Morphological Inference from Bitext for Resource-Poor Languages

by

Terrence D. Szymanski

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Linguistics)
in The University of Michigan
2011

Doctoral Committee:

Associate Professor Steven P. Abney, Chair
Professor Dragomir R. Radev
Professor Sarah G. Thomason
Assistant Professor Ezra R. Keshet

This thesis is dedicated to the memories of Daniel Szymanski, Traianos Gagos, and Richard Ward, without whom it would not have been possible.

ACKNOWLEDGEMENTS

First, I must thank my advisor, Steve Abney, for giving me the freedom and confidence to pursue the things that interest me, and for his unfailing ability to provide insight regardless of the subject. And thanks to the rest of my committe, Sally Thomason, Drago Radev, and Ezra Keshet. I've never learned so much from two courses as I did as a first-year grad student taking NLP with Drago and historical linguistics with Sally. I could not ask for four better models of teaching and scholarship.

For keeping my Fridays interesting, and for patiently enduring my own presentations, many thanks to the members of the CompLing and HistLing discussion groups: Rich Thomason, Ben Fortson, Bill Baxter, Yang Ye, Li Yang, Kevin McGowan, Stephen Tyndall, and too many others to name. I also feel fortunate to have had the opportunity to be a GSI for several excellent professors who inspired me to strive to become a better teacher: Andries Coetzee, John Lawler, and Bill Kretszchmar.

The corpus creation discussed in chapter 3 would not have been possible without the assistance of our annotators Greg Hughes, Laura Katsnelson, and Katie Voss. I also want to thank Ben King for his contributions to the project.

This material is based upon work partially supported by a Google Digital Humanities Research award. I am also indebted to the University of Michigan and the Department of Linguistics for their support throughout my graduate career.

This dissertation is typeset in LaTeX. Many thanks to Jason Gilbert and his predecessors who created the template and style files which helped me painlessly

meet Rackham's formatting requirements.

Thanks to my friends and fellow students, in particular David Medeiros, Autumn Fabricant, Lauren Squires, Matt Ides, Brad Boring, Stephen Tyndall, Joseph Tyler, Eric Brown, Kevin McGowan, and Jon Yip, for readily accompanying me in various intellectual, athletic, and social pursuits, and for making grad school much more than tolerable. And thanks to the classicists, in particular Drew Wilburn, James Cook, Bjorn Anderson, Nikos Litinas, and most of all Traianos Gagos, without whose mentorship I would most certainly not be where I am today.

Finally, I give my love and thanks to my family for their endless inspiration and support. Though they will not read this, I thank my father and my grandfather. I would not have begun, and could not have completed, this process without their support and approval. To Mom, Ben, Beth, and all the rest whom I love, thanks for everything. Thanks to Sophie for distracting me whether I needed it or not. And most importantly, thanks to Rachel, for your love, support, and smiles.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

CL computational linguistics

IGT interlinear glossed text

MDL minimum description length

NLP natural language processing

OCR optical character recognition

RPL resource-poor language

ABSTRACT

Morphological Inference from Bitext for Resource-Poor Languages

by

Terrence D. Szymanski

Chair: Steven P. Abney

The development of rich, multi-lingual corpora is essential for enabling new types of large-scale inquiry into the nature of language (Abney and Bird, 2010; Lewis and Xia, 2010). However, significant digital resources currently exist for only a handful of the world's languages. The present dissertation addresses this issue by introducing new techniques for creating rich corpora by enriching existing resources via automated processing.

As a way of leveraging existing resources, this dissertation describes an automated method for extracting bitext (text accompanied by a translation) from bilingual documents. Digitized copies of printed books are mined for foreign-language material, using statistical methods for language identification and word alignment to identify instances of English-foreign bitext. After parsing the English text and transferring this analysis via the word alignments, the foreign word tokens are tagged with English glosses and morphosyntactic features.

Tagged tokens such as these constitute the input to a new algorithm, presented in this dissertation, for performing morphology induction. Drawing on previous work on unsupervised morphology induction which uses the principle of minimum description

length to drive the analysis (Goldsmith, 2001), the present algorithm uses a greedy hill-climbing search to minimize the size of a paradigm-based morphological description of the language. The algorithm simultaneously segments wordforms into their component morphemes and organizes stems and affixes into a paradigmatic structure. Because tagged tokens are used as input, the morphemes produced by this induction method are paired with meaningful morphosyntactic features, an improvement over algorithms for unsupervised morphology based on monolingual text, which treat morphemes purely as strings of letters.

Combined, these methods for collecting and analyzing bitext data offer a pathway for the automatic creation of richly-annotated corpora for resource-poor languages, requiring minimal amounts of data and minimal manual analysis.

CHAPTER 1

Introduction

1.1 Introduction

For many speakers of English, and to a lesser extent, speakers of a handful of other major global languages, it would be difficult to imagine a life without language technology. No search engines, no text-message autocorrect, no annoying automated telephone help lines. Yet for the vast majority of the world's languages, even something as basic as a spell-checker is out of reach. There are estimated to be more than 6,000 languages spoken on Earth today, yet language processing software exists for less than 1% of these languages. Electronic data, but not software, is available for a slightly larger number of languages, but the vast majority of languages on earth can be categorized as resource poor: they lack any significant digital presence. This is a concern not only to speakers of those languages; indeed, many of these languages have very few speakers, if any, still living. This is a concern to everyone, particularly linguists, because every language which has ever been spoken by human beings has something unique to offer to the science of linguistics.

In this dissertation, I address some of the challenges associated with collecting and processing language data from resource-poor languages (RPLs). Research in natural language processing (NLP) and computational linguistics (CL), which has enabled a variety of new language technologies, has primarily focused on English and

a relatively small number of other widely-used languages. Much of this technology relies on large quantities of annotated data, such as million-word tagged corpora and syntactic treebanks, in order to be successful. These types of resources are expensive to produce, and that cost is part of the reason why NLP research is limited to the few languages for which such corpora have been created.

To make advances in language processing for resource-poor languages, there are two separate but related subproblems that must be addressed. The first problem is that digital data for these languages is either extremely scarce or non-existent, and therefore it is essential that whatever data does exist is collected and organized in a digital format that is suitable for machine processing. The second problem is that the standard statistical NLP techniques do not perform well on small data sets. Therefore, it is necessary to explore new ways to get the most out of the data that is available. In this dissertation, I address both of these problems. First, in chapter 3, I investigate whether existing language resources, such as printed grammars, can be automatically converted to a structured, electronic, machine-readable format. Scanned books are plentiful, but the challenge lies in extracting usable information from them, first by performing optical character recognition (OCR) on the scanned page images, then by identifying the structure of the material to extract, for instance, words and their glosses. In chapters 4 and 5, I address the second problem, exploring how these small collections of bitext can be enriched and analyzed, using a combination of existing NLP methods and a new algorithm for automatic morphological inference.

It is possible to view these different processing steps as stages in a larger system for creating rich corpora for RPLs. In the following section, I describe the structure of this overall system and how the different stages fit together with one another. However, in the dissertation each chapter addresses its subject more or less independently of the others. For evaluation purposes, I often use data sources from languages that do not qualify as RPLs; the reason for this is relatively self-evident: the only languages

for which I can obtain suitable data sets are by definition not RPLs!

1.2 Multi-Stage Processing of Electronic Linguistics Texts

The processing steps described in this dissertation can be thought of as components in an end-to-end system to collect and analyze linguistic data for resource-poor languages. At a high level, this system takes bilingual documents as input, and produces morphological analyses as output. Figure 1.1 illustrates the overall process and the principle processing stages, showing examples of the types of data that are involved at each stage. Below is a brief description of these stages.

Stage 1: Document Collection

OUTPUT	:	A document containing instances of bitext

The first step for any type of processing is to collect a data set. In chapter 3, I discuss how keyword searches can be used to identify relevant books in the University of Michigan Library's Digital General Collection. These books include grammars of foreign languages which have been scanned and have undergone an optical character recognition process, resulting in an electronic text that is primarily in English, but contains words and sentences from the foreign language as well.

Stage 2: Bitext Extraction

INPUT	:	A document containing instances of bitext
OUTPUT	:	A list of bitext sentences

The optical character recognition process which detects the letters on the page does not distinguish English text from foreign text. This stage of processing identifies the foreign words and phrases within the document, and also looks for neighboring

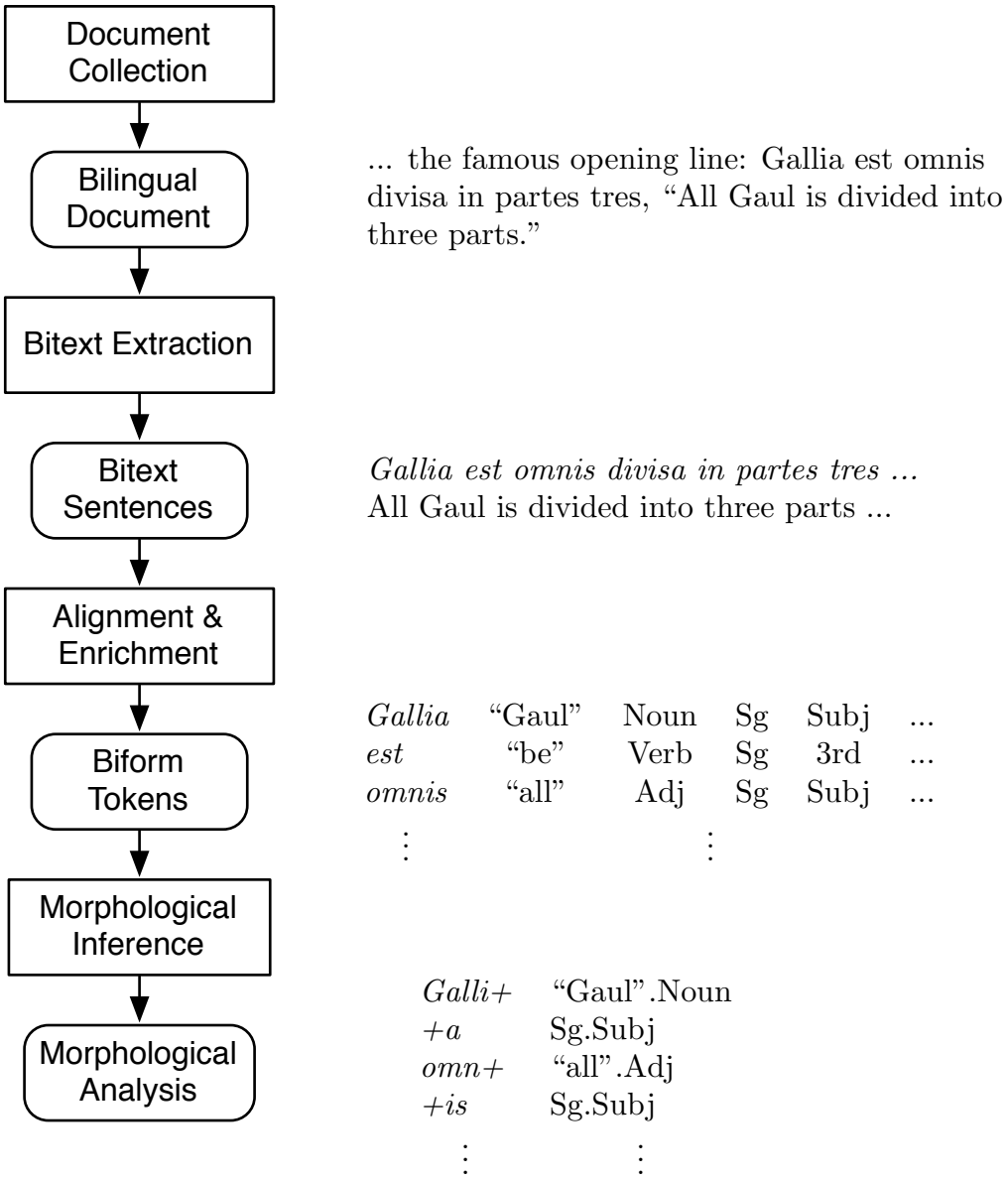


Figure 1.1: High-level system diagram.

English text that provides a gloss of the foreign text. The output of this processing is a collection of foreign sentences and their English translations. Chapter 3 discusses a statistical method for identifying these translation pairs (bitexts).

Stage 3: Alignment & Enrichment

INPUT	:	A list of bitext sentences
OUTPUT	:	A list of word-level biform tokens

In chapter 4, I discuss a method for enriching a bitext corpus by using existing NLP software. The English side of the bitext is analyzed with a syntactic parser and morphological analyzer, creating richly-annotated word tokens. Statistical word-alignment software, used to train machine translation systems, aligns English words with their foreign equivalents, and the morphosyntactic features are transferred to the foreign side of the bitext. The resulting output is a corpus of annotated *biforms*: tokens consisting of foreign wordforms accompanied by an English translation and morphosyntactic features.

Stage 4: Morphological Inference

INPUT	:	A list of word-level biform tokens
OUTPUT	:	A morphology of the language

Chapter 5 presents a novel algorithm I have designed for morphological inference using a corpus of annotated biforms as its training data. The morphological inference process segments each biform in the corpus into a prefix and a suffix, simultaneously producing a morpheme lexicon and a paradigmatic model of the morphology of the language.

This multi-stage model illustrates how automatic methods for language processing can be used to transform an existing resource with limited utility into a machine-

readable corpus of analyzed, richly-annotated, foreign text. In order for this to work as an end-to-end process, it will be necessary to improve the quality of data sources and limit the error rate of each individual stage of processing, something that will need to be addressed in future work. While this broader picture should be kept in mind, in the remainder of this dissertation I address each individual processing stage separately. I do not attempt to carry a single data set through the entire process, as this would result in an accumulation of noise and would not be suitable for assessing the performance of each individual component. Instead, each component is evaluated using whatever data sets were available to me that were most suitable for the given task.

In the following chapter I discuss related work and provide additional background and motivation for this research. The three chapters that follow each address a different component of the system described above, describing how it is implemented and evaluating its performance. The final chapter summarizes the results of this work and provides directions for continued research on this topic. Details on the software and data sets used in this dissertation can be found in the appendices.

CHAPTER 2

Motivation and Prior Work

2.1 Motivation and Goals

The terms natural language processing and computational linguistics are often used interchangeably, though the mere existence of the two different terms implies that there is a distinction to be made. As Abney (2011) points out, judging solely by its name, one would expect "computational linguistics" to denote a subfield of linguistics, just as historical linguistics, socio-linguistics, or psycho-linguistics do, or as computational biology or computational astrophysics denote subfields of biology and astrophysics respectively. Such an interpretation may not match up well with the common usage of the term—research published under the computational linguistics label is often more focused on engineering solutions than linguistic analysis—but I believe there is an important place for computational linguistics within linguistics. Data-driven, computational research methods have the potential to lead to important new discoveries relating to the fundamental linguistic question of how human language works.

Doing linguistic research in a computational manner requires data, ideally a lot of data, in an electronic format that is suitable for automated processing. While more and more linguistic data is being produced in or converted to digital formats every day, only a small portion of this data exists in machine-readable format. Chapter 3 of

this dissertation addresses the question of how human-readable electronic documents, such as the scanned pages of books describing foreign languages, can be converted to a more usable, machine-friendly format.

For this data to be useful for answering the sorts of questions a linguist would like to ask, it must be analyzed and annotated. While annotations produced by trained specialists are ideal, they are expensive to produce. Therefore it is necessary to look for automatic methods for generating analyses and annotations. Even if the automatically-generated analyses are less than perfect, it is usually easier to manually correct an imperfect annotation than to create a new one from scratch. Chapters 4 and 5 of this dissertation discuss my approach to using bitext and statistical word alignment to generate richly annotated corpora, and for performing automatic morphological analysis on the basis of annotated foreign wordforms.

In many ways, the work in this dissertation is unique, particularly in its use of digitally scanned grammars as source material and in using bitext as training material for morphology induction. However, this work is heavily indebted to a vast body of work on statistical methods for NLP, and it is inspired by other work in the emerging field of digital language documentation. In this chapter, I discuss some of this related work, which comes from a variety of fields with different objectives, in order to provide some context and background for the work discussed in the later chapters of this dissertation.

2.2 Resource-poor Languages

As previously stated, the goal of the research described in this dissertation is to enable computational linguistic research on resource-poor languages (RPLs). The term refers only to these languages' lack of *digital* resources; resource-poor languages are not necessarily endangered, under-studied, or minority languages (although they may be). In fact, there is little that RPLs have in common other than *not* being a

member of the "linguistic 1%": that handful of languages for which there do exist substantial quantities of digital data and language-processing software.

Before the internet age, digital text of any language was hard to come by. Classicists rightly claim to have been pioneers of the electronic corpus frontier; the *Thesaurus Linguae Graecae*[1] was launched in 1972, and at first it required its own hardware, the *Ibycus* computer created by David Packard, to run (Berkowitz and Squitier, 1986). (Today, the TLG is an online corpus containing virtually every word of Greek literature written prior to 1453.) However, modern statistical methods for NLP were made possible by large annotated corpora of English, such as the tagged version of the Brown Corpus (Francis and Kučera, 1979) released in 1979 (the original, untagged, version was released in 1964), and then the Penn Treebank in 1993 (Marcus et al., 1993). Early work on statistical machine translation used English and French as the languages of study because the 30-million-word Canadian Hansards was the only large bilingual corpus the researchers could obtain (Brown et al., 1988).

Thanks to the growth of the internet and interest in language technology, the number of digital corpora available today is comparatively enormous. However, it is enormous only when compared to the limited corpora of the past, not when compared to the number of languages that exist on Earth. There is no precise way to define which languages are "resource rich," but the number is probably around a few dozen. Abney and Bird (2010), citing Maxwell and Hughes (2006), put the figure at 20–30 languages. For comparison, the Ethnologue catalogs information for what it claims are "all of the worlds 6,909 known living languages" (Lewis, 2009).

Figure 2.1 illustrates the number of languages that are supported by a sample of modern tools and corpora. A treebank is a corpus of sentences accompanied by syntactic parse trees: because the parse trees must be either manually created or manually verified, treebanks (like other annotated corpora) are much less common

[1] `http://www.tlg.uci.edu/`

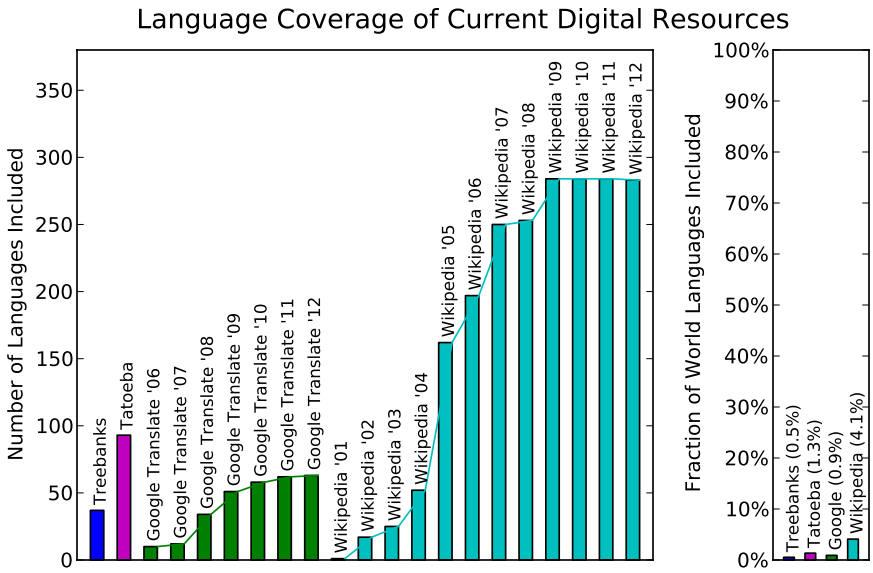


Figure 2.1: Availability of language resources. The main plot shows the number of languages for which treebanks, parallel corpora (represented by the Tatoeba database), MT systems (represented by Google Translate, over the past seven years), and monolingual corpora (represented by Wikipedia, over the past 12 years) exist. The sub-plot on the right shows these same numbers as a fraction of all the 6,900 languages listed in the Ethnologue.

than unannotated corpora. The number of treebanks given in fig 2.1 is an estimate based on information available online[2] and may overstate the truth, since most non-English treebanks do not achieve the scale of the Penn Treebank. Tatoeba[3] is an open, community-driven database of sentences with translations in two or more languages. The number of languages in Tatoeba represents something of an upper limit on the number of languages for which machine translation should be possible: high-quality MT systems require rather large parallel corpora, whereas Tatoeba includes a large number of languages but may only have a handful of sentence pairs for some of them. The Europarl Corpus[4] (Koehn, 2005) provides millions of words of parallel text for eleven European languages, and fairly large amounts of bitext can be found for a number of other European languages as well as Chinese and Arabic. The number of

[2] `http://en.wikipedia.org/w/index.php?title=Treebank&oldid=379424119`

[3] `http://www.tatoeba.org`

[4] `http://www.statmt.org/europarl/`

languages supported by Google Translate[5] has grown from 4 to 65 over the past few years, although the data they used to train their models is not necessarily available to outside researchers. Finally, looking at Wikipedia provides one way of estimating the number of languages for which monolingual text might be found; while the number of foreign-language Wikipedias[6] grew rapidly at first, it has leveled off at 284 languages, a number which might with some caution be construed as representing the current number of digitally-active language communities.

