus should have roots which also occur with high frequency, and that inflections which occur with low frequency should have roots which occur with low frequency.

The Context similarity model (Section 4.4) uses information about the context in which an inflection and its potential roots are found in an unannotated corpus to derive noisy inflection-root alignments. A thorough evaluation of the parameter space of this model is performed: Tables 4.11 through 4.16 show that under certain initial parameterizations, the Context similarity model can be reasonably accurate at isolating a small set of candidate roots for each inflection.

The Levenshtein similarity model uses a weighted variant of the Levenshtein distance to model the orthographic distance between an inflection and a candidate root. This enhanced model of Levenshtein can perform substitutions on clusters of letters, in addition to the standard substitutions on individual letters. As with the Context similarity model, a thorough evaluation of the parameter space was investigated (Tables 4.21 through 4.28) and, as with the Context model, the Levenshtein model was shown to be particularly sensitive to intial model parameters.

Both the intial parameters to the Levenshtein model and the initial parameters to the Context model were iteratively re-trained in Chapter 5.

The final unsupervised similarity model is the Translingual Bridge Similarity model (Section 4.6). This model used a word-alinged bilingual corpus between a language for which a morphological analyzer already exists and a language for which one wishes to perform morphological analysis on. The precision of the Translingual Bridge model is remarkably

high (Table 4.29), though the low coverage across the space of inflections means that this model cannot be used as a stand-alone morphological analyzer.

6.2 Future Work

6.2.1 Iterative induction of discriminative features and model parameters

The model combinations presented in Chapter 5 illustrate the effectiveness of iteratively retraining the parameters of both the unsupervised models as well as the weights associated with combining the supervised models. Not yet explored was the iterative retraining of the parameters of the supervised models. The value of λ_i which served to as the smoothing parameter used for backoff in the trie should be amenable to iterative training, though no exploration into this has yet been attempted.

Additionally, the weighting function $\omega(root)$ should be iteratively retrained such that it maximizes the precision and accuracy of the models in which it is applied.

Additional feature templates and evidence sources may prove to be useful in the models evaluated here. Techniques for large scale search of potential new feature templates and feature combinations is a potentially worthiwhile avenue for model improvement.

6.2.2 Independence assumptions

Many independence assumptions were made when developing the Affix, WFAffix, and WFBase models of Chapter 3. It is likely that many of these simplifying assumptions were detrimental to the overall performance of the models.

As an example, the WFAffix and WFBase models use the unsmoothed, raw counts

of observed internal vowel changes as the probability for applying a vowel change to a test form. The probabilities of these internal vowel changes were not conditioned on either the local context or on the position in the string where this change was observed. This caused the WFAffix and WFBase models to be much more sensitive to noise than then Base and Affix models.

As a second example, the Affix and WFAffix models do not conditionalize the probability of applying the canonical ending on the suffix that was removed from the inflection. The way in which affixes can be clustered into paradigms based solely on the canonical ending in languages such as French and Spanish is a clear indication that this dependency exists for some languages.

In addition, the removal of an affix from the inflection is not currently not conditionalized on the resulting point-of-affixation change. Currently, the affix chosen at training time is determined by a ranked ordering of the affixes based on string length. As mentioned in Section 3.4.3, there are three other potential ways to handle this. The first is to allow a human to provide this ranking, the second is to choose the affix which results in the simplest stem change, and the third is to give each analysis partial weight in the stem change counts.

Removing these independence assumptions from the various affected models should yield performance gains and is a high priority of planned future work.

6.2.3 Increasing coverage of morphological phenomena

While the models are reasonably effective at analyzing the agglutinative inflected forms found in Turkish, and the partially reduplicative and infixed inflected forms found in Tagalog, the seven-way split introduced in Section 3.2.1 was not designed to handle these

phenomena directly. Additionally, this seven way split was not evaluated on inflections exhibiting whole word reduplication or templatic morphologies. Building supervised models capable of inducing the necessary patterns to represent such phenomena is important step in being able to apply the methods presented here to languages which exhibit more complex inflectional schemes than the ones presented here.

6.2.4 Syntactic feature extraction

Nearly all of the experimental studies evaluation presented in this thesis focused on the task of lemmatization in inflectional morphology. A natural extension to this work involves extracting the syntactic features associated with each inflection. This would need be done either in conjuction with a part of speech tagger, a parser, or using the translingual information projection techniques described in Section 4.6.