Even 280 languages only amounts to 4% of the 6,909 language estimate. And while monolingual corpora like Wikipedia are extremely valuable, they are not sufficient to support current methods for building NLP software tools. We can say with confidence that 99% of the languages of the world count as resource poor. For these languages, some resources may exist, but nowhere near the scale and availability of those that exist for resource-rich languages. In order to facilitate computational linguistic research on these languages, emphasis must be placed on exploiting all the resources that do exist and on coming up with new language processing techniques that are effective given limited data as input.

2.2.1 Digital Language Resources

It is not the case that resource-poor languages are neglected. Numerous projects exist to collect, organize, distribute, and preserve data pertaining to these languages, although only a few make data available in a format conducive to automated processing. SIL's Ethnologue[7] (Lewis, 2009) contains basic statistics, information, and references for 6,909 languages. The Open Language Archives Community[8] (OLAC) lists resources relating to 7,665 distinct languages, although for many languages those resources amount to basic statistics about the language, and not any actual data from

[5] http://translate.google.com
[6] http://en.wikipedia.org/w/index.php?title=List_of_Wikipedias&oldid=485086077
[7] http://www.ethnologue.com/
[8] http://www.language-archives.org/metrics/

the language. However, the primary goal of OLAC is to centralize information about various data archives, which are generally isolated from one another and do not necessarily make their data publicly available. Thus, while these databases are useful resources, they do not directly provide the type of primary language data that is necessary for computational linguistics.

Other projects, however, do contain this type of data. Kevin Scannell's *An Crúbadán*[9] project (Scannell, 2007) crawls the web to discover pages written in different languages, and has collected textual data from 473 languages to date. Unfortunately, the data he has collected is not publicly available to download, due to copyright restrictions. The Online Database of Interlinear Text[10] (ODIN) has collected 130k examples of interlinear glossed text (IGT) from 1,274 languages, thanks to the large number of scholarly publications that are available electronically online. Users of ODIN can find links to the online documents from which each instance of IGT was obtained, and a subset of the IGT instances are available to view or download directly through ODIN. The Rosetta Project[11] contains documents and recordings from over 2,500 languages, which are made publicly accessible through the Internet Archive.[12]

The Internet Archive is also home to the Text Archive[13] and Open Library,[14] a general collection of digitized versions of books in English and other languages. Similar resources include Project Gutenberg,[15] Google Books,[16] the Hathi Trust digital library,[17] and a number of other library digitization projects including the University of Michigan's own Digital General Collection.[18] Although they are not focused

[9] http://borel.slu.edu/crubadan/
[10] http://odin.linguistlist.org/
[11] http://rosettaproject.org/
[12] http://archive.org/
[13] http://www.archive.org/details/texts
[14] http://openlibrary.org
[15] http://gutenberg.org
[16] http://books.google.com
[17] http://www.hathitrust.org/
[18] http://quod.lib.umich.edu/g/genpub/

specifically on linguistics-oriented documents, these collections contain millions of books, some of which (for instance, grammars or dictionaries of specific languages) contain valuable linguistic data. Chapter 3 discusses in detail how the data from these scanned books can be extracted for machine processing.

Additional examples of projects aimed at growing the amount of digital language resources are not too hard to find. Spoken Language Technologies for Under-resourced languages[19] (SLTU) is a workshop for speech technology that focused on Asian languages in its first year and on African languages in its second year (2010). The EMILLE project[20] (Enabling Minority Language Engineering) extended the GATE text engineering architecture to be Unicode-compliant and used it to create a 97-million-word corpus of South Asian languages. More recently, Microsoft announced its Translator Hub,[21] a cloud-based platform where users can upload data for their own language, then use Microsoft's back-end to train a machine translation system. According to their website, Translator Hub "[helps] smaller languages thrive by putting the power to build machine translation systems in the hands of local communities."

Other projects provide language data, but not necessarily machine-readable data that is most useful for NLP and CL. The Archive of Indigenous Languages of Latin America (AILLA)[22] includes field recordings with transcriptions and translations, but no annotation of the foreign text. The data being collected by the Basic Oral Language Documentation project[23] includes recordings of indigenous languages of Papua New Guinea; a subset of the recordings for each language will be transcribed and translated as part of the documentation project. These are extremely valuable sources of language documentation data, but they can only be maximally utilized if they are made available in an machine-friendly interface and format.

[19] `http://www.mica.edu.vn/sltu/`
[20] `http://www.emille.lancs.ac.uk/`
[21] `http://hub.microsofttranslator.com/`
[22] `http://www.ailla.utexas.org/`
[23] `http://www.boldpng.info/`

2.2.2 Cross-lingual, Data-Driven Research

While collecting and organizing digital data from all of these languages is itself a significant task, the main motivation behind creating these digital resources is to enable research. While any given corpus is useful for performing research on one particular language, a large-scale, multi-lingual corpus is useful for performing research on whole language families and language as a whole. One of the stated goals of the RiPLes project[24] (Information Engineering and Synthesis for Resource-poor Languages, of which ODIN is a sub-project) is "to perform cross-lingual study on a large number of languages to discover linguistic knowledge". This seems to parallel the sentiments of Abney and Bird (2010) and their vision of a "a new science of empirical universal linguistics" based on a universal corpus of linguistics meant to include data from all of the world's languages.

Not waiting for a universal corpus, Daumé III and Campbell (2007) performed a simpler type of cross-lingual investigation using the WALS[25] (World Atlas of Linguistic Structures) database. The authors used the database, which includes 141 typological features for 2,650 languages (although not every feature is represented for every language), to perform automatic inference of the interrelationships between typological features (things like, e.g., "if a language has Object-Verb word order, then it also has postpositions"). Their induced list of typological implications largely matched those previously identified by linguists. In a another study, Iwata et al. (2010) used a non-parallel, multilingual text corpus to automatically infer common syntactic structures across languages; this work has obvious implications for research on the presence of universal grammar. These projects have not yet produced any major new discoveries, but the inferences drawn by these automated approaches are valid (they match traditional linguistic analyses), and this line of work shows the in-

[24] `http://faculty.washington.edu/fxia/riples/`
[25] `http://wals.info/`

terest and the potential for cross-lingual statistical methods as an avenue for research on linguistics.

From an engineering standpoint, multilingual resources (as opposed to monolingual resources) can also improve performance of NLP systems. For example, Berg-Kirkpatrick and Klein (2010) demonstrated that learning dependency parsers based on multilingual (but not parallel) training data can improve performance over monolingual data. The authors' model assumed a prior distribution over the parameters (in the probabilistic sense, not the Chomskyan sense) of grammars for eight languages (English, Dutch, Danish, Swedish, Spanish, Portuguese, Slovene, and Chinese). When training the grammar, the parameters of closely-related languages (as determined by a pre-defined phyolgeny) are coupled with one another. By treating all the languages as if they belonged to a single family (ignoring the fact that in reality these languages range from closely-related to completely unrelated), the average performance of the parsers for all languages improved by 10% over parsers trained on monolingual data. By additionally incorporating phylogenetic information about the family relationships between languages (e.g. English is closely related to Dutch, but is not related to Chinese), the performance improved 21%. That is, by using more data from more languages, it is possible to improve the performance of a parser for any single given language.

These two studies are just a sample of some of the potential benefits that could be gained, both in scientific research on the nature of language and in the development of NLP technology, as a result of increased electronic resources from a broad base of diverse languages. In a sign that the linguistics community recognizes the usefulness of freely available data, at its 2010 Business Meeting the Linguistic Society of America passed its *Resolution on Cyberinfrastructure*[26] which encourages making linguistic data (full, annotated data sets) public for research (to the fullest extent possible given

[26] `http://lsadc.org/info/lsa-res-cyberinfrastructure.cfm`

the nature of the data). I consider this type of data accessibility a key component to the future of linguistics research and through my dissertation research I hope to improve both the quality of data that is available for research and the quality of methods used to analyze such data.

2.3 Related Work in NLP and CL

The processing described in chapters 3 through 5 employs a combination of existing NLP tools and new techniques designed specifically for this dissertation. The following sections summarize prior work that is related, either directly or indirectly, to the present work.

2.3.1 Bitext Discovery

Chapter 3 of the dissertation addresses the problem of extracting bitext from bilingual documents. This is a novel task, particularly in terms of the granularity of the data: the task in this dissertation is to classify individual words or sentences within a document, whereas prior work almost exclusively works on classifying whole, monolingual documents. While there is no directly-comparable prior work, there are some similar efforts that deserve mention.

ODIN is a similar project in that it detects instances of IGT within a text. However, IGT is easily identified by its format (three sequential lines of tabular text, often containing characteristic abbreviations in the middle line), and the creators of ODIN exploit this by using templates and other formatting features to automatically identify instances of IGT (Lewis and Xia, 2010). Bitext, as opposed to IGT, may appear embedded within a paragraph, with few formatting cues to distinguish it from the surrounding text. Therefore the approach discussed in chapter 3 is based on text content rather than formatting, identifying foreign-language text and possible translations.

ODIN researchers also address the problem of identifying the language of instances of IGT within a document (Xia et al., 2009); their approach combines contextual features within the document (such as the name of a language mentioned in the title or near the IGT) with character and morph ngram features. Xia et al. (2009) also point out the unseen language problem; standard approaches to language ID fail when the target language is outside of the set of languages used to train the classifier. Statistical learning approaches to language ID can achieve high classification accuracy even on documents that are relatively small, for instance hundreds of words (Kruengkrai et al., 2006), but they require a training set containing text from each language. No prior work to my knowledge addresses the question of identifying the language of individual words within a multilingual document.

Previous efforts to automatically discover parallel text on the web, summarized well by Resnik and Smith (2003), are conceptually relevant. One approach works by identifying documents from the same domain that closely resemble one another in terms of HTML structure, ignoring the textual content of the pages. Another approach uses a bilingual lexicon to look for documents from the same site that have a similar textual content. These two approaches may be used separately or combined. At a high level, the idea of looking for parallelism in structure and content apply to the work described in chapter 3 of this dissertation. However, due to major differences in the data (bilingual sentences within unstructured text documents, compared to monolingual documents in structured HTML format) and different assumptions about the existence of additional resources (I assume no bilingual or monolingual sources exist other than the document being classified), the actual algorithms and procedures described in this dissertation are unrelated to those used in prior web-as-corpus work.

2.3.2 Structural Analysis using Bitext

The feature transfer process described in chapter 4 of this dissertation is based on the "align and transfer" methodology initiated by Yarowsky et al. (2001). In the align and transfer methodology, parallel bilingual text (bitext) is used, with the assumption that one of the two languages involved is a reference language for which language processing tools exist (for example, English). The English text can be parsed and analyzed using existing tools, then the English words aligned with the foreign words using standard alignment tools from machine translation like GIZA++ (Och and Ney, 2003). Then, the linguistic features from the English side (e.g. part-of-speech tags, grammatical roles, phrase structure) are transferred via the alignment to the foreign words. This effectively produces labeled data which can be used to train language processing tools for the foreign language.

In their work, Yarowsky et al. (2001) used the align and transfer methodology to create a number of NLP tools for non-English languages, including noun-phrase-bracketers, part-of-speech taggers, named-entity taggers, and lemmatizers for French, Chinese, Czech, and Spanish. The performance of the French stemmer was extremely high, but the Czech performance was significantly lower; the authors cited poor alignment between English and Czech (which is a highly-inflecting language) as one source of error. This approach has also been shown to work for lemmatizing verbs, nouns, and adjectives in German, using a combination of English glosses from aligned bitext and edit distance measured between German word forms (Moon and Erk, 2008). One major difference between this work and the present work is the scale of the corpus: Yarowsky et al. (2001) used corpora of roughly 2 million words per language, whereas the corpora used in this dissertation are on the scale of tens of thousands of words.

There is a chicken-and-egg problem in which word alignments with English text can help train morphological analyzers for foreign languages, but at the same time morphological segmentation of the foreign language can help improve word align-

ments. This circularity can be side-stepped if segmentation and alignment are both jointly learned in a single process. This is the approach taken by Snyder and Barzilay (2008), who use sampling techniques to estimate the parameters of a Bayesian model that generates parallel text by drawing from a set of "abstract morphemes", i.e. pairs of morphemes in more than one language that accomplish the same or similar function. In their work, Snyder and Barzilay discovered that learning bilingual morphological models in this way improved performance significantly over monolingual models. Furthermore, their approach was effective even for unrelated language pairs (e.g. English and Aramaic) although performance was better for related language pairs (in their study, Heberew, Arabic, and Aramaic). Another interesting component of their approach was their choice of data: starting with a statistical word-level (actually, phrase-level) alignment, they selected pairs of recurring short phrases that were always translated the same way, and which occurred more than four times in the corpus. It seems likely that their careful selection of training data played a role in the effectiveness of their learning algorithm, and that is something to be emulated.

In related work, ODIN researchers (Xia and Lewis, 2007; Lewis and Xia, 2009) used an align-and-transfer methodology to enrich IGT by transferring parse information from the English translation to the foreign text. Rather than using statistical alignments, the interlinear layer itself provides a form of alignment. The enriched IGT was then used to automatically discover typological features, like canonical word order, of the languages in the database.

2.3.3 Morphological Inference

In chapter 5 of this dissertation, I investigate computational methods for morphology induction, the automated learning of the morphological system of a language. Morphological analysis has a rich history in computational linguistics research, as morphology is an important component of the analysis of many languages (the ex-

ceptions being isolating languages which exhibit little or no morphology). However, current approaches to automated morphological analysis are not well suited to applications involving languages with limited resources. Furthermore, the goals of previous work on morphological analysis do not necessarily align with the goals of digital language documentation. Therefore, I address the question of how existing approaches can be improved upon, making maximal use of available data, in order to facilitate the rapid development of language processing tools for under-resourced languages.

There are two distinct ways in which automated morphological analysis can be achieved. In the rule-based approach, morphological rules are manually composed for a specific language, and these rules are applied to analyze textual input. In the learning approach, data is analyzed by a learning algorithm to automatically produce a grammar and an analyzer.

Systems based on language-specific rules or morphological descriptions can be very effective and accurate, although they can require some time to develop, particularly for morphologically complex languages. For example, the TreeTagger system[27] consists of a generic algorithm for tagging text combined with a language-specific parameter file specifying the part-of-speech tags and corresponding forms (Schmid, 1994). The TreeTagger website includes user-contributed, downloadable morphological grammars for thirteen different languages: Bulgarian, Chinese, Dutch, English, French, Old French, German, Greek, Italian, Portuguese and Galician, Russian, Spanish, and Swahili. Similarly, PC-KIMMO[28] provides a morphological parser, and users can specify their own language-specific lexicon and rules. Likewise, XFST and its open-source counterpart, HFST[29] (Helsinki Finite-State Transducer Technology), allow users to write rule-based morphologies and phonologies and compile them into efficient finite-state automata.

[27] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`
[28] `http://www.sil.org/pckimmo/`
[29] `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/`

If the morphology of a language is known, and the user is willing to spend the time specifying the morphological rules in the format required by the program (which requires not only adopting standard conventions for inputting the rules, but possibly also theoretical requirements for the formulation of the rules), then these programs can be very useful. Specifying a list of morphological rules is much easier than writing an entire morphological analyzer, and these programs have been shown to achieve high accuracy.

However, in some cases it may not be possible, practical, or desirable to use one of these programs. If no expert in the language is available (or if the grammar of the language is simply unknown), or if time and resources are limited, then it may not be possible to create the necessary rule specifications that these programs require. If other resources in the language exist, such as texts that have been scanned or manually entered into an electronic format, then the learning approach can be used: rather than manually specifying the morphological rules for a language, these rules are learned automatically by a machine learning algorithm.

Machine learning algorithms are often classified by the degree of "supervision" that they require. A supervised morphological analyzer would require a certain amount of labeled data to train on: for example, a corpus in which inflected forms have been manually broken into morphemes, and perhaps each morpheme has been labeled with a grammatical tag. IGT is an example of this type of labeled data. Inflected forms that have been organized into paradigms can also be viewed as labeled data suitable as input for supervised learning algorithms. Unsupervised algorithms, on the other hand, require no labeled data as input. Instead, they analyze unannotated text to discover patterns that exist. Semi-supervised methods are designed to take advantage of a combination of labeled and unlabeled data.

Unsupervised morphological induction can be viewed as a grammatical inference problem, with implications for learnability theory as well as practical applications for

language processing. It has been shown that infants can learn to segment an acoustic speech stream into words (sequences of syllables) based purely on statistical patterns in the data (i.e. word-internal vs. word-boundary syllable transition frequencies); no additional acoustic cues or environmental cues (i.e. word-meaning) are required (Saffran et al., 1996). This parallels the unsupervised learning scenario, where only plain text, with no indication of the true morpheme boundaries or word meanings (or even parts of speech) is available. Likewise, machine learning algorithms can use the inherent structural patterns in language data to build theories about how the language works. Although focused more on syntactic structure than morphology, the Ph.D. theses of Carl de Marcken (de Marcken, 1996) and Dan Klein (Klein, 2005) provide good descriptions of the unsupervised grammar induction problem and algorithmic approaches to solving it.

The state of the art for unsupervised morphological inference is represented by two publicly available software implementations. *Linguistica*[30] (Goldsmith, 2001) uses a minimum description length framework to discover stems and signatures. A signature is the set of affixes that co-occur with a given stem: for example, in a given corpus the English stem *walk* might have the signature (*–s*, *–ed*, *–ing*, *–er*). Goldsmith's algorithm segments a text by essentially finding the most compact encoding of the corpus; a dictionary of stems and affixes is constructed, and each token in the corpus is encoded with pointers to its stem and affixes in the dictionary. This approach is effective, and Goldsmith claims that as little as 5,000 words of text from English is sufficient to produce an acceptable analysis. *Morfessor*, maintained by the Morpho project[31] at the Helsinki University of Technology, is another software package for unsupervised morphological analysis (Creutz and Lagus, 2005). Also based on the minimum-description-length principle, current versions of the Morfessor software incorporate improvements on the heuristic search that drives the learning process.

[30] `http://linguistica.uchicago.edu/`
[31] `http://www.cis.hut.fi/projects/morpho/`

Morfessor is used as the baseline for evaluation of entries to the Morpho Challenge,[32] a contest for unsupervised morphology induction systems held annually since 2005.

Although they are both interesting (showing to what extent language learning is possible from distributional information alone) and practical (requiring only monolingual text), purely unsupervised approaches to morphological analysis are unlikely to be satisfactory for linguistic analysis. Even if the segmentation itself were perfect (which is unlikely), other tasks such as glossing morphemes, disambiguating inflected forms, or identifying allomorphs, are very difficult if not impossible in an unsupervised, monolingual context. As a practical matter, improved performance can often be obtained by semi-supervised learning, in which a relatively small amount of labeled data is combined with unlabeled data. Using semi-supervised methods, one can avoid the high costs associated with fully-supervised learning (i.e. the costs of manually annotating a large corpus of data), while improving on the performance of fully unsupervised learning. Much work exists in morphological analysis that fit into the semi-supervised category (although variously referred to as "minimally supervised" or "partially supervised"), and these differ both in the type of data they incorporate and the manner in which that data is used.

Morphological analysis is rarely an end in itself. Stemming and part-of-speech tagging, two common examples of shallow morphological analysis, may be used as components in larger language processing tasks, such as machine translation or syntactic parsing. Also, when building a corpus of linguistic data for further analysis (either automatic or manual), labeling the morphemes of an inflected form can make it much easier to later find relevant data items to support a certain analysis. In the following sections, I discuss the applications of morphology in machine translation and in language documentation.

[32] `http://www.cis.hut.fi/morphochallenge2010/`

2.3.4 Morphology in Machine Translation

A significant drawback of traditional statistical machine translation systems is that they are essentially lexical: fully-inflected words or phrases are translated into other fully-inflected words or phrases. Machine translation into a morphologically rich language from a morphologically poor language is generally harder than vice-versa, in part due to the uncertainty introduced by pronoun coreference when generating morphological agreement (Koehn, 2005; Minkov et al., 2007). Performance can be improved by using syntactic features (e.g. word-order) from the morphologically poor source language to generate morphological features for the rich target language (Minkov et al., 2007). However, this required pre-existing morphological analyzers for the languages (Russian and Arabic) involved.

Factored translation models (Koehn and Hoang, 2007) improve upon traditional word-based or phrase-based translation by incorporating lexical factors, such as part of speech tags or morphological analysis of word tokens. This improves translation performance in part by reducing the data sparseness problem (since multiple inflected forms of a single lemma can be pooled together). However, the factorization of the source text must be performed before translation, and this requires a morphological analyzer. If an effective analyzer could be induced, this would avoid the necessity of hand-building a morphological analyzer or hand-tagging data to train a statistical morphological analyzer.

However, in the general case of a resource-poor language, a pre-existing morphological analyzer is not likely to exist. Therefore it is important to have some way of inducing a morphological analysis for the language. Sharif-Razavian and Vogel (2010) used fixed-length suffixes, rather than inducing a full morphology, as features in factored translation systems for three scenarios: a "small" (650,000 sentence pairs in the training set) English to Iraqi Arabic system; a "medium" (1.2 million sentence pairs) Spanish to English system; and a "large" (3.8 million sentence pairs) Arabic

to English system. In all cases, the addition of fixed-length suffix factors improved the translation quality. Sereewattana (2003), used a character n-gram frequency-based method for unsupervised morphological segmentation for French-English and German-English translation, which resulted in an improvement in the the word error rate of translations over non-segmented translation. The training corpus involved in this project was 40,000 sentences, which may suggest that morphological analysis is most useful when training data is relatively limited. Virpioja et al. (2007) used the Morfessor analyzer to perform unsupervised segmentation for statistical machine translation between Finnish, Danish, and Swedish. Their morph-based system failed to achieve higher BLEU scores (the standard measurement for machine translation evaluation) than the word-based system, although it greatly reduced the number of sentences that could not be fully translated due to out-of-vocabulary problems.

According to Virpioja et al., most other morphology-informed machine translation aside from their work has looked at how to translate from a morphologically complex language into English, a morphologically simple language. (They cite papers using this paradigm to translate from German, Arabic, Czech, Finnish, Spanish, Catalan, and Serbian.) A notable exception is the ongoing work of Kemal Oflazer looking at translation from English into Turkish (Durgar-El-Kahlout and Oflazer, 2006; Yeniterzi and Oflazer, 2010). It is important to emphasize that all of the work on morphology-based MT (aside from the work mentioned in the previous paragraph) involves using morphological analyzers that are specifically tailored to the languages involved: either the morphological analyzer must be hand-built, or it must be trained on data that has been manually annotated. Either one of these processes is time-consuming and not extensible to languages that lack the appropriate annotated data.

2.3.5 Morphology and Language Documentation

A very different application of morphology induction is language documentation. When describing a language that has never been studied before, determining the morphemes of the language is one of the first tasks a linguist must undertake, along with phonemic analysis and collecting a lexicon of native vocabulary. The use of software tools to organize and analyze data can certainly help a linguist with the painstaking processes of documentation and description of the language, but these tools must also meet the unique challenges of documentary data, which differs in significant ways from the types of data sets typically used in machine learning experiments. Bird (2009) points out the problem that, since the analysis is in a state of flux, the data itself may change over time; for example a linguist may revise his or her transcription system, or may re-analyze data that was previously analyzed. Furthermore, Bird observes that language processing tools for language documentation must meet the potentially opposed challenges of *upscaling*—that is, designing algorithms that work on a broad range of typologically varied languages—and *downscaling*—designing algorithms that can function with relatively small amounts of data. Previous work on morphology induction, which often assumes that a language is exclusively prefixing or exclusively suffixing, and ignores non-affixal morphology almost entirely, fails to meet the challenge of upscaling. The problem of quantity of data is also not often addressed in previous work; most results are reported for data sets of tens of thousand words — relatively little data by machine learning standards but still a sizable amount to be collected in the field.

At the same time, the goals and assumptions of a morphological analyzer for linguistic fieldwork differ from those generally assumed in morphological induction experiments. Palmer et al. (2010) investigate how active learning can be used to speed up the process of manual annotation of linguistic data. The authors design a system that uses previously annotated examples to identify challenging unannotated

Affix Type	Number
Little affixation	141
Strongly suffixing	**406**
Weakly suffixing	124
Equal prefixing and suffixing	147
Weakly prefixing	94
Strong prefixing	59

Table 2.1: Number of languages with various prefixing vs. suffixing preferences for inflectional morphology. Of the 971 languages considered, 530 (55%) are strongly or weakly suffixing, compared to just 153 (16%) that are strongly or weakly prefixing. From the World Atlas of Language Structures (Dryer, 2011).

examples to be manually annotated by a linguist. Their system helped increase the rate at which two linguists (one expert and one non-expert) selected the appropriate gloss tag for morphemes in an utterance from Uspanteko, a Mayan language. In this view, the role of machine learning is not to replace human annotators, but to maximize efficiency by handling the simple tasks and identifying the complex tasks for human intervention. The annotation software they created for this project is available as the OpenNLP IGT Editor[33].

Hammarström (2009) takes a fully unsupervised approach to morphology induction for language documentation. From a smallish corpus of 60,000 words of text from Mpielo, a Bantu language with about 29,000 speakers (Lewis, 2009), his morphology induction approach is successful at identifying the morphemes and categorizing stems based on the affixes they take. He observes although precision is high, recall is low; many stems do not appear frequently enough in the corpus to be properly categorized. Also, without access to semantics, his approach cannot identify allomorphs. He briefly describes an attempt to cluster related words using latent semantic analysis, but based on the poor results concludes that the corpus is too small for this type of analysis. His work points to two areas for improvement: generalizing about

[33] `http://igt.sourceforge.net/`

sparse data (certainly there is a lower limit on how much text is necessary, but we would like to minimize this as much as possible); and adding semantic information (for example, using bitext). Hammarström rightly takes the view, like Palmer et al., that automated approaches like this are in no state to replace human analyses, but their usefulness lies in reducing the amount of time and effort required to perform the analysis.

It could be argued that one rarely encounters large amounts of completely unanalyzed data from an entirely unknown language. However, one can imagine scenarios in which data which has previously been collected and somehow made available (either in print or electronically), yet it is not possible to find an expert in that language to help annotate the data. For example, given unannotated texts with translations from grammars, field notes, or websites, it may be useful to perform an automated analysis of their linguistic structure in order to identify phenomena of interest for future, detailed linguistic study.

For example, the Perseus Project[34] includes classical texts in Ancient Greek and Latin, and also provides tools, including a morphological analyzer and Greek-English and Latin-English dictionaries, to assist language learners in reading the texts. Their morphological analyzer is very useful for what it does; it list all possible analyses of the wordform, ranked according to the estimated probability of that analysis in the given context; the estimate is based on unigram and bigram estimates of the likelihoods of various morphological features (e.g. first-person, plural, accusative case) in the whole corpus, as well as votes cast by other readers about which form is correct. Still, the disambiguation in context is not always correct, and votes are not available for most words in the corpus. However, English translations exist on Perseus for many of these texts, and it may be possible that they could be used to disambiguate ambiguous morphological parses.

[34] `http://www.perseus.tufts.edu/`

This would in turn enable both learners and researchers to more efficiently search the corpus for specific morphological forms and grammatical constructions. This same approach could be used to enrich existing texts in any number of languages, including under-resourced languages with even smaller text collections. For example, among the many resources available in the Archive of Indigenous Languages of Latin America (AILLA)[35] are field recordings with transcriptions and translations, but no annotation of the foreign text. The data being collected by the Basic Oral Language Documentation project[36] includes recordings of indigenous languages of Papua New Guinea; a subset of the recordings for each language will be transcribed and translated as part of the documentation project. In all of these cases, assuming that the transcriptions and translations are available in a suitable electronic format, then morphological analysis could increase their usability to researchers and language learners. As more and more older publications are being scanned into electronic formats, while newly collected data is more likely to exist primarily in electronic form, the potential for broad-scale, automated linguistic analysis is becoming ever more promising.

2.4 Summary

The work discussed in this chapter frames this dissertation within the emerging field of digital language documentation, whose ultimate goal is to create digital resources for all of the worlds languages in order to facilitate large-scale, cross-lingual research. Each of the processing stages discussed in chapters 2 through 5 of this dissertation extends on previous work in important ways designed to make them effective when applied to RPLs; the aim of this work is to increase the number of languages for which digital resources are available.

The bitext extraction task discussed in chapter 3 is a novel task, although it is

[35] `http://www.ailla.utexas.org/`
[36] `http://www.boldpng.info/`

conceptually similar to prior work on language identification and automatic detection of parallel text on the web (Resnik and Smith, 2003), and it utilizes standard approaches to word alignment of bitext (Och and Ney, 2003). The bitext extraction task is unique in that it performs language identification for individual words and short utterances (as opposed to larger texts), and in its innovative application of translation models trained on noisy data to identify translations.

The method for enriching bitext described in chapter 4 is very much based on previous work within the align-and-transfer paradigm (Yarowsky et al., 2001), and it makes use of a number of off-the-shelf NLP software tools (Karttunen, 1983; Klein and Manning, 2003b; Och and Ney, 2003). However, the morphological inference algorithm described in chapter 5, which uses the enriched bitext as its input, is completely new, extending previous work on unsupervised morphological inference (Goldsmith, 2001; Creutz and Lagus, 2005) to perform inference using labeled wordforms derived from bitext. The algorithm's performance is demonstrated on datasets that are much smaller than previously used in bitext-based studies, showing its applicability for use on RPLs for which limited data is available.

With the goals and background of this project established, the following three chapters present in detail the major stages of collecting, enriching, and analyzing bitext for resource-poor languages.

CHAPTER 3

Extracting Bitext from Electronic Documents

3.1 Introduction

As language users and language researchers alike transition from the print era to the digital era, the opportunities for finding linguistic data in digital formats are continually increasing. New digital content, in a multitude of languages, is written and published online every day, and large amounts of written text, once available only in print, are being scanned into digital formats and distributed on the web. Linguists today use a range of software programs to organize, analyze, and publish newly-collected linguistic data. In this chapter, I discuss an approach to digital linguistic data collection which targets this last type of digital language data: scanned versions of books describing foreign languages, such as grammars and lexicons. From a grammar, written in English, describing a foreign language, it is possible to extract a small corpus of words and utterances from that foreign language, accompanied by English translations. Figure 3.1 illustrates the high-level objective of the data extraction process. It begins with an online electronic document, and ends with a corpus of bitexts—foreign sentences with English translations.

While there is currently no shortage of digital texts to mine for data, the challenges associated with extracting high-quality, machine-readable data are as numerous as the texts themselves. The data is spread out across the web, and not all of it is easily

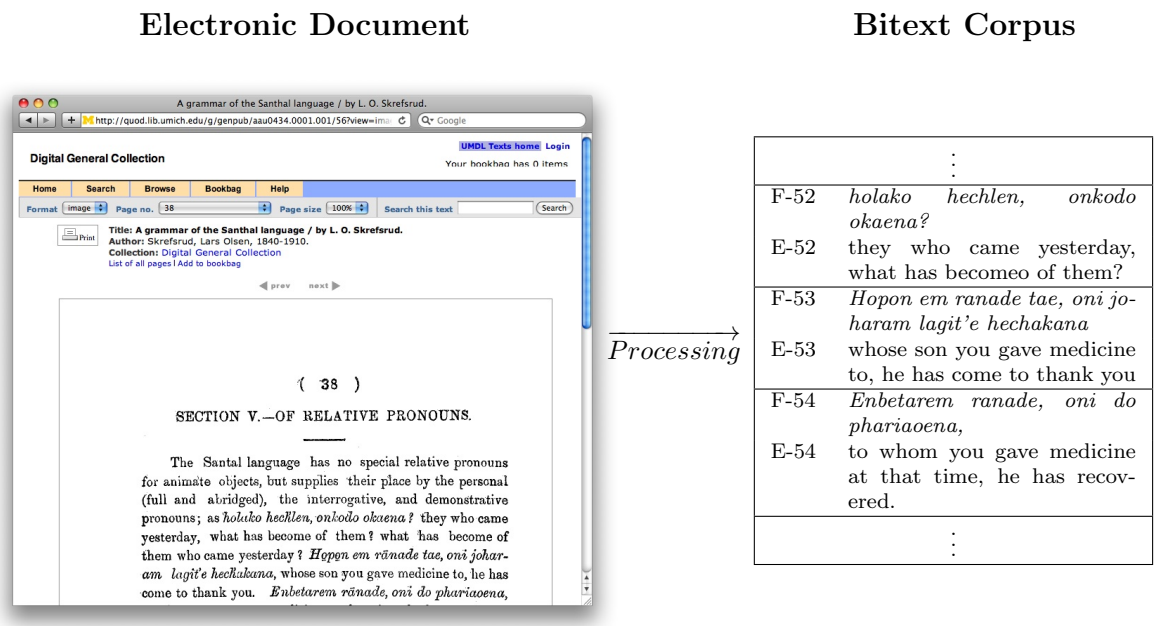


Figure 3.1: The high-level objective of bitext data collection.

accessible for researchers. Metadata, such as an identification of the language in which the text is written, is often limited or lacking entirely. While some sources, such as HTML web pages, may have been composed directly as digital text, in other cases the data exists only as scanned images of print materials. While many scanned documents are accompanied by text produced via optical character recognition software, this text is often plagued with errors, particularly in the case of non-English text. Even when the data is in a reasonable format, the quantity of data available for a given language is generally far less than is required for many traditional NLP methods to perform well.

Acknowledging these challenges, in this chapter I describe an approach to extracting linguistic data from scanned digital documents. This approach begins with a collection of digital books that have been scanned into the University of Michigan's Digital General Collection and partially annotated by hand, as described in section 3.3. The OCR text of these books is analyzed to detect foreign words within the majority English text, using techniques for language identification discussed in

section 3.4. Then, a probabilistic translation model is used to identify nearby English text that constitutes a translation of the foreign text: this process is described in section 3.5, and its performance is evaluated in section 3.5.2. The result of this process is a collection of English-foreign bitexts, which may be used as a stand-alone corpus or passed on as input for additional language processing, such as the bitext enrichment and morphology induction described in chapters 4 and 5.

The work discussed in this chapter was partially funded by a Digital Humanities grant from Google; these grants aim to promote humanities research that utilizes electronic text collections like Google Books. The project was a collaboration involving myself, Steven Abney, Ben King, and a small group undergraduate annotators. Specifically, the corpus-building and annotation process discussed in section 3.3 was a group effort in nearly every aspect; the language classification experiments discussed in section 3.4 were mainly performed by Steven Abney; and the gloss detection process covered in section 3.5 is my own work.

3.2 Rich Linguistic Bitext

Before getting into the details of the project, I want to first consider the goals of the project and the types of textual data that we aim to exploit. Bitext is a commonly used NLP resource, but linguistic documents like the ones described in this chapter contain a very different variety of bitext. Traditional bitext corpora, such as those used to train machine translation systems, consist of large quantities (millions of words) of text, typically drawn from one or two genres, such as news articles, government documents, or works of fiction. A descriptive grammar of a language, on the other hand, typically contains a much smaller quantity of bitext, but that bitext often contains much more information than a standard bitext. The usual text+translation format may be accompanied by additional data (e.g. linguistic terms or word-by-word glosses), and the utterances included in a grammar tend to

be chosen to represent the full range of expression of a language, whereas a generic corpus will tend to contain only a subset of lexical and grammatical elements that are typical for that genre. In this way, corpora drawn from linguistic sources can at least partially offset some of the problems associated with small sample sizes, because they are not random samples but artificially constructed samples designed to capture a lot of information.

In this section, I briefly mention several different styles of bitext that occur in linguistic documents, and give an example of each style. The examples are illustrative, but not representative of all the possible forms that bitext can take; while the general styles discussed below can be found recurring in nearly any grammar, every author presents the data in her own unique way. These examples show that within a given document the presentation of a bitext follows a fairly rigid pattern, with typography, punctuation, and layout used consistently to demarcate the foreign text, the gloss, and other components of the bitext. However, these regularities do not hold across different documents by different authors, and it is important that the extraction procedure is adaptive and not simply based on pattern-matching.

Interlinear Glossed Text

Interlinear glossed text (IGT) is probably the most common way in which linguistic data is presented in modern linguistics publications, although historically it is less common, which means it is less frequent among the older, out-of-copyright, books in our collection. IGT's defining feature is the interlinear gloss line, which provides detailed glosses of individual words in the foreign text and serves as an intermediary between the foreign text and the free English translation. The way in which this information is displayed on the page, however, can vary drastically across different sources; Bow et al. (2003) give an informative catalog of what they call the different *presentational formats* of IGT. Figure 3.2 gives an early example of interlinear glossed

text from a grammar of Japanese: although it is not in the standard three-line format, it still contains the basic elements of a foreign sentence, a free English translation, and a more detailed word-by-word gloss.

Wordlists

Not every grammar includes a wordlist, but many do, and dictionaries are essentially books consisting of nothing but one large wordlist. One challenge for extracting text from wordlists is dealing with multi-column formatting; depending on the OCR software used, columnar layouts can result in unpredictable output. Figure 3.3 illustrates a typical example of a wordlist.

Tables and Paradigms

Another source of rich bitext is paradigms, in which several morphologically-related wordforms are presented as a list or in a table. Figure 3.4 gives an example of a paradigm, in which various forms of a verb are listed in a single column, accompanied by glosses and linguistic term labels. Figure 3.5 shows an example of a paradigm in a complex tabular format. Key to properly recognizing these structures is proper handling of the tabular format; if the table cells are not treated as such then the text from different cells can get blended together and become unusable for extraction.

Facing Page Bitext

Facing-page bitext is a relatively common format for bilingual texts; alternating pages of text and translation are displayed in such a way that the reader can freely glance across from one to the other. Because each individual page is monolingual, this format significantly simplifies the problem of language ID. If facing-page bitext and inline bitext are both available for the same language (some documents combine both), then an effective approach may be to train a language model on the facing-

Yuki ga ii kagen ni
snow good condition
yameba, yoroshi ga,
if-stop is-good but
If it ceases snowing in reasonable time, it would be a good thing.

Warui koto sureba, warui
bad thing if-do
mukui ga aru,
reward is
If you do evil, there is an evil reward.

Areba, yō gozaimasŭ ga,
if-there-be good is
If there were some I should be glad.

Dekitara(ba), motte kite
if-has-forthcome carrying coming
kudasai,
condescend
If it is ready, please bring it with you.

Figure 3.2: Example of interlinear glossed text, in a non-standard presentation format (from H. Weintz *Hossfeld's Japanese grammar*, 1904).

YAO–ENGLISH VOCABULARY.

A

A- (1), *pers. pron. connect.*, he she, it.
-A, *prep.*, of, for : *-a cheni ?* whose? *-a chi ?* what kind of ? *-a chichi ?* what for ?
ACHA, ACHI, *plur. prefix. Class* 2, see *M-*, *Mw-*, *Mu-* or the stem to which these prefixes are attached.
ACHAMBA (2), women.
ACHIKULU (2), mother (*always used with the poss. pron. : e.g.* achikuluwangu, achikulugwe, etc.).
ACHIMBUMBA (2), women.
ACHIMSYENE (2), themselves.
ACHIMWENE (2), an honorific used among natives, but familiar. Used also in reference to an elder brother.
AKAŴE, *conj.*, but, except (*more emphatic than* NAMBO).
ALAMU (2), contemporary relatives by marriage (*vide Appendix II*).
ALUMBU (2), brother, sister.
AMAO (2), mother, maternal aunt (*vide Appendix II*).
AMBI, therefore : *Ambi uli ?* Well, what about it ?
AMBUJE (2), any grand relation, and those of wife ; a title of respect.
AMBUSANGA (2), friend (of same sex), paramour (of opposite sex).
ĂNĂ ? (*indicates a question*).
-ANA, *adj. pron.*, having, of : *Juana machili*, a strong man.
-ANA -OSE, *adj. pron.*, every one : *Mundu juana juose*, every single man.
ANAGA, *conj.*, if.
-ANGALI, *adj. pron.*, not having, without : *Mundu juangali machili*, a weak man.

Figure 3.3: Example of a wordlist (From M. Sanderson, *A Yao grammar*, 1922.)

APPENDIX VI.

VERBAL PREFIXES.

1. **Nga-** (*ngi-*, *nge-*, *ngu-*, *ngo-*).