6.2.5 Combining with automatic affix induction

Both the Affix model and the WFAffix model utilize an (often empty) set of canonical prefixes, suffixes and root endings as a component of model parameterization. The work of Goldsmith [2001] and Schone and Jurafsky [2001] both present unsupervised methods for automatically identifying candidates for these canonical affix sets. Additionally, the inflection-root pairs derived from these models can be used to iteratively retrain and improve the affix extraction systems, which should provide further benefit to the Affix and WFAffix models.

6.3 Summary

This dissertation has presented a successful original paradigm for both morphological analysis and generation by treating both tasks in a competitive linkage model based on a combination of diverse inflection-root similarity measures. Previous approaches to the machine learning of morphology have been essentially limited to string-based transduction models. In contrast, the work presented here integrates both several new noise-robust, trie-based supervised methods for learning these transductions, and also a suite of unsupervised alignment models based on weighted Levenshtein distance, position-weighted contextual similarity, and several models of distributional similarity including expected relative frequency. Through iterative bootstrapping, the combination of these models yields a full lemmatization analysis competitive with fully supervised approaches but without any direct supervision. In addition, this dissertation also presents an original translingual projection model for morphology induction, where previously learned morphological analyses in a second language can be robustly projected via bilingual corpora to yield successful analyses in the new target language without any monolingual supervision.

Collectively both these supervised and unsupervised methods achieve state-of-the-art performance on the machine learning of morphology. Investigating a diverse set of 32 languages across a representative subset of the world's language's families and morphological phenomena, this dissertation constitutes one of the largest-scale and most comprehensive studies of both the successful supervised and unsupervised multilingual machine learning of inflectional morphology.

Appendix A

Monolingual resources used

Language	Dictionary entries	Evaluation data						Corpus size (millions of words)	
		Verbs		Nouns		Adjs			
		Roots	Infls	Roots	Infls	Roots	Infls	plain text	tagged
Basque	33020	1185	5842	7851	24801	3511	7329	700K	700K
Catalan	0	103	4058	0	0	0	0	0	0
Czech	29066	5715	23786	36791	64088	12345	34239	1.3M	1.3M
Danish	51351	5197	1062	0	0	0	0	14M	14M
Dutch	41962	5768	1016	0	0	0	0	1.3M	0.9M
English	264075	1218	4915					118M	118M
Estonian	344	147	5932	220	5065	0	0	43M	0
Finnish	0	1434	79734	0	0	0	0	600K	0
French	27548	1829	63559	0	0	0	0	16.6M	16.6M
German	45779	1213	14120	0	0	0	0	19M	0
Greek	35245	9	201	6	28	0	0	14M	0
Hindi	0	15	255	0	0	0	0	0	0
Icelandic	0	314	3987	0	0	0	0	25M	0
Irish	0	54	1376	0	0	0	0	0	0
Italian	27221	1582	62658	416	488	466	466	46M	46M
Klingon	2114	699	5135	0	0	0	0	0	0
Norwegian	0	547	2489	0	0	0	0	0	0
Occitan	0	180	7559	0	0	0	0	0	0

Table A.1: Available resources (continued on next page)

Language	Dictionary entries	Evaluation data						Corpus size (millions of words)	
		Verbs		Nouns		Adjs			
		Roots	Infls	Roots	Infls	Roots	Infls	plain text	tagged
Polish	42005	601	23725	0	0	0	0	23M	0
Portuguese	30145	584	22135	0	0	0	0	4.5M	0
Romanian	25228	1070	24877	0	0	0	0	135K	135K
Russian	42740	191	3068	0	0	0	0	34M	0
Sanskrit	0	867	1968	0	0	0	0	0	0
Spanish	32895	1190	57224	1044	1844	3501	2811	58M	58M
Swahili	0	818	27773	0	0	0	0	450K	0
Swedish	46009	4035	13871	36193	53115	36193	53115	1.0M	1.0M
Tagalog	0	212	9479	0	0	0	0	4.0M	0
Turkish	25497	87	29130	0	0	0	0	85M	0
Uzbek	0	434	27296	0	0	0	0	1.8M	0
Welsh	0	1053	44295	0	0	0	0	0	0

Table A.2: Available resources (continued from previous page)

Bibliography

Y. Al-Onaizan, J. Curin, M. Jahn, K. Knight, J. Lafferty, D. Melamed, F.J. Och, D. Purdy, N. Smith, and D. Yarowsky. Statistical machine translation. Technical report, Johns Hopkins University, 1999.