(1)	Neg. Relative,	juan**ga**lola, *one who does not look.*
(2)	Neg. Present,	**nga**lola, *he does not look.*
(3)	Neg. Future,	**nga**lola, *he will not look.*
(4)	Incomplete,	**nga**nalole, *he has not yet looked.*
(5)	Negative Past,	**nga**nalola, *he has not looked.*
(6)	Past Conditional,	**anga**lole, *if he had looked.*
(7)	Pres. Contingent,	**anga**lolaga, *he would be looking.*
(8)	Past Contingent,	**anga**lolite, *he would have looked.*
(9)	Neg. Contingent,	**nga**kanalola, *he would not have looked.*

Figure 3.4: Example of a verbal paradigm (From M. Sanderson, *A Yao grammar*, 1922.)

page bitext and use it to identify inline bitext. Figure 3.6 shows an example of what facing-page bitext looks like.

Inline Bitext

Inline bitext occurs when an utterance in a foreign language is given, accompanied by a translation of that utterance, within a larger text. This type of bitext uses the fewest typographic cues; although the foreign text is often printed in a unique font face (e.g. italic text), this information is generally not preserved in the OCR output. However, its comparatively simple layout can be a benefit when dealing with imperfect OCR; since there are no complex columnar or tabular layouts involved, inline bitext is unproblematically recognized as contiguous text.

In our work, we focus mainly on this type of bitext, using statistical approaches to first identify spans of text that appear to belong to the foreign language, then using a translation model to determine which of the surrounding text constitutes the gloss of that span of text. While this approach may also capture other types of bitext mentioned here, we do not do any special processing to accommodate or exploit the unique layouts of wordlists, paradigms, and tables, instead leaving that for future work.

3.3 Building a Corpus of Digital Grammars

3.3.1 Collecting Online Documents

A critical first step in collecting language data from digital sources (for example, the various digital repositories discussed in chapter 2) is to identify documents of interest. For this project we utilized a hands-on approach, manually searching for books with specific keywords. Our searches focused on texts in the University of Michigan's Digital General Collection, which includes books that have been scanned

(ANIMATE.) *DATIVE WITH NOMINATIVE.*

TENSES.	ADJECTIVE PARTICIPLES.	ADVERBIAL PARTICIPLES	GERUNDS.
FUTURE.			
Dal-joiṅ-a-e, He will strike for himself.	*Dal-joṅ*,* Who will strike for himself.	*Da-joṅ-khan*, Striking for himself.	*Dal-joṅ-reak, te, re*, Of, by, in striking for himself.
SPECIAL INCOMPLETE PRESENT.			
Dal-joṅ-kan-a-e, He is striking for himself.	*Dal-joṅ-kan*, Who is striking for himself.	*Dal-joṅ-kan-khan*, Striking now for himself.	*Dal-joṅ-kan-reak, te, re*, Of, by, in striking for himself now.
RECENT PAST.			
Dal-a n-a-e, He struck or has struck for himself.	*Dal-an*, Who struck for himself.	*Dal-an-khan*, Having struck for himself.	*Dal-an-reak, te, re*, Of, by, in having struck for himself.
PERFECT.			
Dal-akao-an-a-e, He has struck for himself.	*Dal-akao-an*, Who has struck for himself.	*Dal-akao-an-khan*, Having struck for himself.	*Dal-akao-an-reak, te, re*, Of, by, having struck for himself.

Figure 3.5: Example of a paradigm as a table (From L. O. Skrefsrud *A grammar of the Santhal language*, 1873).

zāwa," äicitähätci. Ähägōzītcitcā me'tegwi; wāgigenigi me'tegwi. Inā ä'pyätci äpemegi, "Tcinänänänä!"[1] ähitci ä'kwäskwizahutci nepigici. Me'tegwitcii. Äpetcämätci! 'Ō, ä'pa'kitäcigi, "'Ō'ho'hwa', necōskonāwa mecinämäza!" ähinätci Tcinänähäni. "Hō, kīne'tōne kutäga," ähinätcitca īna Tcinanā'a äne'taätci kutägäni. Äwätcähetci Tcinänäa. Käpōtwe kīcizenyätci ä'penutci. Inimegä'kwike'känemägi.

2. Wīza'kä'ą äēgi Tā'u'wäą.

Äcäcegicigi Wīza'kä'ą. Äataataäpäcigi ä'pemegi ähinäpitci. "O kätena mäye menwigenwi maça'kwīnigäni. Täniyuyätuge ämō'tcīnakaskipyääni," äicitähätci.

Käpōtwänäätci Tā'huwääni. "Tā'huwä necizē, pyänō!" ähinätci. Ä'pyänitcitcā. "Nahi', wīawaciyäni kekataänemene kīcegugici," ähinätci. "Īnī," ähigutci, "Kihawanene. Inā 'ku 'wīna nepäpya kīcegugi," ähigutci.

Äääpuzäätcitcąį.[2] Käpōtwänahi äpyäätci kīcegugi, "Nahī', ayō ainu, nenegwa, Wīza'ke," ähigutci īnini Ta'huwä'äni, "Nīnäte mäya äwazi memänwigenwi maca'kwīnigäni," ähigutci īnini tä'huwä'äni. Äzägenamä'kwītcitcąį kīcegwi ä'tcīgyänigi. Kapōtwä'paipyänitci Tā'huwä'äni äaskakōtcigi. "Ta'huwä necizē!" äcäcōgegi, "Ta'huwä necizē!" äcäcōgegi awäzi, "Ta'huwä necizē!" Äicgwäegezitci wiicinenu'tägutci.

[1] Observe the word is bungled. [2] Read *ääp-*.

good deal! I will kill a much larger one," he thought in his heart. Verily then he climbed a tree; it was a crooked tree. When he got up aloft, "Kingfisher," he said as he jumped off toward the water. Lo! it was a tree. He made a mistake! He was knocked senseless. "Oh, ho, ho, I missed a big fish!" he said to the Kingfisher. "I will kill another for you," that Kingfisher said to him truly. Then he killed another for him. Then a meal was prepared for the Kingfisher. Soon after he had eaten he went home. This is as far as I know.

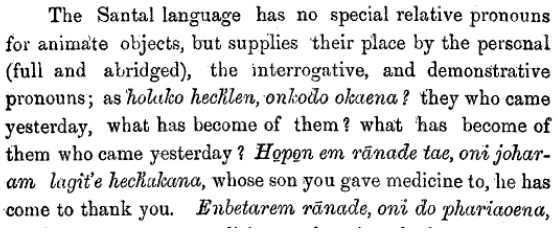
2. Wīza'kä'ą and Buzzard.

Wīza'kä'ą was lying down. As he lay on his back he looked up at the sky. "Oh dear, yonder is fine arrow-paint. I wonder how I could get up there," he thought in his heart.

Suddenly he saw Buzzard. "Oh my Uncle (mother's brother) Buzzard, come!" he said to him. Verily the other came. "I will now earnestly beg of you that you take me up towards the sky," he said to him. "All right," he was told, "I will carry you. Frequently do I go up there in the sky," he was told.

Verily they started to get there. Soon when they came to the sky, "Well you stay here, my nephew (sister's son), Wīza'kä'ą," he was told by that Buzzard, "I will go after the very best arrow-paint yonder," he was told by that Buzzard. Then he got a hold of the edge where the sky extended. After a while when Buzzard did not return, he was tired hanging. "Oh my Uncle Buzzard!" he whistled, "Oh my Uncle Buzzard!" he whistled louder, "Oh my Uncle Buzzard!" He made a big noise, so that he could be heard by him.

Figure 3.6: Example of facing-page bitext (from W. Jones, *Kickapoo Tales*, 1914).



The Santal language has no special relative pronouns for animate objects, but supplies their place by the personal (full and abridged), the interrogative, and demonstrative pronouns; as *holako hecñlen, onkodo okaena?* they who came yesterday, what has become of them? what has become of them who came yesterday? *Hopon em rānade tae, oni joharam lagit'e hecñakana,* whose son you gave medicine to, he has come to thank you. *Enbetarem rānade, oni do phuriaoena,*

Figure 3.7: Example of inline bitext (from L. O. Skrefsrud *A grammar of the Santhal language*, 1873).

for preservation and are available to download in their entirety, both as page image and as OCR. Texts were identified using search queries, for example searching the for the word "language" in the subject field (which matches subject code like "Thai language – dictionaries" or "Czech language – Grammar"), as well as searches for terms like "Grammar of" or "Dictionary of" in the title of the book. Although some approaches to language collection, such as that of The Crúbadán Project (Scannell, 2007), crawl through documents searching for text that appears to be written in a foreign language, we would unable to adopt this approach for our work without downloading all of the texts in advance, which would be prohibitively expensive both in time and storage space.

Our searches yielded more than enough results for initial work, and we created a list of 110 relevant documents, mainly from Google Books and the U-M Digital General Collection. Not all of these texts were suitable for automated processing: for example, some used non-Roman orthography, which is not recognized by the OCR software that was used for these collections. Ultimately, we selected a subset of 20 books for annotation and additional processing. The texts we chose and some basic statistics about this corpus are given in table 3.1.

3.3.2 Annotating Foreign Text

We designed and created a custom annotation tool that allows users to select portions of the text and label it as foreign text. The annotation software is written

Bilingual Texts	11 (Caddoan, Fox, Haida, Kickapoo, Koryak, Kutenai, Maidu, Menomini, Ojibwa, Passamaquoddy, Zuni)
Dictionaries	2 (Burmese, Hungarian)
Grammars	7 (Arapesh, Filipino, Italian, Navaho, Malayan, Pangasinan, Santhal)
Annotated pages	304 (from 9 documents)
Total pages	7,479
Total words	780,000 (estimated)

Table 3.1: The make-up of our corpus of scanned linguistics documents.

in Javascript, using a Python back-end server, and this allows annotators to access the tool from their web browser either locally or remotely. The annotation scheme allows annotation of foreign text on its own, foreign text accompanied by an English gloss, or foreign text (with or without an English gloss) accompanied by grammatical category labels (e.g. terms like "singular", "plural", "noun", or "past tense").

The goal of annotation is not to mark up entire documents, but to create a set of annotated pages which can be used for evaluating the performance of automated annotation methods. In all, 131 pages of text sampled from nine documents were annotated by five different annotators.

We chose a small subset of pages to be annotated by more than one annotator, in order to get a measure of inter-annotator agreement. The annotator-pair agreement rates are given in table 3.2. The agreement rate is simply the fraction of tokens that the annotators assigned the same label to. The kappa statistic (Carletta, 1996) is a measure of inter-annotator agreement that takes into account the expected rate of accidental agreement between annotators. Kappa is defined as $\kappa = \frac{P_a - P_e}{1 - P_e}$ where P_a is the actual rate of inter-annotator agreement and P_e is the expected rate of accidental agreement, based on the relative frequency of the different tagging categories. For this data, the rates of two of the three categories (foreign text, gloss of foreign text) are roughly similar, but the third category (unlabeled) is much more frequent, leading

Annotators	Agreement Rate	Kappa
1,2	0.98	0.94
1,3	0.94	0.86
2,3	0.93	0.84
Overall	0.95	0.88

Table 3.2: Inter-annotator agreement rates, for three annotators on a subset of data comprising 5 pages and 946 tokens.

to a relatively high number of expected chance agreements. Kappa values range from zero (agreement equal to chance) to 1.0 (total agreement), and a kappa of 0.5 or higher is generally considered a good level of agreement. The results given in the table show that there is strong inter-annotator agreement, both by the raw agreement rate and by the kappa statistic, which is encouraging for the possibility of high-accuracy automated tagging.

3.3.3 OCR Challenges

OCR technology is better today that it ever has been, and OCR text is perfectly acceptable for a variety of uses. However, for the research described in this chapter OCR is a major issue which needs to be addressed. The first problem is that the texts that we have collected are particularly prone to OCR errors. To avoid copyright issues, many of the books we collected are around one hundred years old; the printing is not as clear as modern books, and it may have faded over time; physical wear on the pages results in specks in the scanned image, affecting the OCR. Several of the dictionaries and grammars we originally identified use non-Latin scripts, which are either skipped entirely by the OCR software or produce gibberish output; in both cases the result is unusable for our purposes.

Even when the foreign-language text uses Latin-based orthography, it is often embellished with various diacritic marks which produce errors in the OCR. Figure 3.8 illustrates a typically frustrating example: the grammar presents a paradigm of the

Scanned Image

Instr. *Ṭaṅga-te,* by, with, the axe.
Dat. *Ṭaṅga-ṭhen,* to the axe.
Acc. *Ṭaṅga,* the axe.
Abl. *Ṭaṅga-khọn, khọcḥi,* etc., from the axe.
Loc. *Ṭaṅga-re,* in, on the axe.
Voc. *e Ṭaṅga !* O, axe !

OCR Text

```
Instr.  Tasga-te, by, with, the
axe.  Dat.  Taiga-then, to the
axe.  Acc.  Tagga, the axe.  Abl.
Tariga-khon, khoci, etc., from
the axe.  Loc.  Tatiga-re, in, on
the axe.  Voc.  e Talga!  0, axe
```

Figure 3.8: Comparison of a portion of a scanned page and its OCR output.

noun *Ṭaṅga*. Although the stem is identical in all six forms of the noun, the OCR software has rendered the same stem in six different ways: *Tasga*, *Taiga*, *Tagga*, *Tariga*, *Tatiga*, and *Talga*. This type of error poses a serious impediment to our hopes of using the OCR text for statistical inference (such as morphology induction or statistical word alignment), since those processes rely on items recurring multiple times in order to estimate model parameters. This example also illustrates how the OCR text omits information that is present on the printed page which indicates how the table is to be interpreted: by discarding line breaks and spaces, the original tabular format is obscured; and by normalizing font variations (e.g. bold or italics), an important indicator of foreign vs. English text is lost.

Many of these problems could be resolved by making different choices when generating the OCR text. When dealing with the typical prose documents that probably constitute the majority of a library's holdings, it makes sense to ignore line breaks. And if the text mainly consists of long passages of running English text, then it may be that OCR-detected "weird" characters (like a *t* with a dot under it) are more likely to be the result of noise in the image than actual non-English writing. But these are not the appropriate choices for multi-lingual texts of the sort I am interested in.

Fortunately, OCR software does exist that is capable of preserving this information. Unfortunately it costs money and time to re-process the page images using this software. After beginning work with the OCR provided with the online documents

and realizing that it was problematic, we performed some initial trials using a commercial OCR software product, ABBYY, to create our own OCR text, and the results are encouraging. However, due to time constraints we were unable to utilize the new OCR text in this stage of the project. As future work, we hope to produce new OCR versions of the texts, and we will also need to update the software we have written to accommodate the new OCR texts, since our software was designed to handle plain text and the improved OCR is packaged in an HTML format.

3.4 Language Identification in Bilingual Texts

Linguistics documents are unique in that they are bi- or multi-lingual, combining text from multiple languages in a single document. Outside of texts which are explicitly about language, such as grammars, dictionaries, language textbooks, or bilingual readers, it is rare to find texts that combine significant amounts of material from multiple languages. While it is not uncommon for a document to contain a foreign phrase or quotation, these usages tend to be sporadic, whereas the language-switching in linguistic texts is pervasive throughout the entire document.

Perhaps because true multilingual texts are rare, there is fairly little prior research on automatic language identification of individual words within a text. While language identification of entire documents is a well-studied problem, research on this topic assumes that the material to be identified is a relatively large quantity of monolingual text, and that the language belongs to a limited set of known languages. In that scenario, the normal approach to language ID is to compare the text to samples of known text from a variety of languages and identify the sample that best matches the test data. While it is possible to achieve 99% identification accuracy using samples of just a few hundred sentences apiece (Kruengkrai et al., 2006), these approaches still require a sample of monolingual text for training.

The creators of the ODIN corpus of interlinear glossed text faced a slightly different

variation on the language ID problem; in their case the IGT instances are already identified within the text, but each IGT needs to be associated with a language. The texts to be identified (either a single IGT or a handful of IGTs from the same document) are exceedingly small, and the number of languages to choose from is large. The ODIN researchers overcame these challenges by using an approach that combines contextual information (i.e. names of languages mentioned in nearby text) with character information (Xia et al., 2009).

The language ID task for our project, however, is significantly different from both previous forms of language ID. Our goal is generally not to identify the language from a set of languages; most of the documents we are interested in reveal the identity of the language in the title (e.g. the title "A Grammar of the Santhal Language" reveals that the text is written in English, and that it contains additional text written in Santhal). Rather, we aim to identify the language of individual words within a text more or less independently of one another. Identifying the language of a single word is a much more challenging task than identifying the language of a complete text, and it is even more challenging when there is no external source of information about the language in question.

Since the reference language is presumed to be known, and is often English, one approach to language ID for bilingual documents is to simply classify each word as English or non-English. Since monolingual English text is plentiful, and so are English dictionaries, one can easily build a model of English to be used in a classifier. The most significant hurdle to this approach is that errors due to the OCR process can often make English words look very un-English-like.

Our group performed some initial experiments with several methods for word-level language ID. These experiments can be broken into two categories: in the first category the foreign language is unknown, in the sense that we have no monolingual data available to train a model of that language. In the second case the foreign

language is known, meaning that we have at least some monolingual text from that language.

3.4.1 English vs. Unknown

When the foreign language is unknown (or it is known but we have no additional data from that language), the classification problem is essentially one of distinguishing English words from non-English words. The natural approach to this problem is to use a dictionary. However, due to OCR errors many English words will not match their dictionary entries, and so a statistically relaxed approach is necessary to accommodate these errors.

One approach to this problem is to train an n-gram model of English orthography and select a decision boundary to distinguish English from non-English text. Using English (51k words), Dutch (113k words), and German (49k words) tokens from the Celex database, I trained a letter-level trigram model of English and used it to assign probability scores to all the words in the corpus. When the trigram model is used to estimate the probability of a given word token, it generally assigns a lower probability to Dutch and German words than it does to English words; therefore we can use the probability score of each word as an estimate of which language produced that word. For a given language pair, I calculate a decision boundary that optimizes the classification rate. Any word with a probability above that boundary is classified as English, and any word with a probability below that boundary is classified as non-English.

First, I estimated the decision boundary for German and Dutch separately. In all cases the same trigram model was used, trained on the English corpus. For the English-Dutch classification task, the optimal decision boundary was $-logp = 8.8$ and yielded an 86.3% classification accuracy for distinguishing English words from Dutch words. In the German-English case, the decision boundary was $-logp = 10.1$,

	Evaluate on Dutch	Evaluate on German
Train on Dutch	86.3	86.4
Train on German	85.8	87.8

Table 3.3: N-gram-based language ID results on Celex data.

with a 87.8% accuracy.

In order to estimate the performance of the letter-trigram decision-boundary method on *unseen* language, I applied the German-trained decision boundary to the Dutch data, and the Dutch-trained decision boundary to the German data. This resulted in classification accuracies of 84.8.% and 86.4% respectively. These results are summarized in table 3.3.

As expected, training and evaluating the model on the same language yields better results than using a decision boundary trained on a separate data set, but in these experiments the difference is not particularly large. That is, even if we do not have access to the non-English language in question, this classifier may still be effective using a decision boundary that was trained on data from a different language. Since German and Dutch are both linguistically fairly close relatives of English, it is not unreasonable to expect that performance of this classifier would only improve when evaluated on other non-English languages with even less-English-like spelling. Still, the accuracy is still far less than perfect, even when training on these fairly large corpora.

3.4.2 English vs. Known

When the foreign language is known and we have a sample of monolingual text in the foreign language, then it is possible to train a discriminative model. In order take advantage of our annotated data for evaluation purposes, we gathered a separate monolingual sample of 39k Arapesh words. For English, we used a large dictionary as well as a collection of English names. Using a combination of unigram, bigram,

Classifier	Accuracy
SVM	78%
Naive Bayes	72%

Table 3.4: English vs. Arapesh language ID results.

and trigram letter features, we trained a support vector machine and a naive Bayes classifier. The results on our small test sample of Arapesh text is given in table 3.4.

In addition to these classifiers, a dictionary-based approach was employed based on an English dictionary and a list of proper names. This approach achieved 79% accuracy. Although the dictionary approach currently has the best performance, it may be possible to improve the SVM performance adjusting the features or the parameters in the model. There is also likely a difference between the data used to train the model (taking from an HTML source) and the OCR text it is tested on. In any case, the language ID problem remains open for additional improvement in future work.

3.5 Gloss Identification

3.5.1 Gloss Selection Using a Statistical Translation Model

Once foreign text has been identified in a document, we would like to be able to identify nearby English text that acts as a gloss of the foreign text. In the case of inline bitext, the gloss is either immediately preceding or immediately following the foreign text, but we do not know which. Using a statistical translation model could help identify the gloss, but first we need to train the translation model.

In the absence of a separate corpus of bitext to train the translation model, we are forced to somehow train a translation model without knowing in advance what the bitexts are. A straightforward approach to this problem is to simply train a translation model on all of the noisy data, including two candidate bitexts (one with

Sentence and Candidate Translations		Cost
"He abused our trust."		
a)	Il a abusé de notre confiance.	18.5
b)	Il éclata en larmes.	40.3
"The floor was covered with blood."		
a)	Le sol était couvert de sang.	15.9
b)	La machine était recouverte de poussière.	46.7

Table 3.5: Example alignment costs of true and false translation pairs. In both cases, the cost of aligning the true translation is much less than the cost of aligning the false translation.

the preceding text, and one with the following text) for each span of foreign text in the document. Then, for each foreign sentence, we can compare the alignment costs of the two candidate translations and choose the one with the better score. This concept is illustrated in table 3.5, which shows a couple of English sentences, each accompanied by a pair of French sentences. The first French sentence in each case is the actual translation, while the second sentence is a non-translation. The alignment costs for the true translations are much lower than the costs of aligning the English sentence with the non-translation sentences.

To summarize, my proposed method for detecting the proper gloss is as follows:

1. Use language ID to detect regions of foreign text in the document.
2. For each foreign text, create two candidate bitexts, taking text from before and after the foreign text.
3. Train a translation model on all of these bitexts, holding out a sample for prediction.
4. Use the translation model to align the held-out sample. For each foreign text, the candidate with the lower alignment cost is chosen as the true translation.

In the following sections, I describe two experiments that illustrate the effectiveness of this approach on data sets of varying size and quality. Setting aside the issue

of language identification, in the first experiment I focus on evaluating my method for gloss selection from two candidate translations.[1] The data set is a taken from an online database of English-French translation pairs. The second experiment is carried out on the noisy OCR data from one of the texts in our collection of scanned grammars.

3.5.2 Experiment 1: French

In order to determine the efficacy of this approach, a first experiment was performed using a corpus of true bitext, with noise artificially added. The data consist of all the French-English sentence pairs from the Tatoeba database, an open, online collection of user-submitted translations. (More information on the Tatoeba database can be found in Appendix B.) In order to generate bitext pairs, each English sentence was paired with its actual translation as well as another French non-translation. Because a length mismatch will increase the score simply by increase the number of individual word alignments, the non-translation was chosen to closely match the length of the original sentence.

This experiment is also an exploration of how well this approach can be expected to work on various types of data. To explore the effect of corpus size, I created subcorpora of 500 and 5,000 sentence-pairs, in addition to the full corpus of 53k sentences. To see how effective a translation model trained on noisy data is, I performed the sentence-selection process for each corpus under two scenarios: in the first, "gold" scenario, the translation model was trained only on the true translation pairs; in the second, "both" scenario, the translation model was trained on both the true and the false sentence pairs. Thus, the "both" scenario is trained on a corpus with twice as many sentences, but half of those sentences are not in fact actual translation pairs. This mimics the actual case encountered in bitext extraction, where the true

[1]The same experiments could be carried out with more than two candidate glosses, using essentially the same procedure, but here I focus on the case in which there are only two candidates.

Corpus size (sentences)	Accuracy (train on gold)	Accuracy (train on both)
500	71.2% (4.7)	72.8% (5.2)
5,000	89.3% (.98)	87.9% (1.3)
53,129	95.4% (.15)	94.4% (.11)

Table 3.6: Accuracy of true bitext selection for experiments on different corpus sizes. Accuracy is averaged over five folds of cross-validation, with standard deviation in parentheses.

translations are not known in advance. The gold scenario represents an upper bound which is not achievable in practice.

Each experiment was carried out using five-fold cross-validation. So, for example, in the 500-word case, five trials were carried out in which the translation model was trained on 400 sentence-pairs, then used to produce and score alignments on the remaining 100 sentence pairs. (Actually, it is trained on either 400 or 800 sentence-pairs, depending on whether the false translations are included, and it is evaluated on 200 sentence pairs: two French translations for each of 100 English sentences.) Table 3.6 shows the accuracy achieved in each scenario, averaged over the five folds with standard deviation given in parentheses. Accuracy is defined as the percentage of test sentences for which the true translation received a better alignment score than the false translation.

From these results, it is clear that the size of the corpus has a strong effect on the prediction accuracy, which is expected. Also, training on only the true translation pairs improves the prediction accuracy, but this effect is not that large, and for the 500-word corpus any advantage this may have offered is completely obscured by the noise associated with training on a small data set.

Because the corpus size affects accuracy to a large degree, I performed a second trial on the 500-word data set using 100-fold cross-validation: thus, instead of 400 sentences per fold, the model is trained on 495 sentences per fold. This trial achieved a mean accuracy of 70.8% across the 100 folds (22.8% standard deviation). Thus,

increasing the number of folds did not have a major effect on the performance, at least for this particular data set.

It is important to keep in mind that for the smaller datasets, it is likely that corpus effects are significant, and that a different 500-sentence subset of the full corpus would have different characteristics than this one. Still, the overall trends are obvious, and even for the smallest corpus, this approach to gloss selection gives a much better performance (70%) compared to a random-choice baseline (50%), with significantly better performance for larger corpora. Therefore it seems safe to conclude that it is possible to effectively use a translation model, even one trained on noisy data, to select true glosses from a candidate set containing both true and false glosses.

3.5.3 Experiment: Santhal

In this experiment, predicted bitexts are extracted from the OCR text of *A grammar of the Santhal language*, which was downloaded from the U-M Digital General Collection.

First, all word tokens in the text are classified as either English or Santhal using the SVM classifier, trained on the portion of the text which was manually annotated. Next, all spans of two or more sequential foreign words (ignoring non-word tokens such as numerals or punctuation) are collected. Each span of foreign text is associated with two candidate translations; one consists of the sequence of tokens immediately preceding the foreign span, and the other consists of the sequence of tokens immediately following the foreign span. The size of these candidate translations is determined by adding tokens one a time until the length (in characters, excluding punctuation and whitespace) of the candidate translation is equal to or greater than the length in characters of the foreign span. Finally, the statistical gloss selection procedure described in the experiment above is applied to select the best translation for each bitext.

Question	Yes	No	Pct
Is the predicted foreign text actually foreign text?	99	1	99%
Is this actually an inline bitext?	69	31	69%
If this is an inline bitext, is the prediction *approximately* correct?	19	50	38%

Table 3.7: Santhal bitext extraction evaluation questions.

This procedure produces 3,503 predicted Santhal bitexts. To evaluate the quality of these predicted bitexts, I chose a random sample of 100 predicted bitexts for manual inspection. None of the predicted bitexts is exactly perfect; even the most accurate predictions fail to precisely identify the beginning and end of the foreign or English spans. Therefore, in order to get a softer measure of performance, I asked three yes/no questions of each predicted bitext: these questions and the overall responses are given in table 3.7.

The first question is meant to assess how well the language ID component performed. 99 out of the 100 bitexts I inspected were in fact centered on foreign text, indicating that the precision of the SVM language classifier, when combined with the two-or-more token restriction, is sufficiently high. (The recall is unknown; it could be calculated given additional labeled data, but I used all the existing labeled data to train the SVM classifier.)

The second question addresses the fact that not all instances of foreign text are inline bitexts. That is, some instances of foreign text do not have an English translation immediately preceding or following the foreign text. In the sample of 100 predicted bitexts, 69 were classified as inline bitext, meaning that an English translation was present immediately before or after the span of foreign text, and therefore retrievable in principle. The 31 remaining instances were mainly cases where the foreign text occurred within a table: in the Santhal grammar, translations are typically given one row above or below the foreign text within a table. This means that the translation is not directly before or after the foreign text span within the OCR text, and thus it

had struck him. had struck him. he had struck hitn.
DUAL. DUAL. DUAL.

1. I D-al-a1,kat'-ti;4-**ta- Dal-akat'-li.-tcth'-** Paset'-e-dat-a~cat'-liti..
lt-1can-a-e, He kan-A-han-e, If tcth~1oan, Perhaps
had struck us he had struck us he had struck us

strike.
INCHOATIVE PAST.

2. **Dal-Jko-dagido11-kan-tahVkan,**
Tlhey whom they were about
to strike.
OPTATIVE.

oni hola-m del-led-e, what has become of him whom you
saw yesterday? This is much more elegant and certainly more

3. correct than to say: **oni hola-m diel-ed-e-a, oni** do okare,
for the latter means literally: you saw him yesterday, what
has become of him?

Figure 3.9: Three examples of predicted bitexts from the Santhal grammar. The predicted foreign span is in bold, and the predicted translation is underlined.

is not retrievable without more sophisticated techniques for identifying tabular text.

As mentioned above, none of the predicted bitexts is perfect. Therefore, I adopted a loose standard for determining if the predicted bitext is *approximately* correct. The following criteria were employed: whether the predicted foreign span covers the majority of utterance it belongs to; whether the predicted translation is in the correct direction (right or left); and if so, whether the predicted translation roughly covers the majority of the true translation.

Figure 3.9 gives three examples of the predictions made by this procedure. The first example shows a three-column table, which are common in the Santhal grammar. The fact that the foreign text is accurately identified is largely luck, since the text preceding and following it (from the two neighboring columns) is also foreign text. This example illustrates the need for a method to detect the table structure and deal with it appropriately. The second example illustrates a case where the prediction is correct; this comes from a single-column table, so adjacent columns do not pose an issue. The third example shows foreign text within a paragraph; the foreign span is cut short (by the presence of the word "do"), and the translation is misidentified. In this particular case the translation is non-adjacent; although cases like this are

relatively rare, they pose yet another challenge for bitext extraction.

The hit rate of 19 correct out of 100 predicted bitexts leaves something to be desired. However, in 31 instances it would be impossible to identify the English translation simply by looking at adjacent text. If we omit these cases, then the hit rate increases to 38%. If the foreign text spans were detected perfectly, then a simple baseline of always choosing the text to the left or the text to the right would be expected to be correct 50% of the time overall. However, the most common reason for a predicted bitext to be judged incorrect is that the foreign span is too short. If the foreign span is predicted too short, then this will usually throw off the range of the predicted English translation as well. A smaller number of predicted bitexts are judged incorrect because the gloss selection was incorrect (e.g. choosing left instead of right).

3.6 Conclusions and Future Work

The results of the French experiment shows that it is possible to use a statistical translation model to select the correct translation of a sentence, even if the translation model is trained on a corpus in which the true translations are not known beforehand. This technique outperformed a baseline even for a very small test corpus of 500 sentences, suggesting that it is appropriate for use on the types of small datasets associated with resource-poor languages. While these results are promising, the results of the Santhal experiment show that the same approach is much less effective when applied to OCR data from the wild. Similarly, the experiments on language identification within bilingual OCR documents are reasonably effective, but leave room for improvement, particularly for accurately detecting longer spans of foreign text.

The errors caused by not detecting the full span of foreign text could be overcome by using a sequential model, such as a Hidden Markov Model, to label sequences of foreign words in a soft manner. This should help in cases where an English-looking

word appears in the midst of a sequence of foreign words. For example, in Santhal the tokens “an”, “a”, “do” and “than”, among others, could be either English or Santhal, depending on the context. (In addition to truly shared words, noisy tokens also pose a challenge.) When such words occur within a Santhal sentence, they incorrectly cause a break in the predicted foreign span.

The gloss selection procedure is based on statistical word alignment, which relies on recurring correspondences between foreign and English words in the training data. However, errors in the OCR text can obscure these correspondences (see figure 3.8 above, which shows how a single Santhal word is mis-recognized by the OCR in several different ways). Thus, while the gloss selection process was shown to be effective on the noise-free French data set, it is less effective on the noisy Santhal data. Another challenge for the alignment-based gloss selection process is that neighboring bitexts within a grammar tend to be relatively similar. For example, one table in the Santhal grammar contains text and translation of the phrases “they will strike him”, “they will strike it”, “they will strike for him”, “they will strike at it”. The high number of shared words among these English phrases makes it difficult for a statistical approach to accurately select between them.

In one regard, however, the alignment-based approach to gloss selection is overkill for this application. Within a given document, and especially within a given section of a document (e.g. a paragraph or a table), the ordering of text and translation is consistent. Thus, it may make more sense to use the statistical alignments to determine the optimal direction to look for glosses, and then apply that direction uniformly within a given document or section.

Other areas for future work include re-processing the text using different OCR software and developing techniques to better handle data in tabular, rather than paragraph, layout. The layout question may at least partially be answered by fixing the OCR, since some OCR software will detect tables and represent them ap-

propriately (e.g. using HTML markup rather than plain text). As mentioned, we experimented with ABBYY, a commercial OCR product which claims to be able to recognize text in roughly 150 languages, including several non-western scripts including Arabic and Chinese. Our group's initial experiments with ABBYY yielded positive results; although we did not perform any direct comparisons, the OCR quality appeared better than the OCR provided with the texts. One drawback is that while ABBYY supports characters with diacritic marks, ABBYY requires the user to specify the language prior to recognition; there is no way to run the software in a language-agnostic mode. Still, there is reason to be optimistic that the OCR quality can be improved relative to the current state. Not only does ABBYY produce better character recognition, it is capable of preserving typographic and layout information (for example, font-face and tabular layouts) that is not present in the existing OCR text. However, even ABBYY cannot produce perfect results, and any future work will need to remain tolerant of noisy data.

I am particularly hopeful about the possibilities for utilizing typographic and layout information to identify some of the more structured varieties of bitext, such as wordlists and paradigms. One possible approach would be to use a Hidden Markov Model in which emission probabilities are related not only to the language models but also to the formatting information; thus a change from bold to normal font might also indicate a change from a foreign-text-generating state to an English-generating state. While belonging to an entirely different domain, this is conceptually related to work using HMMs to extract structured information from classified ads (Grenager et al., 2005), or using conditional random fields to convert plain text bibliographic entries into a regular database format (Mansuri and Sarawagi, 2006).

Grammars contain a wealth of information about the language in addition to the primary data (i.e. bitexts) that are considered here. While extracting an author's linguistic analysis from his prose is a natural-language understanding task that far

exceeds the capability of existing NLP methods, there is a reasonable expectation that some aspects of the linguistic analysis could be recovered automatically. For example, in addition to housing wordforms, paradigms in a grammar often have row and column headers that indicate how to analyze the forms within that paradigm. Extracting this information is yet another area for future work.

Ultimately, the types of documents described in this chapter contain a wealth of information, and they have much to offer to researchers in linguistics and computational linguistics. However, these documents are challenging to work with in their current state, and there remains much additional research and work to be done before their full utility is realized.

Despite the many challenges yet to be fully dealt with, one very important outcome of this project is the corpus of texts we have collected and annotated, which constitutes a unique and valuable resource for future work in this area. As the only collection of its kind that I know of, this corpus will be essential for continued research on improved language identification methods and techniques for identifying glosses of foreign text within bilingual documents.

CHAPTER 4

Generating Biforms from Parallel Text

4.1 Introduction

Next to monolingual text, the most commonly-occurring type of language data both in print and on the web is parallel text, in which a document in one language has been translated into another language. While parallel text comes in many varieties, the fundamental feature shared by all parallel text is that it represents the same information in two (or more) languages, and this is what makes it so useful to language researchers: modern statistical machine translation systems rely on large quantities of parallel text to build translation models, and linguists may use a parallel text to begin to analyze the structure of an unknown language (consider, for example, of the role of the Rosetta Stone in deciphering ancient Egyptian hieroglyphics, or of using a bilingual informant to collect data from an unknown language). The amount of information contained in parallel text is vast, particularly if one of the two languages is known to the researcher. However, none of this information is explicitly encoded, and must be extracted through careful analysis.

For present purposes, I am only interested in bitext which fits this scenario, in which one language is known and the other unknown: the side of the bitext in the well-known language is referred to as the English side; and the other side of the bitext, in an unknown language, is referred to as the foreign side. The "English" side does

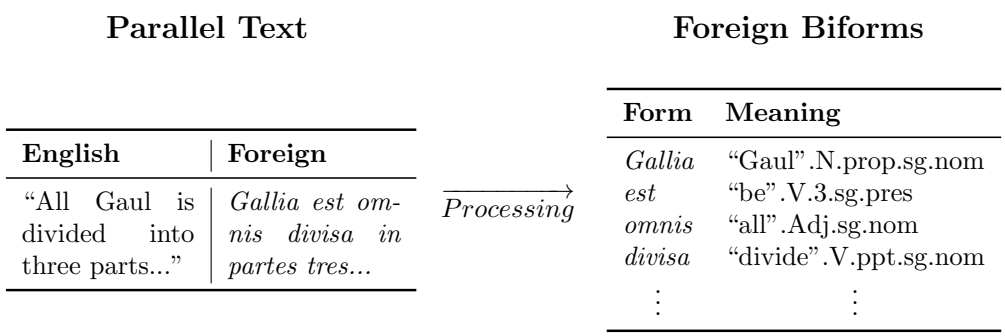


Figure 4.1: The high-level objective of biform generation.

not necessarily need to actually be English; it could be any language for which there exists software to performing parsing and morphological analysis. Also, the foreign language is not strictly unknown; we probably know at least the name of the language and perhaps much more. However, it is unknown in the sense that we do not have the software tools to analyze the foreign text in the way that we do have tools to analyze the English text.

In this chapter, I describe an automated technique for performing word analysis (essentially, rich part-of-speech tagging) of a foreign language, starting with parallel text written in that language and English. This process is illustrated at a high level in figure 4.1. The English text is processed using existing software tools for analyzing English sentence and word structure, then aligned at the word level to the foreign text, again using existing software. In the final stage, features extracted from the English text are assigned to wordforms in the foreign language, resulting in a output of foreign word tokens tagged with morphological features, which I call *biforms*: two-part word tokens that consist of both a string form and a feature-based representation of the meaning and grammatical properties of the word. The various stages of this process, and the sections of this chapter which discuss each of them, are listed below.

Section 4.2	1.)	Collect the parallel text data from its original source.
	2.)	Perform word and sentence tokenization on both English and foreign text.
	3.)	If the data is not sentence-aligned, then perform sentence alignment.
Section 4.3	4.)	Parse the English sentences.
	5.)	Perform morphological analysis on the English tokens.
	6.)	Extract morphosyntactic features for each English token, based on the output of steps 4 and 5.
	7.)	Perform foreign-side analysis, if desired.
Section 4.4	8.)	Perform word-to-word alignment on the bitext.
	9.)	Transfer morphosyntactic features from English tokens to foreign tokens via the alignment from step 8.

Much of the processing is performed by existing software that is freely available for research use, and this software is discussed in detail in later sections and the appendices. Use of such software is essential for the feasibility of a study like this, and it has the added advantage of making this research open and easily repeatable. Extensive scripting is used to convert data to and from the various formats required by each piece of software: after the initial tokenization, data is maintained at all times in a simple format of either one token per line, or one sentence per line, which allows the output of various components to be conveniently stored in a stand-off manner in separate but parallel files. Additional technical details about these software packages and the scripts and data formats can be found in Appendix A and in the code documentation.

In section 4.5 I give an analysis of this system's performance on a test corpus of

parallel English–Estonian text, and show how the system can be refined for improved accuracy. In particular, I show that by incorporating unsupervised morphological inference in order to stem the foreign text, it is possible to improve the alignment quality. This is a novel use of unsupervised morphological inference which, to my knowledge, has never been applied to the word-alignment problem.

The idea of using existing tools for analyzing one language to help build tools for another language is not new. This technique, known as the "align and transfer" approach, was introduced by Yarowsky et al. (2001), and used to induce various tools for analyzing foreign text. The key to success in that project was not relying entirely on direct alignments, but instead using noise-robust methods for training the the resulting tools from inherently noisy data. Their experiments on English-French bitext showed that although hand-generated alignments led to better performance than statistically-induced alignments, even perfect alignments yielded only an 86% accuracy of projected part-of-speech tags, despite the relative typological similarity of English and French and the use of a two-million word corpus. I am aware of no previous work applying the align-and-transfer procedure to small amounts of parallel text to produce a richly-annotated corpus of text in an otherwise resource-poor language.

4.2 Stage 1: Parallel Text to Bitext

It is necessary to make a distinction between text which is aligned at a high level (e.g. a document and its translation), for which I use the term *parallel* text, and text aligned at a more fine-grained level, for which I use the term *bitext*. An individual bitext, as the term is used here, can range in size from a single word to one or two sentences, but typically not more than that. By explicitly aligning individual sentences from a parallel text across the two languages, it is possible to convert a large parallel text into a corpus of smaller bitexts.