M. Baroni, J. Matiasek, and T. Harald. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning*, pages 48–57, 2002.

M. R. Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–106, 1999.

M. R. Brent, S. K. Murthy, and A. Lundberg. Discovering morphemic suffixes: A case study in minimum description length induction. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, 1995.

P. Brown, J. Cocke, S. DellaPietra, V. DellaPietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16 (2):29–85, 1990.

A. Clark. Learning morphology with pair hidden markov models. In *Proceedings of the*

Student Workshop at the Annual Meeting of the Association of Computational Linguistics, pages 55–60, Toulouse, France, July 2001a.

A. Clark. Partially supervised learning of morphology with stochastic transducers. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, pages 341–348, Tokyo, Japan, November 2001b.

A. Clark. Memory-based learning of morphology with stochastic transducers. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 513–520, 2002.

S. Cucerzan and D. Yarowsky. Language independent minimally supervised induction of lexical probabilities. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, volume 38, 2000.

C. de Marcken. *Unsupervised language acquisition*. PhD thesis, MIT, 1995.

R. Florian and R. Wicentowski. Unsupervised italian word sense disambiguation using wordnets and unlabeled corpora. In *Proceedings of SigLEX'02*, pages 67–73, 2002.

J. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.

D. Hakkani-Tür, K. Oflazer, and G. Tür. Statistical morphological disambiguation for agglutinative languages. In *18th International Conference on Computational Linguistics*, 2000.

D. Jones and R. Havrilla. Twisted pair grammar: Support for rapid development of machine

translation for low density languages. In *Association for Machine Translation in the Americas*, pages 318–332, 1998.

F. Karlsson, A. Voutilainen, J. Heikkila, and A. Antilla, editors. *Constraint grammar: A language independent system for parsing unrestricted text*. Walter de Gruyter, 1995.

D. Kazakov. Unsupervised learning of naive morphology with genetic algorithms. In W. Daelemans, A. van den Bosch, and A. Weijtera, editors, *Workshop notes of the ECML/Mlnet Workshop on Empirical Learning of NLP Tasks*, 1997.

K. Koskenniemi. *Two-level morphology: A General Computational Model for Word-Form Recognition and Production*. PhD thesis, Department of Linguistics, University of Helsinki, Finland, 1983.

R. J. Mooney and M. E. Califf. *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, chapter Learning the past tense of English verbs using inductive logic programming, pages 370–384. Springer Verlag, 1995.

K. Oflazer and S. Nirenburg. Practical bootstrapping of morphological analyzers. In *Conference on Natural Language Learning*, 1999.

K. Oflazer, S. Nirenburg, and M. McShane. Bootstrapping morphological analyzers by combining human elicitation and machine learning. Technical report, Bilkent University, 2000. Computer Engineering Technical Report BU-CE-0003, January 2000.

S. Pinker and A. S. Prince. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28:73–193, 1988.

M. F. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3):130–137, 1980.

P. Resnik, M. Olsen, and M. Diab. The bible as a parallel corpus: annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33(1-2):129–153, 2000.

D. E. Rumelhart and J. L. McClelland. On learning the past tense of english verbs. In D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2, pages 216–271. MIT Press, Cambridge, MA, 1986.

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

P. Schone and D. Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop*, pages 67–72, Lisbon, 2000. Association for Computational Linguistics.

P. Schone and D. Jurafsky. Knowledge-free induction of inflectional morphologies. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*, 2001.

M. Snover and M. R. Brent. A bayesian model for morpheme and paradigm identification. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, volume 39, pages 482–490, 2001.

R. Sproat and D. Egedi. Connectionist networks and natural language morphology. In *Conference on Grammar and Language Processing*, 1988.

P. Theron and I. Cloete. Automatic acquisition of two-level morphological rules. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 103–110, 1997.

D. Wu. An algorithm for simultaneously bracketing parallel texts. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 244–251, 1995.

D. Wu. Statistical inversion transduction grammars an bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3), 1997.

D. Yarowsky, G. Ngai, and R. Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the Human Language Technology Conference*, pages 161–168, 2001.

D. Yarowsky and R. Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 207–216, 2000.

Vita

Richard Wicentowski lived his first 18 years in Freehold, New Jersey. He graduated from Manalapan High School in 1989, received his B.S. in Computer Science from Rutgers College in 1993, and completed his M.S. in Computer Science at the University of Pittsburgh in 1995. In September 2002, while successfully completing his Ph.D. at Johns Hopkins University, he joined the faculty of the Computer Science Department at Swarthmore College in Swarthmore, Pennsylvania.