Sentence alignment is an important, but oft-overlooked, stage in processing parallel text. Perhaps it is overlooked because some corpora are inherently aligned at the sentence level, and therefore have no need for automated sentence alignment. For example, corpora derived from parliamentary proceedings, such as the Hansards or Europarl corpora, usually have the contents neatly organized into paragraphs and subsections that are often as small as a single sentence. However, many other parallel text corpora, particularly ones assembled from resources not initially intended for machine-translation purposes, lack this type of structure. Often these corpora consist of documents that are translations of one another at the document level, which may range in size from a single news article to an entire novel. In these cases it is necessary to determine which sentences are in fact translations of each other before word-alignment tools like GIZA++ may be used.

The Gale-Church algorithm (Gale and Church, 1993) uses the length, in letters, of the sentences in both languages to determine how sentences are to be paired up, and allows for a sentence in one language to be aligned with zero, one, or more sentences in the other language. Perhaps due to its simplicity, speed, and good results, this remains the standard solution to the sentence alignment problem. However, more recent approaches have been explored that take into account additional types of information, such as HTML structure (Shi and Zhou, 2008), or bilingual dictionaries (Varga et al., 2005; Ma, 2006; Li et al., 2010). Lexical translation-based approaches that do not require dictionaries (Moore, 2002) have been described, but the statistical translation models employed there require larger data sets than are available for resource-poor languages.

For alignment, I choose to use the Hunalign[1] software (Varga et al., 2005), because it is freely available under a GPL license, and it combines the length-based approach of the Gale-Church algorithm with lexical alignments from an optional translation

[1] `http://mokk.bme.hu/resources/hunalign/`

dictionary. Although translation dictionaries are not generally available in the types of scenarios I am interested in, it is often possible to quickly generate by hand a small translation dictionary that covers a few high-frequency words as well as transparent translations such as numerals and proper nouns that occur on both sides of the corpus.

For sentence tokenization both of English and foreign text, I use the *Punkt* sentence tokenizer (Kiss and Strunk, 2006), which has the advantages of being unsupervised, language-independent, and freely-available through the Natural Language Toolkit (NLTK).[2] The Punkt tokenizer uses unsupervised learning methods to identify abbreviations, numerals, and initials in the text before determining sentence boundaries. The system is easy to use and has been shown to achieve very good results across a number of different languages and text genres.

For word tokenization of the foreign text, a simple method is used that tokenizes based on whitespace and non-alphabetic (e.g. punctuation) characters. For English, word tokenization is performed during the parsing stage (see below).

At the end of the tokenization and sentence alignment stage, the result is a collection of sentence-sized, tokenized bitexts. These tokenized bitexts are the basis for all further processing and will not change; all later processing simply enriches the existing bitexts by adding features (e.g. alignment information, part-of-speech tags, or other morphosyntactic features) to the individual tokens.

4.3 Stage 2: English Text Enrichment

Because no two languages encode exactly the same linguistic features in morphology, there is fundamentally a form mismatch between the two sides of any given bitext. The ideal of a complete morph-to-morph alignment is hardly ever attainable. In the case of English, the most common reference language in bitexts, a handful of prominent linguistic features are encoded in morphology (e.g. number on nouns,

[2] `http://nltk.org`

some sparse verb-subject agreement) but a great amount of grammatical information is found at the syntactic level. Much of the same information which is encoded syntactically in English is encoded morphologically in highly-inflecting languages.

Therefore, to make such features explicit, a two-pronged approach is used: first the sentence is parsed, to both a phrase-structure and a dependency tree representation; then the individual word tokens are analyzed for morphological structure. Since the morphological structure is often ambiguous (the inflectional suffix -s, for example, has many functions in English), the part-of-speech tags from the phrase-structure parse are used to select the best morphological analysis of a wordform in context. The results of parsing and morphological analysis are then distilled down to a set of salient features assigned to individual English word tokens.

4.3.1 Morphological Analysis of English Words

The English word tokens are analyzed morphologically by the PC-KIMMO (Karttunen, 1983) software package, using the English morphology that is included in that package. The morphological analyses are used to produce features for the English tokens that cannot be derived from the parse structure. For pronouns this includes number, person, and case features. For all other words, this means the stem and inflectional suffix that comprise the wordform.

Using PC-KIMMO in this way fails to take advantage of its full power, particularly its ability to analyze derivational morphology; however, these features are all that are required for the present study, whose focus is on inflectional morphology. The detail of analysis provided by PC-KIMMO can also be problematic for finding stems of words. For example, the verb *smelt* as a past tense for *smell* is analyzed by PC-KIMMO as monomorphemic, derived from the root *smell*. Thus, the stem of *smelt* is *smelt*, but its root is *smell*. Relying on the root feature is problematic, however, for any word analyzed as derivationally complex: for example, the verb *present* is analyzed as

deriving from the root *send*. Thus, I use PC-KIMMO's stem feature, stripping only inflectional morphology, and accept that this will not always give the result that is desired.

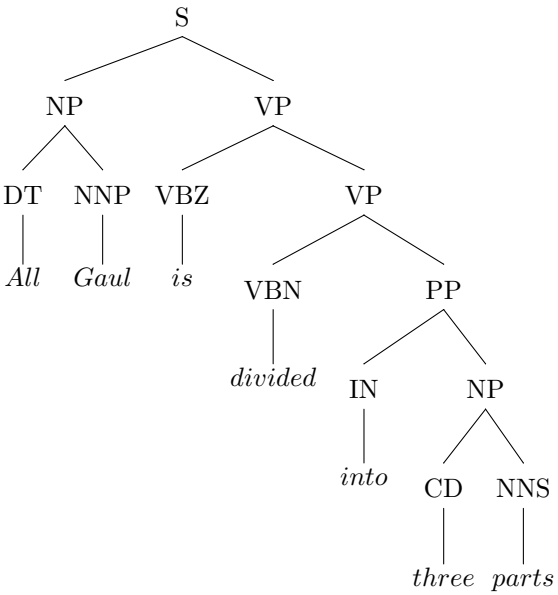
4.3.2 Syntactic Analysis of English Sentences

The English sentences are parsed using the Stanford Parser.[3] The Stanford Parser performs comparably to other benchmark parsers, namely the Charniak and Collins parsers (Klein and Manning, 2003a), and it is well-suited to this project due to the pre-trained English grammar included with the parser, as well as the output the parser produces. English parsers typically use grammars trained on parse trees from the Penn Treebank, which consists of newswire text from the Wall Street Journal; however, the English text used in these experiments is generally unlike newswire text. The Stanford Parser download includes a pre-trained probabilistic phrase-structure grammar of English based on an augmented data set which the authors claim is better suited for parsing non-news type sentences, including commands and questions (as stated in the software's documentation). Thus, the Stanford Parser may be applied to this data set without any need for additional training. Furthermore, the Stanford Parser produces both phrase-structure and dependency parses (de Marneffe et al., 2006), which allows us to easily combine both types of syntactic information without the need for a separate dependency parser. Illustrations of the phrase-structure and dependency tree outputs are given in figures 4.2 and 4.3 respectively.

4.3.3 Morphological Analysis of Foreign Words

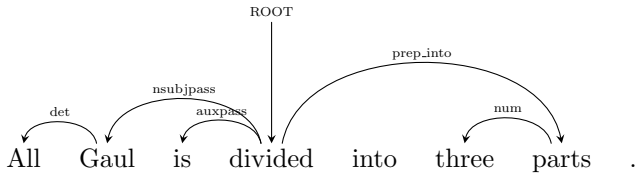
Prior to alignment, the foreign text is processed only for word and sentence tokenization. Additionally, in some experiments, the foreign wordforms undergo unsupervised morpheme segmentation, removing affixes prior to alignment in an effort to

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`




```
(S
  (NP (DT All) (NNP Gaul))
  (VP (VBZ is)
    (VP (VBN divided)
      (PP (IN into)
        (NP (CD three) (NNS parts)))))
  (. .))
```

Figure 4.2: Phrase-structure parse of the English text, graphically and in bracket notation.




```
det(Gaul-2, All-1)
nsubjpass(divided-4, Gaul-2)
auxpass(divided-4, is-3)
root(ROOT-0, divided-4)
num(parts-7, three-6)
prep_into(divided-4, parts-7)
```

Figure 4.3: Dependency parse of the English text, graphically and in notation.

improve alignment accuracy.

4.3.4 Summary

The morphological analysis and parse provide a large amount of detailed information about the English text. For the purposes of this dissertation, however, only a subset of this information is used. Each English token is enriched with the following features:

1. The stem of the word, as determined by PC-KIMMO.
2. The label of the dependency arc leading into the wordform (if any).
3. The base part-of-speech, as determined by the part-of-speech tag.
4. Additional features determined by the part-of-speech tag. For nouns, this is a number feature (singular or plural) and a proper-noun feature (proper or common). Other features (such as degree of adjectives or tense of verbs) may be extracted but do not play a role in the analyses presented in this dissertation.

4.4 Stage 3: Word Alignment and Feature Transfer

Once the English text has been analyzed and the English tokens have been annotated with additional features, these features are transferred to the foreign wordforms via statistical word alignment. Below I discuss the details of the process as used in these experiments.

4.4.1 Word Alignment

The de-facto standard for bitext alignment is GIZA++, a software implementation of IBM's translation models. For my experiments, I use an off-the-shelf GIZA++ aligner, although it is worth noting that this is not the only option and does not

necessarily represent the current state-of-the-art. Factored translation models provide better alignment for morphologically-complex languages, but they are not appropriate for widespread use because they require language-specific morphological analysis to be performed prior to alignment.

The alignment approach used is fairly simple. First, each token in the English text is replaced by its stem, discarding any inflectional morphology. If the foreign text has been analyzed, it can also be stemmed at this point. The goal of stemming is to reduce the low-frequency effects that result from using a small corpus: instead of the aligner treating forms like "house", "housed", or "housing" separately, their observations are lumped together. Then, the texts are aligned, and the inflectional morphology, as well as all the additional features that were produced during the text enrichment stage, are re-associated with the tokens.

4.4.2 Feature Transfer

After alignment is complete, the feature transfer process is straightforward; for each token on the English side of the alignment, all of its relevant features (i.e. the features that were discussed in section 4.3) are copied and assigned to each foreign token that has been aligned to it. (The alignment is done in such a way that each foreign word is aligned with at most one English word, although one English word may align with more than one foreign word.) Any unaligned tokens are assigned an "X" part-of-speech tag.

4.5 Performance & Evaluation

The align-and-transfer procedure described above results in a valuable product: a morphologically tagged version of the foreign-language corpus used as input. Thus, we are directly interested in procedure's accuracy when performing this task. Furthermore, the output of the align-and-transfer procedure is used as the basis for further

morphological processing: the segmentation and paradigm-building process discussed in chapter 5. Any errors introduced during the alignment stage will affect the results of that process, so clearly we would like to keep errors minimal.

The align-and-transfer process itself is the sum of various components, each of which can introduce its own amount of noise into the overall result. The quality of the data itself may contain errors or simply be ill-suited for the align-and-transfer task. Major areas where noise may be injected include:

- Errors in the English-side features (i.e. incorrect English morphological and syntactic analyses).
- Errors due to incorrect alignment of English and foreign words.
- Errors due to grammatical mismatch between English and the foreign language.

The first type of error is of least concern, as it lies outside the scope of this project. The only likely errors arising from the morphological analysis of English words are the results of ambiguous wordforms (e.g. the verb *runs* and the plural noun *runs*). These ambiguities are resolved by the part-of-speech tags produced by the parser, so any remaining errors are in fact parse errors, not morphological errors. English parsing is one of the most widely-studied areas of Natural Language Processing, and methods are continually being refined for better performance. While improving parsing accuracy may be possible (for instance, by using a bleeding-edge parser tuned specifically for the types of sentences found in this genre), the potential gains do not outweigh the advantages of using the off-the-shelf version of the Stanford Parser, and our time is better spent on other areas of the system that are the true focus of this work.

In the absence of hand-aligned bitext to use as a gold standard, it is difficult to evaluate the accuracy of the alignment process. However, if a bilingual dictionary is available, then it is possible to evaluate individual word-to-word alignments on

the basis of whether the dictionary considers those words to be translations of one another. (Of course, if the dictionary was used in the alignment process, then this is is less useful as an evaluation metric.)

In an effort to evaluate how well the align and transfer system described in this chapter performs, the following sections describe two experiments on an English-Estonian parallel corpus, using the transferred part of speech tags to estimate the alignment accuracy.

4.5.1 Experiment 1: Direct Transfer

The Estonian morphologically-disambiguated corpus gives a nice opportunity to evaluate the quality of the aligned features. Since each Estonian token has already been manually annotated for morphological analysis, we can use those tags to evaluate how well the transferred features match up with the true features.

The first experiment performs a standard GIZA++ alignment on the English and Estonian wordforms, then uses this alignment to project a set of twelve basic part-of-speech tags from English to Estonian. (The basic part-of-speech tags are noun, verb, adjective, etc.; more details of the basic part-of-speech tagset are given in Appendix E.) This is called the "direct transfer" approach by Yarowsky et al. (2001); in that study, the authors reported a 76% accuracy for part-of-speech tags projected from English to French on a corpus of 2 million words. Clearly, higher error rates can be expected for smaller corpora, and for languages that are typologically more dissimilar from English than French is.

This experiment was conducted using the parallel English-Estonian text of George Orwell's novel *1984*, which consists of 94k Estonian tokens. (Details of this corpus are given in Appendix D.) Overall, the direct transfer approach achieved a tag prediction accuracy of 53.4%. If punctuation tokens are ignored, accuracy falls to 45.2% Figure 4.4 shows the tag-prediction performance breakdown by tag.

The direct-transfer tag prediction results are far from optimal, but for each of the major word classes, the correct tag is still predicted more often than any other tag. There are two ways to improve the accuracy of the predicted features: the first is to improve the word alignment, and the second is to reduce noise in the foreign tag predictions. Yarowsky et al. (2001) managed a 9-point performance increase (from 76% to 85%) by using perfect alignments instead of statistically-induced alignments, demonstrating both that better alignments lead to better predictions, but also that there is a limit to the direct transfer approach even given perfect alignments. The remainder of this chapter deals with methods for improving alignment quality to the extent possible, although we acknowledge that the projected morphological features will inevitably contain a significant amount of noise, no matter how good the alignments are, and this must be dealt with by any downstream processing (e.g. morphological inference).

4.5.2 Experiment 2: Stemmed Alignment

Since I am using an off-the-shelf aligner, there is only so much that can be done to improve alignment: it is not within the scope of this project to refine the alignment algorithm itself. However, one possibility is to preprocess the bitext in a way that could help mitigate the effects of the low sample size. To accomplish this, we perform stemming on the wordforms in the corpus. In principle, this should encourage alignments between inflected forms of the same word: for example, if the English words *friend* and *friends* are both stemmed to *friend*, and if the Latin words *amicum*, *amicus*, *amicorum*, etc. are all stemmed to *amic*, then this will result in a single *friend*:*amic* alignment that occurs several times in the corpus, rather than several more specific alignments (*friend*:*amicum*, *friend*:*amicus*, *friends*:*amici*) that each may only occur once or twice in the corpus. Since a statistical alignment approach relies on recurring alignments in the text, stemming should have the effect of improving the alignment

accuracy.

Since English is not a highly-inflecting language, stemming of English is likely to have a fairly limited effect on alignment, although it should help at least a little bit if the corpus is small. When aligning English to a foreign language that uses a lot of morphology, however, the potential benefits of stemming become more significant. While research on machine translation in highly-inflecting languages often relies on analyzing morphologically-complex words, this type of approach requires software to perform the morphological analysis; see, for example, recent work on Greek and Czech (Avramidis and Koehn, 2008) and Turkish (Durgar-El-Kahlout and Oflazer, 2006; Yeniterzi and Oflazer, 2010).

Furthermore, statistical machine-translation research generally relies on very large corpora for which data sparseness is less of an issue. For example, Avramidis and Koehn (2008) improve translation from English into Greek and Czech by adding *more* morphological information to the English tokens; this is essentially the opposite of the approach I am taking.

To see if stemming the wordforms could improve the alignment quality, I performed experiments: in the first experiment, the English words were stemmed based on the morphological analysis produced by PC-KIMMO. Since PC-KIMMO produces very detailed morphological analyses, I only removed the ultimate inflectional morpheme of a word, to produce the stem. (This avoids, for example, yielding *organ* as the root of the verb *organizing*.) In the second experiment, English words are stemmed the same way, but foreign words are also stemmed on the basis of an unsupervised morphological analysis. The morphological analysis is produced by running Morfessor (version 1.0) with its default settings on the foreign text, then taking the first (leftmost) morpheme of each token as its stem. The stemmed forms of both English and foreign text are used to produce the alignment, but the full forms of the tokens are used in all other processing stages.

Baseline	19.1%
No stemming	54.3%
English stemming only	54.5%
Both stemming	58.4%

Table 4.1: Part-of-speech prediction accuracy, measured as the percentage of foreign tokens which were assigned the correct base POS tag from the universal tag set.

The alignments produced in these two experiments, plus the direct-transfer experiment from the previous section, yielded the overall base part-of-speech prediction accuracy listed in table 4.1. The overall results for the three cases are given, as well as a simple baseline for comparison. The baseline predicts that each token is a noun, since noun is the most common POS tag in the English text; this is a slightly better baseline than randomly choosing one of the twelve POS tags.

Figure 4.5 shows the detailed results from the first experiment, using only English stemming. This has little overall effect on the accuracy of the predicted part-of-speech labels (53.5%) compared to the non-stemmed, direct-transfer alignment (54.3%: see figure 4.4). While most categories were predicted more accurately after stemming, there was a drop in the accuracy of verbs that offset those improvements. It is interesting that the verb alignment scores decreased after stemming the English tokens, and I can think of no good explanation for why this occurs.

Figure 4.6 shows the detailed results of the second experiment, in which both the English and the Estonian wordforms are stemmed. This results in a more significant improvement, up to 58.4% overall accuracy. While this is still a fairly low level of accuracy, it is clearly an improvement over the unstemmed alignments, demonstrating that it is possible to use unsupervised morphological inference to improve word alignment when dealing with morphologically complex languages.

		True POS tags											
		ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	PUNCT	VERB	X
Predicted POS tags	ADJ	**2457**	219	1239	216	0	2172	158	1113	3	93	1771	13
	ADP	7	**359**	231	1468	0	47	4	118	0	46	98	0
	ADV	187	122	**2909**	406	0	501	27	302	7	78	1347	0
	CONJ	0	0	11	**2592**	0	2	0	1	0	1	4	0
	DET	14	1	75	43	**0**	4	26	767	1	89	214	0
	NOUN	1947	680	2489	490	0	**12000**	319	2753	22	402	4284	49
	NUM	23	9	26	7	0	74	**426**	32	0	14	37	1
	PRON	2	3	16	4	0	265	2	**4748**	0	610	15	0
	PRT	0	6	128	165	0	0	0	1	**0**	72	16	0
	PUNCT	0	1	15	251	0	4	0	26	0	**16563**	80	0
	VERB	900	396	2253	408	0	3248	152	2220	9	386	**9459**	11
	X	318	96	507	334	0	1002	51	511	8	1124	867	**5**
	Precision	26.0	15.1	49.4	99.3	0.0	47.2	65.6	83.8	0.0	97.8	48.7	0.1
	Recall	42.0	19.0	29.4	40.6	0.0	62.1	36.6	37.7	0.0	85.0	52.0	6.3
	F1	32.1	16.8	36.9	57.6	0.0	53.6	47.0	52.0	0.0	91.0	50.3	0.2
54.3% overall accuracy (51518 / 94905)													

Figure 4.4: Part-of-speech prediction performance on the Estonian corpus, using direct transfer.

		True POS tags											
		ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	PUNCT	VERB	X
Predicted POS tags	ADJ	**2606**	249	1360	224	0	2615	174	1147	3	102	1952	15
	ADP	11	**382**	252	1459	0	70	5	176	0	57	112	0
	ADV	230	128	**3169**	417	0	638	39	361	7	62	1520	0
	CONJ	1	0	12	**2578**	0	4	0	1	1	2	4	0
	DET	21	3	98	52	**0**	10	32	870	1	88	253	0
	NOUN	1952	704	2485	467	0	**12171**	313	2747	23	397	4449	49
	NUM	25	11	50	9	0	84	**445**	52	0	18	58	1
	PRON	2	7	38	5	0	284	2	**5016**	0	633	27	0
	PRT	0	5	147	164	0	3	1	9	**0**	74	25	0
	PUNCT	1	1	21	390	0	9	0	59	0	**16636**	190	0
	VERB	687	306	1761	285	0	2433	103	1643	7	289	**8738**	9
	X	319	96	506	334	0	998	51	511	8	1120	864	**5**
	Precision	24.9	15.1	48.2	99.0	0.0	47.3	59.1	83.4	0.0	96.1	53.7	0.1
	Recall	44.5	20.2	32.0	40.4	0.0	63.0	38.2	39.8	0.0	85.4	48.0	6.3
	F1	32.0	17.3	38.5	57.4	0.0	54.0	46.4	53.9	0.0	90.4	50.7	0.2
54.5% overall accuracy (51746 / 94905)													

Figure 4.5: POS-prediction performance, using stemmed English tokens.

		True POS tags											
		ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	PUNCT	VERB	X
Predicted POS tags	ADJ	**3079**	240	1302	182	0	1925	176	1016	3	90	1643	17
	ADP	15	**429**	294	1522	0	76	6	195	0	46	138	0
	ADV	203	139	**3394**	415	0	514	37	351	7	71	1483	0
	CONJ	1	0	13	**2613**	0	7	0	1	1	4	9	0
	DET	32	3	144	47	**0**	11	40	1120	1	91	252	0
	NOUN	1607	669	2173	413	0	**13505**	286	2321	21	336	3756	49
	NUM	25	11	43	6	0	77	**497**	57	0	7	37	1
	PRON	9	8	79	7	0	326	2	**5442**	0	601	37	0
	PRT	2	5	169	180	0	5	0	15	**0**	70	24	0
	PUNCT	2	1	22	390	0	22	1	114	2	**16797**	259	0
	VERB	561	291	1761	274	0	1849	69	1452	8	251	**9685**	7
	X	319	96	505	335	0	1002	51	508	7	1114	869	**5**
	Precision	31.8	15.8	51.3	98.6	0.0	53.7	65.3	83.6	0.0	95.4	59.8	0.1
	Recall	52.6	22.7	34.3	40.9	0.0	69.9	42.7	43.2	0.0	86.2	53.2	6.3
	F1	39.7	18.6	41.1	57.9	0.0	60.8	51.6	57.0	0.0	90.6	56.3	0.2
58.4% overall accuracy (55446 / 94905)													

Figure 4.6: POS-prediction performance, using stemmed English and stemmed Estonian.

4.6 Conclusion

In this chapter I showed how various off-the-shelf NLP software packages can be combined to produce a system to enrich standard bitext by transferring morphosyntactic features from the English text to the foreign text. Although the align-and-transfer methodology is an established one, and all of the components are readily available NLP tools, constructing a system to automate this entire process is not trivial and requires extensive scripting to connect the different components.

While the accuracy of the predicted part-of-speech labels is less than perfect, it significantly outperforms a baseline classifier. The performance is improved by stemming both the English text (using an English stemmer) and foreign text (using unsupervised morphology induction) prior to alignment.

There are two ways to improve the accuracy of the predicted labels beyond the figures achieved here. The first is to improve the word alignments, and the second is to re-classify the foreign forms. Either of these tasks could potentially be aided by the results of a morphological analysis, such as is described in chapter 5. It would also be fairly easy to assign a confidence score to the predictions (for example, based on the sentence alignment scores), and use this to extract a subset of the data that is more likely to be correctly tagged. For some applications, the decrease in size of the data set may be made up for by the increase in quality.

To my knowledge, no previous research has used unsupervised morphology for the purpose of improving word alignments. Since the only difference in the three Estonian experiments was the alignment itself, POS-prediction can be used as a proxy measure of the alignment quality. The improvement gained by stemming the Estonian text based on the Morfessor output suggests that this is a technique that could be useful in the domain of machine translation, if the target language is one with rich morphology but for which morphological software is unavailable.

CHAPTER 5

Paradigmatic Morphology Induction

In this chapter I describe a paradigm-based approach to morphology induction based on biform data. In its broadest sense, morphology induction is the process of starting with observed data and inferring from it a morphological grammar—a representation, of some type, of the underlying morphological processes that generate that data. In my approach, the observed data consist of biform word tokens (tokens consisting of both a form component and a meaning component, such as those produced by the procedure described in the previous chapter), and the morphological grammar consists of a set of suffixing paradigms (tables that pair prefix stems with appropriate inflectional suffixes).

First, a note on terminology. Linguistic utterances, be they sentences, phrases, or individual morphemes, are often conceptualized as having a form component and a meaning component. In the present work, the fundamental data item is what I refer to as a *biform* token: a token with distinct form and meaning components.[1] I use the word *form*, in its normal linguistic usage, to refer to the form component of a biform token: this is a string of characters typically representing its orthographic form (its spelling, in a given alphabet), but it could just as well be a phonological or phonetic

[1]This same concept is referred to as a "grammatical word" by Katamba and Stonham (2006), but this term does not seem to be in wide use, and it is both unwieldy and ambiguous, being used in other contexts as a synonym for "function word." The term *biform* is meant to be succinct and to evoke the term *bitext*, the ultimate source from which the biform tokens are produced.

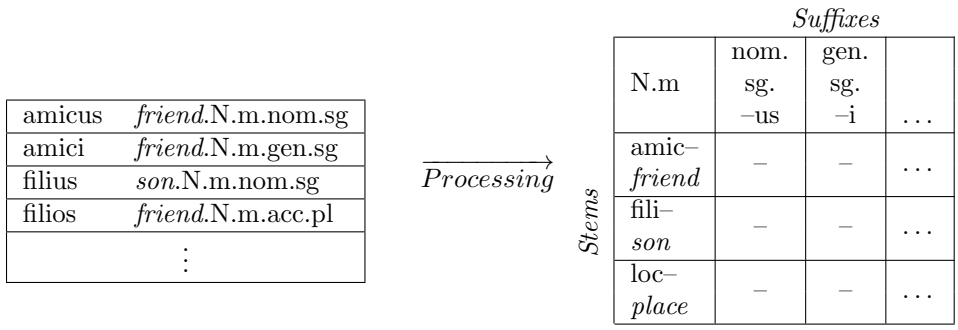


Figure 5.1: The high-level objective of paradigm induction. The input is bitext word tokens, and the output is a set of paradigms (only one is shown).

Biform	
Form	Features
amo	lemma:"love" tense:present voice:active number:singular person:first

Figure 5.2: Illustration of the components of a biform token representing the Latin word *amo*.

form, depending on the source of the data.[2] I use the term *features* to refer to the meaning component of a biform, since it is composed of a set of features. Again, the exact nature of the features will depend on the data source, but features are usually represented as attribute-value pairs and will typically represent grammatical features (e.g. tense, number, person, etc.) and include a gloss feature and/or a lemma feature which can be used, with some level of accuracy, to identify biforms belonging to the same lexeme. Figure 5.2 illustrates a typical biform token, showing the contents of its form component and its feature component.

I begin this chapter by framing my work in the context of previous work on mor-

[2]This is simply a matter of utilizing existing resources. I am in no way claiming that morphological processes truly operate on orthographic forms. The goal of the inference procedure is to produce the best analysis given the data available. Thus, even if the data is less than ideal, we still hope to arrive at an analysis that is useful and descriptive, even if it is not necessarily a "true" analysis in the sense that it accurately models the underlying linguistic processes.

phological induction, and I discuss the minimum description length (MDL) principle, which has played a significant role in recent approaches to unsupervised morphological induction. After describing an iterative clustering algorithm for paradigm induction, I present the results of some experiments showing the effectiveness of that algorithm.

5.1 Minimum Description Length

Several current approaches to unsupervised morphology induction, including John Goldsmith's *Linguistica* system (Goldsmith, 2001) and the early versions of *Morfessor* (Creutz and Lagus, 2005) are based on the principle of MDL (Rissanen, 1989; Grünwald, 2005). MDL, which draws on concepts from information theory, is a measure of compression; in MDL the aim is to take a data set and represent it in a very efficient, compact way. As it relates to morphological induction, this compression of the data involves finding and exploiting regularities in the data (i.e. recurring morphemes). Below, I briefly describe some of the underlying principles of MDL and give a definition of MDL. I will proceed to introduce a MDL model of biform morphology, and then show how this model can be used in a search algorithm to infer the the morphological structure of a language, using bitext as its input.

5.1.1 Information Theory and Code Length

MDL comes from the field of information theory and has to do with codes and code lengths. In information theory, a code is a system for representing a message (say, a series of words) as a sequence of bits (ones and zeros). For example, consider an alphabet consisting of three letters a, b, and c, and assume that the messages we want to encode are sequences of these letters. (The letters used in this example could just as easily be replaced by any other symbols, for example, whole words.) If we adopt a code C such that $C(a) = 0$, $C(b) = 10$, and $C(c) = 11$, then we can unambiguously encode and decode any sequence of the letters a, b, and c using this

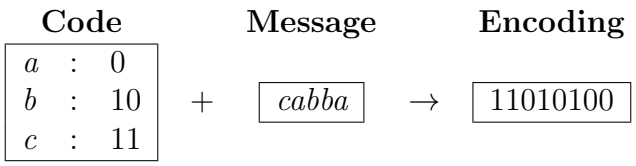


Figure 5.3: Example of encoding and decoding a message with a prefix code.

code. For example, given the message "cabba", the encoding of this message would be "11010100". In this example, given that same encoded bit string, "11010100", we can decode it back to the original message "cabba," simply by looking at the inverse code function, where $C^{-1}(11) = c$, $C^{-1}(10) = b$, and $C^{-1}(0) = a$.

Note that this is a *lossless* code: the message is perfectly preserved after encoding and decoding. Not all codes are lossless: imagine the same scenario, only using a code where $C(a) = 1$, $C(b) = 11$, and $C(c) = 111$. In this case, the encoding is simple enough; we can encode the same message "cabba", resulting in the encoded bit string "111111111". However, when we try to decode that same string, we find that there are many possible ways to decode it, and we are not guaranteed to get out the same the message we put in. For any alphabet, it is always possible to design a type of lossless code known as a *prefix code*. A prefix code is one in which no code word is the prefix of any other code word. The first example given above is an example of a prefix code.

In these two examples, we use code words of varying length (e.g. the code word for the letter a has one bit, whereas the code word for b has two bits). While it is always possible to design a code for an alphabet containing n letters where each code word consists of $\log(n)$ bits, it is often advantageous to use code words of varying length. For example, if a is a very frequent symbol, and b is an infrequent symbol, then it makes sense to use fewer bits to encode a than to encode b. If we assume a probability distribution over symbols, then it can be shown that the optimal code length for a symbol x is the negative log probability of x (Rissanen, 1989, p. 25).

$$\lambda^*(x) = -\log_2 p(x) \tag{5.1}$$

I use λ to represent the function that gives the code length in bits of a symbol. While the optimal code length is not always achievable in practice, if the messages being encoded are arbitrarily long and that the probability distribution from which the letters of the alphabet are sampled is known, then it is always possible to design a prefix code that approaches this optimum (Grünwald, 2005). (This is partly accomplished by assigning codes to sequences of symbols, resulting in an average code length that is not an integer number of bits.) MDL traditionally assumes arbitrarily-long messages, and thus uses the ideal code length function.

5.1.2 MDL Definition

While it is helpful to understand the basics of codes, MDL is only concerned with the lengths of encoded messages, not the actual encodings themselves. Furthermore, in an MDL scenario, the message is not encoded directly: instead, the message is first *described* in terms of a model, and the resulting model description is *encoded*, for example using prefix codes. For someone to be able to decode the message, we must send them both the model M that was used as well as the message (also called the data sequence) D, described in terms of the model. Thus the encoded length of the data sequence is the sum of the length of the encoded model plus the encoded length of the data sequence described using the model:

$$\lambda(D) = \lambda(M) + \lambda(D|M) \tag{5.2}$$

In an MDL learning scenario, the goal is to minimize $\lambda(D)$, the description length of the data (hence the name, minimum description length). This is accomplished by considering different models $M \in \mathcal{M}$, describing the data D using the model M, and

evaluating the value for $\lambda(D)$ under that model. The optimal model M^* is the model which yields the smallest description length:

$$M^*_{MDL} = \underset{M}{\operatorname{argmin}}\, \lambda(M) + \lambda(D|M) \tag{5.3}$$

Below, I will discuss how the MDL criterion can be applied as an objective function in a search over a set of competing morphological models.

5.1.3 Compression as Learning

MDL is a compression technique; it seeks to encode the given data using as few bits as possible. Data compression is widely used in computing and is essential for modern applications such as streaming music and video. However, compression is also a way of thinking about the complexity of a data sequence. The Kolmogorov complexity of a data sequence is defined as the shortest computer program that prints that sequence. It has been shown that Kolmogorov complexity is independent of the programming language used, but Kolmogorov complexity is also incomputable; there is no programmatic way to calculate it for an arbitrary input sequence (Grünwald, 2005).

Random sequences will have a Kolmogorov complexity roughly equal to the length of the sequence, while sequences containing repeated patterns or internal structure will have shorter Kolmogorov complexity. Since language is a system with regular patterns, we can expect that sequences representing natural language utterances will have a lower complexity than random sequences. While Kolmogorov complexity is incomputable, MDL provides a framework to find a model of the data that approximates the Kolmogorov program, capturing the regularities that are contained in the data sequence. In the sections below, I describe a search procedure that generates a series of candidate models in search of the MDL-optimal model. By evaluating competing models of the data using the MDL minimization equation (5.3), we aim

to discover the model that best captures the regularities within the data.

Still, it is important to keep in mind that we are not really interested in compressing the data. Rather, we are interested in discovering the true morphological structure of the language; i.e. the analysis that linguists would find most satisfactory. We keep MDL in check by imposing restrictions on the structure of the encoding algorithm. This structure may prevent us from finding the optimal compression of the data, but it helps us find a better morphology of the language. MDL, or any other objective function, is simply a useful metric for guiding our search.

5.2 An MDL Model of Biform Morphology

This model draws heavily from Goldsmith's model, but with important modifications to accommodate *biform* morphemes; Goldsmith's model, being an unsupervised method, represents morphemes as strings, essentially the form half of the bitext tokens described here. In my model, the form is represented as a string of letters drawn from an alphabet of letters $\mathcal{L}$, and the meaning is represented as a set of features drawn from an alphabet of features $\mathcal{F}$. (A summary of the notation and the data objects described in this section can be found in figure 5.4.)

The model is based on a data sequence D, which is a corpus of biform word tokens. Every data item d in D consists of a string L_d consisting of letters drawn from the letter alphabet $\mathcal{L}$, and a set of features F_d drawn from the feature alphabet $\mathcal{F}$. This is the same structure used for individual stems s_i, and affixes a_i, within the morphology.

The model of the morphology itself, M, consists of a list of paradigms. A given paradigm p consists of a list of stems S_p, a list of affixes A_p, and a set of features shared by all stems in that paradigm, F_p.

Given a complete model, every data item in the corpus can be represented as a triple (i, j, k), where i is a pointer to a paradigm $p \in M$, j is a pointer to a stem within the paradigm $s \in S_p$, and k is a pointer to an affix within the paradigm $a \in A_p$.

Symbol		Denotes
D	=	the data sequence (the corpus, aka the message), consisting of a sequence of data items.
	=	$[d_1 d_2 d_3 ...]$
M	=	the model (morphology), consisting of a list of paradigms
	=	$[p_1 p_2 p_3 ...]$
p	=	a paradigm, consisting of a list of stems, a list of affixes, and a set of features
	=	(S_p, A_p, F_p)
S	=	a list of stems $[s_1 s_2 ...]$
A	=	a list of affixes $[a_1 a_2 ...]$
F	=	a set of features $(f_1, f_2, ...)$, $f_i \in \mathcal{F}$
L	=	a string of letters $[l_1 l_2 ...]$, $l_i \in \mathcal{L}$
d	=	a biform data item, consisting of a string of letters and a set of features.
	=	(L_d, F_d)
$\mathcal{F}$	=	the alphabet of all features
$\mathcal{L}$	=	the alphabet of all letters

Figure 5.4: Notation of data objects.

5.2.1 The MDL Objective Function

We use the MDL minimizing function $\lambda(M) + \lambda(D|M)$ as the objective function guiding our search. Recall equation 5.3:

$$M^*_{MDL} = \underset{M \in \mathcal{M}}{\operatorname{argmin}}\, \lambda(M) + \lambda(D|M) \tag{5.3}$$

Here $\mathcal{M}$ is the model space, which is the space we will need to search through to find the optimal model. Since the objective function is a minimizing function, we do not need to fully calculate the description length: we only need to calculate relative differences in the description lengths of competing models. Thus, any quantity that is shared in the lambda function for all models can be safely ignored.

Now, let us spell out the objective function in more detail, addressing the two terms (the model length and the data length) separately. The notation λ is used generically to reference the function that returns the description length of its argu-

$\lambda_X(x)$	=	description length of a pointer to an item x in a list X
λ_N	=	description length of an arbitrary positive integer
λ_d	=	description length of a data item, stem, or affix
λ_f	=	description length of a feature in $\mathcal{F}$
λ_l	=	description length of a single letter in $\mathcal{L}$
λ_p	=	description length of a paradigm

Table 5.1: Definitions of description length functions.

ments, but we will introduce several specific λ functions for various data types: these are summarized in table 5.1

Since the model M is simply a list of paradigms, the model description length is the number of bits necessary to encode a list of $|M|$ paradigms. To encode this list, we first encode the length of the list, then encode each member of the list.

$$\lambda(M) = \lambda_N(|M|) + \sum_{p \in M} \lambda_p(p) \tag{5.4}$$

A paradigm consists of 1) a set of stems, 2) a set of affixes, and 3) a set of features:

$$\begin{aligned} \lambda_p(p) \quad = \quad & \lambda_N(|S_p|) + \sum_{s \in S_p} \lambda_d(s) + \\ & \lambda_N(|A_p|) + \sum_{a \in A_p} \lambda_d(a) + \\ & \lambda_N(|F_p|) + \sum_{f \in F_p} \lambda_f(f) \end{aligned} \tag{5.5}$$

Before continuing, let us spell out the definition of λ_d, the description length of a data item, which is the same as the description length for a stem or affix. Any one of these items consists of a set of letters (the string form) and a set of features. We avoid the need to specify the length of the sequence or set by assuming that a stop character is included in the alphabet. It would be possible to use frequency estimates to produce an optimal code for each item in the alphabet.[3] However, for simplicity

I simply treat each letter and each feature as having a uniform cost c.

$$\lambda_d(d) = \lambda_l(L_d) + \lambda_f(F_d) \quad (5.6)$$

$$= c\,(|L_d| + |F_d|) \quad (5.7)$$

Without using the ideal code length function, the code length of the encoding for letter or feature would depend on the size of the alphabet, so that $c = \log_2(|\mathcal{F}| + |\mathcal{L}|)$. However, even this calculation is unnecessary in practice (since we are only interested in the relative description lengths of competing hypotheses, not the absolute description lengths). Therefore I arbitrarily set the alphabet item cost $c = 1$.

Having spelled out how to calculate $\lambda(M)$, the next step for calculating the MDL cost is to describe the data sequence given the model and calculate its length $\lambda(D|M)$. As stated, each data item d can be encoded by a specification of a paradigm p_d, a stem s_d within that paradigm, and an affix a_d within that paradigm, and this can

[3]In this case, the length function becomes:

$$\begin{aligned}\lambda_d(d) &= \lambda_l(L_d) + \lambda_f(F_d) \\ &= \sum_{l \in L_d} -\log_2 p(l) + \sum_{f \in F_d} -\log_2 p(f)\end{aligned}$$

The description length function for a string of letters, λ_l, and the description length function for a set of features, λ_f, would determined by the cost of encoding members of the respective alphabets ($\mathcal{L}$ and $\mathcal{F}$), and thus are given by the ideal code length equation (5.1).

$$\begin{aligned}\lambda_l(L) &= \sum_{l \in L} -\log_2 p(l) \\ \lambda_f(F) &= \sum_{f \in F} -\log_2 p(f)\end{aligned}$$

The probabilities should be interpreted as maximum-likelihood estimates based on the occurrences of the letters and features in the corpus:

$$\begin{aligned}p(l) &= \frac{\text{count}_D(l)}{\sum_{l_i \in D} \text{count}_D(l_i)} \\ p(f) &= \frac{\text{count}_D(f)}{\sum_{f_i \in D} \text{count}_D(f_i)}\end{aligned}$$

be represented as a triple of pointers (i, j, k). While it is possible to spell out the description length for the data sequence,[4] at this point I question whether this is actually necessary for the present purposes.

The MDL principle is based on minimizing the cost of encoding both the model and the data, but for a linguistic analysis, the cost of encoding the corpus is not the primary concern. Rather, the concern is finding the best model of the morphology of the language. If we were to include the cost of encoding the corpus, this could potentially place a high value on a model that accurately encodes high-frequency wordforms at the cost of low-frequency forms, and high-frequency items (such as the words for "be", "do", or "say") often follow irregular patterns in a language. Therefore, for this algorithm I only use the $\lambda(M)$ term and ignore the $\lambda(D|M)$ term in the objective function. With this choice, I am focusing on the properties of the morphology itself, and not the properties of the corpus.

At this point, the objective function cannot accurately be called an MDL function, but it is an MDL-inspired cost function. Simplifying some of the formal aspects of MDL, the cost function measures the total cost of all the paradigms defined in the

[4]Each pointer can be represented using a prefix code for the members of the relevant set, thus:

$$\begin{aligned} \lambda(i) &= -\log_2 p(p_d) \\ \lambda(j) &= -\log_2 p(s_d|p_d) \\ \lambda(k) &= -\log_2 p(a_d|p_d) \end{aligned}$$

Then the length of encoding the entire corpus of pointers is:

$$\begin{aligned} \lambda(D|M) &= -\log_2(|D|) \sum_{d \in D} \lambda_d(d) \\ &= -\log_2(|D|) \sum_{d \in D} \lambda(i_d) + \lambda(j_d) + \lambda(k_d) \\ &= -\log_2(|D|) \sum_{d \in D} -\log_2 p(p_d) - \log_2 p(s_d|p_d) + -\log_2 p(a_d|p_d) \end{aligned}$$

Here, the probabilities should be thought of as the maximum-likelihood estimates based on the corpus: $p(p)$ is the relative frequency of tokens in the corpus analyzed as coming from paradigm p, $p(s|p)$ is the relative frequency of the stem s among tokens analyzed as coming from paradigm p, and $p(a|p)$ is the relative frequency of the affix a among tokens analyzed as coming from paradigm p.

model, where the cost of a paradigm is the cost of its paradigm features, stems, and suffixes, using the data-item cost function from equation 5.6.

$$\begin{aligned} M^* &= \operatorname*{argmin}_M \lambda(M) \\ &= \operatorname*{argmin}_M \sum_{P \in M} \left(|F_p| + \sum_{s \in S_p} \left(|L_s| + |F_s| \right) + \sum_{a \in A_p} \left(|L_a| + |F_a| \right) \right) \end{aligned} \tag{5.8}$$

This easy-to-calculate cost function forms the basis a search for the optimal model. The search procedure is described in the following section.

5.3 Paradigm Induction via Search

The model described in the previous section only provides a way to compute the cost of a hypothesis; it does not provide a method for finding the best hypothesis. In order to arrive at a satisfactory hypothesis, it is necessary to perform a search through the hypothesis space, considering several alternative hypotheses before ultimately selecting the one that yields the minimum value for our cost function.

Here, I describe a rather straightforward, greedy approach that iteratively builds clusters by merging groups of biforms, then infers a paradigm for each cluster. The search is guided by evaluating the cost of the hypothesis at each stage. Certainly, there are many other ways to search for hypotheses, and this method could be refined and optimized, but for now I leave any refinements and optimizations for future work.

5.3.1 Iterative Clustering Procedure

In this clustering approach, each biform in the dataset is assigned to a cluster, and for each cluster a paradigm is induced. The paradigm should include as many biforms as possible from the cluster, but not necessarily all of them. The remaining biforms are left unanalyzed; in this way the search accommodates some amount of

```
P ← {Paradigm(f) : f ∈ F}
P* ← P
while True do
  for all p1, p2 ∈ P × P do
    P' ← P − p1 − p2 + merge(p1, p2)
    if cost(P') < cost(P*) then
      P* = P'
    end if
  end for
  if cost(P*) < cost(P) then
    P ← P*
    continue
  else
    break
  end if
end while
```

Figure 5.5: Pseudocode of the biform clustering algorithm

noise in the input data. This set of paradigms and unanalyzed forms yields a model which can be scored according to the model given in section 5.2.

The search begins by assigning each form to a unique cluster, then proceeds by considering all possible merges of clusters, choosing at each iteration the merge that produces the largest decrease in the model cost. When no further merges yield a cost decrease, the search halts. A pseudocode description of the search is given in figure 5.5.

The clustering procedure forms the "outer loop" of the search process. Each iteration of re-clustering, however, requires an "inner" process to be performed which evaluates the relative cost benefits of re-clustering. This is accomplished by inferring a new paradigm for the new cluster, via the procedure described in the following section.

5.3.2 Inferring Paradigms from Clusters

As mentioned, a cluster is distinct from a paradigm, but clusters are built in such a way that the majority of forms in a cluster should belong to a paradigm. Therefore,

when learning a paradigm for a given cluster, we make the assumption that all of the forms belong together in one paradigm, and we can use deterministic methods that would be inappropriate for a inference on a larger, more diverse, dataset.

A paradigm, as defined in section 5.1, consists of a set of stems, a set of affixes, and a set of paradigmatic features. The cross-product of stems and affixes generates all of the possible forms in the paradigm, although not all of these forms are necessarily attested in the training data set. To infer a paradigm from a cluster of biforms, the biforms are arranged into an array such that each row in the array contains all of the forms associated with a given stem, and each column in the array contains all of the forms associated with a given affix. Any features shared by all forms are removed and assigned to the set of paradigm features. Then, each stem is inferred by removing all features shared by all forms in a single row, and removing the longest shared prefix on all forms in that row. Next, the affixes are inferred for each column as by removing the largest set of features features and the longest suffix shared by all forms in that column. The procedure by which a paradigm is inferred from a cluster of wordforms is explained in more detail with an example in the following section.

5.3.3 A Latin Example

The induction procedure is best illustrated with an example. Consider the three Latin first-declension nouns *amicus* "friend", *locus* "place", and *filius* "son". To simplify and save space, consider only a subset of the forms (the singular and plural forms of the nominative and accusative cases) of these nouns. These forms can be organized into a paradigm like so:

	Sg.	Pl.
Nom.	*amicus, locus, filius*	*amici, loci, filii*
Acc.	*amicum, locum, filium*	*amicos, locos, filios*

Now, consider the biform tokens corresponding to these twelve wordforms, and assume that they have all been clustered together into a single cluster c. Each biform consists of the form itself and its meaning features, so the cluster would look like this:

$$c = \left[\begin{array}{cccccc}
\textit{amicus} & \text{GLOSS:“friend”} & \text{POS:noun} & \text{NUM:sg.} & \text{CASE:nom.} & \text{GENDER:m} \\
\textit{amicum} & \text{GLOSS:“friend”} & \text{POS:noun} & \text{NUM:sg.} & \text{CASE:acc.} & \text{GENDER:m} \\
\textit{amici} & \text{GLOSS:“friend”} & \text{POS:noun} & \text{NUM:pl.} & \text{CASE:nom.} & \text{GENDER:m} \\
\textit{amicos} & \text{GLOSS:“friend”} & \text{POS:noun} & \text{NUM:pl.} & \text{CASE:acc.} & \text{GENDER:m} \\
\textit{filius} & \text{GLOSS:“son”} & \text{POS:noun} & \text{NUM:sg.} & \text{CASE:nom.} & \text{GENDER:m} \\
\textit{filios} & \text{GLOSS:“son”} & \text{POS:noun} & \text{NUM:pl.} & \text{CASE:acc.} & \text{GENDER:m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}\right]$$

Now, given this cluster the next step is to infer the appropriate paradigm for these forms. We notice that all of the forms are masculine nouns; they share the features POS:noun and GENDER:m. Thus, these two features are assigned as the paradigm features. Each unique gloss is assumed to represent a different lexeme, so forms are organized into rows according to their gloss.

$f_p = \left[\begin{array}{c}\text{POS:noun}\\ \text{GENDER:m}\end{array}\right]$				
$\left[\ \text{GLOSS:“friend”}\ \right]$	*amicus* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:sg}\end{array}\right]$	*amici* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:pl}\end{array}\right]$	*amicum* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:sg}\end{array}\right]$	*amicos* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:pl}\end{array}\right]$
$\left[\ \text{GLOSS:“place”}\ \right]$	*locus* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:sg}\end{array}\right]$	*loci* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:pl}\end{array}\right]$	*locum* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:sg}\end{array}\right]$	*locos* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:pl}\end{array}\right]$
$\left[\ \text{GLOSS:“son”}\ \right]$	*filius* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:sg}\end{array}\right]$	*filii* $\left[\begin{array}{c}\text{CASE:nom}\\ \text{NUM:pl}\end{array}\right]$	*filium* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:sg}\end{array}\right]$	*filios* $\left[\begin{array}{c}\text{CASE:acc}\\ \text{NUM:pl}\end{array}\right]$

At this point, the forms are organized into rows, but the forms within a row have not yet been organized into columns. After the paradigm features have been removed,

and the row features have been removed, any remaining features are assumed to be inflectional features and define an inflectional "slot" in the paradigm. These feature sets are assembled and a column is created for each set. The features then belong to the column, and can be removed from the individual cells. The result is a table with features specified for the entire paradigm, for each row in the paradigm, and for each column in the paradigm. The contents of the cells are the wordforms:

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	$\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$	$\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:pl} \end{bmatrix}$	$\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:sg} \end{bmatrix}$	$\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:pl} \end{bmatrix}$
$\begin{bmatrix} \text{GLOSS:"friend"} \end{bmatrix}$	*amicus*	*amici*	*amicum*	*amicos*
$\begin{bmatrix} \text{GLOSS:"place"} \end{bmatrix}$	*locus*	*loci*	*locum*	*locos*
$\begin{bmatrix} \text{GLOSS:"son"} \end{bmatrix}$	*filius*	*filii*	*filium*	*filios*

Arranged this way, it is evident that all the forms in a given column share a common suffix, and all forms in a given row share a common prefix. The next step in the paradigm inference procedure is to extract longest suffix shared by the forms in each column and assign this material to the paradigm suffixes:

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	*us* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$	*i* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:pl} \end{bmatrix}$	*um* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:sg} \end{bmatrix}$	*os* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:pl} \end{bmatrix}$
$\begin{bmatrix} \text{GLOSS:"friend"} \end{bmatrix}$	*amic*	*amic*	*amic*	*amic*
$\begin{bmatrix} \text{GLOSS:"place"} \end{bmatrix}$	*loc*	*loc*	*loc*	*loc*
$\begin{bmatrix} \text{GLOSS:"son"} \end{bmatrix}$	*fili*	*fili*	*fili*	*fili*

Then, the longest prefix shared by the forms in each row is stripped and assigned to the paradigm stems:

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	*us* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$	*i* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:pl} \end{bmatrix}$	*um* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:sg} \end{bmatrix}$	*os* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:pl} \end{bmatrix}$
amic $\begin{bmatrix} \text{GLOSS:“friend”} \end{bmatrix}$	-	-	-	-
loc $\begin{bmatrix} \text{GLOSS:“place”} \end{bmatrix}$	-	-	-	-
fili $\begin{bmatrix} \text{GLOSS:“son”} \end{bmatrix}$	-	-	-	-

5.3.4 Handling Noise in the Data

If there is no noise in the paradigm (i.e. the predicted forms perfectly match the attested forms), then the cells at this point will be empty. However, if there is any noise, it must be encoded in the paradigm so that the paradigm's output for each cell matches the attested form for that cell. Errors are encoded as a three-component (M, S, N) tuple: M is an integer indicating the number of characters, if any, to delete from the end of the stem. Similarly, N is an integer representing the number of characters, if any, to delete from the beginning of the suffix. S is a string of characters to be inserted between the stem and suffix. These errors are incorporated into the cost function of the paradigm; the cost of an error code is defined as the number of letters deleted plus the number of letters added. Encoding errors in this way allows for easy handling of phonological changes occurring at the morpheme boundary, but it is flexible enough to allow wholesale rewriting of the form if necessary.

For example, in Classical Latin, the nominative plural of *filius* is *fili*. However, the paradigm, given the prefix stem *fili*, the inflectional suffix *i*, and an empty error tuple $(0, -, 0)$, predicts the form *filii*.

$$\textit{fili} + (0, -, 0) + i \rightarrow \textit{fili}$$

This discrepancy between the attested form and the form predicted by the paradigm

could be handled either by deleting the trailing *i* from the stem, or by omitting the *i* of the suffix; both of these operations can be handled by (M, S, N) edit codes. To remove the stem's *i*, the code $(1, -, 0)$ would be used. To remove the suffix *i*, the code $(0, -, 1)$ would be used.

$$\textit{fili} + (1, -, 0) + i \rightarrow \textit{fili}$$
$$\textit{fili} + (0, -, 1) + i \rightarrow \textit{fili}$$

The search function will naturally decide when to include a form with an error and when to omit it from the paradigm, based on whether the cost of including the error in the paradigm outweighs the cost of encoding the error-containing biform elsewhere in the morphology. For example, the Latin noun *comes* "companion" could possibly be translated as "friend". Suppose that a form of *comes* was intermingled with the forms of *amicus*. It is not allowed for any cell in the paradigm to contain more than one form.[5] However, it is possible that, say, the accusative form *comitatem* might be included in the place of *amicum*. Assuming that paradigm inference arrived at the optimal solution (which does not differ from the true solution), the result would be a lot of noise coded in the *comitatem* cell.

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	*us* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$	*i* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:pl} \end{bmatrix}$	*um* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:sg} \end{bmatrix}$	*os* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:pl} \end{bmatrix}$
amic $\begin{bmatrix} \text{GLOSS:"friend"} \end{bmatrix}$	-	-	(4,comitate,1)	-
loc $\begin{bmatrix} \text{GLOSS:"place"} \end{bmatrix}$	-	-	-	-
fili $\begin{bmatrix} \text{GLOSS:"son"} \end{bmatrix}$	-	-	-	-

[5] This would defeat the purpose of the paradigm; in cases where a language legitimately has alternate forms for the same inflection of a given word, these alternate forms must be handled by alternate paradigms.

The error tuple correctly predicts the form for the cell by deleting all four characters of the stem, as well as the first *u* in the suffix, and inserting the remainder of the *comitatem* form:

$$amic + (4, comitate, 1) + um \rightarrow comitatem$$

But the cost of the error tuple, when weighted against the cost of encoding the biform for *comitatem* elsewhere in the morphology, would most likely prohibit any merge step that grouped forms of *comes* with *amicus*. Thus, although the gloss feature is not a perfect predictor of whether two forms belong to the same lexeme, the gloss feature in combination with the inference method and the objective function works to group together biforms that are related both by their features *and* by their forms.

This example can also illustrate the limitations of the procedural paradigm inference method described here. In the table above, the paradigm analysis is in fact exactly the same as the correct analysis (same paradigm features; same stems; same suffixes), but differs only in the error contents. However, results of the inference method as described here could very well be sub-optimal; the fact that *comitatem* ends with *-em*, not *-um*, means that the longest shared suffix by the forms in its column is *-m*, resulting in an incorrect analysis of that suffix. Likewise, the stem inference for the *amicus* row would be impeded. In practice, a majority rule is used for prefix and suffix inference: if a the prefix or suffix is shared by more than 50% of the forms in its group, then it a net benefit cost-wise to encode it on the row or column. This is true because the cost of encoding the error associated with deleting the prefix or suffix, in less than half of the cells, should still be less than the cost of encoding that same prefix or suffix in more than half of the cells. However, this is only a heuristic, and if a sufficient number of noisy forms are present in the cluster, a sub-optimal result will ensue.

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	*us* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$	*i* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:pl} \end{bmatrix}$	*m* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:sg} \end{bmatrix}$	*os* $\begin{bmatrix} \text{CASE:acc} \\ \text{NUM:pl} \end{bmatrix}$
- $\begin{bmatrix} \text{GLOSS:“friend”} \end{bmatrix}$	(0,amic,0)	(0,amic,0)	(0,comitate,1)	(0,amic,0)
loc $\begin{bmatrix} \text{GLOSS:“place”} \end{bmatrix}$	-	-	(0,u,0)	-
fili $\begin{bmatrix} \text{GLOSS:“son”} \end{bmatrix}$	-	-	(0,u,0)	-

This example calls attention to the fact that the paradigm inference method is procedural and deterministic, though still effective in the context of the broader search procedure. Finding the *optimal* paradigm analysis would be more costly in terms of processing time, since it involves considering many possible prefixes, suffixes, and feature assignments. The trade-off is worth making since the difference in results of an optimal and sub-optimal search is most prominent when the data is noisy, and the outer loop of the search procedure is designed to limit noise within the clusters. Thus, any cluster which survives to the final analysis should be relatively noise-free, minimizing the impact of the sub-optimal paradigm inference procedure. The fact that the table above is filled with errors merely illustrates that it is a poor candidate for a paradigm, and any merge that combined forms of *comes* and *amicus* would be rejected by the search procedure.

5.3.5 Handling Sparse Data

This paradigm inference method works best when the paradigm table is well-populated, but it is still effective with smaller subsets of the data, provided certain conditions are met. For example, to infer that the biform {*filium*, GLOSS:“son” POS:n GEN:m CASE:acc NUM:pl} contains the morpheme {*um*, CASE:acc NUM:pl}, it is necessary to have in the training corpus at minimum either an example of an-

other instance of the *um* suffix or an example of another instance of the *fili* stem. This is true regardless of the inference method and whether the analysis is performed by hand or by computer.

The main consideration when dealing with small datasets is the cost-based metric guiding the search. MDL is based on an assumption of arbitrary long message length, which is clearly not the case when dealing with an actual corpus of language data. When the data set becomes truly small, the relationship between the cost of the paradigm and the cost of the data forms comes into play. In general, if the number of stems in a paradigm is less than the number of suffixes, then extra material will tend to be added to the stems, since this is cheaper than encoding the same material on multiple affixes. However, if the number of stems is greater than the number of suffixes, then the reverse will be true.

For example, if the corpus contained the same stems as the earlier examples, but only the nominative singular forms, then it could become cheaper to encode the /*c*/ on the suffix, rather than twice on the stems:

$f_p = \begin{bmatrix} \text{POS:noun} \\ \text{GENDER:m} \end{bmatrix}$	*cus* $\begin{bmatrix} \text{CASE:nom} \\ \text{NUM:sg} \end{bmatrix}$
ami $\begin{bmatrix} \text{GLOSS:“friend”} \end{bmatrix}$	- amicus
lo $\begin{bmatrix} \text{GLOSS:“place”} \end{bmatrix}$	- locus
fili $\begin{bmatrix} \text{GLOSS:“son”} \end{bmatrix}$	(0,-,1) filius

5.3.6 Search Limitations

For another limiting case, imagine a scenario where a certain inflection is only attested by a single form, and that form's stem does not appear elsewhere in the

corpus. Even in the absence of these other forms for comparison, a linguist might make a reasonable guess at what material belongs to the stem and what to the suffix, based on the lengths of other suffixes in the language. However, from a cost standpoint, there is no justification for choosing how to break up the form, or even whether to break it up at all. In fact, from a cost standpoint there may be no motivation to even include that form in the paradigm, but rather to leave it out as an unanalyzable form.

The clustering procedure described here is a greedy, hill-climbing search and is not guaranteed to yield the optimal analysis. However, by choosing the best merge operation at each iteration, most missteps can be avoided. For example, in Latin there are different noun classes with genitive plural suffixes *-arum*, *-orum*, and *-um*. Merging an *-um* form with an *-orum* form could in principle yield a cost savings, since the *-um* material of both suffixes is shared. If this choice was taken early in the inference process, it would yield messy results with two separate noun classes merged into a single paradigm. However, in practice this merge should not occur, since there would be greater cost savings to be had by other merges (e.g. merging forms that share a stem, or merging two *-orum* forms).

A misstep may occur in some unlikely circumstances. For example, if there exists a very long suffix that is shared by two paradigms, then clustering together forms that share this suffix could be the chosen as the best merge, *if* the length of the suffix is longer than any of the stems. In order for this step to be incorrect, it must also be the case that there are many other (shorter) suffixes in the two paradigms which differ from one another, such that it would have been better not to make the initial merge operation.

These are extreme cases, and certainly there is a lower limit on the size of the corpus necessary to produce good linguistic results. While it is possible to imagine artificial scenarios like this in which particular data sets will yield suboptimal analyses,

for most real-world data sets this clustering approach will be effective.

To be truly representative of the language, a corpus should produce an analysis that does not change with the addition of further data from the language (Harris, 1951). Furthermore, it is not a weakness, but a strength, of the algorithm that it does not make predictions that are not supported by the evidence in the corpus, even if those predictions produce reasonable analyses. The virtue of an algorithmic approach to linguistic analysis is that it always provides consistent results when presented with the same input. However, in many cases when dealing with electronic data sources, it is not possible to collect additional data, and the objective must always be to produce the best possible analysis with the data given.

The experiments described below illustrate the performance of this algorithm. The first experiment, using tagged biforms from a Latin dictionary, explores effect that corpus size has on the quality of the inferred morphological analysis. The second experiment, using data derived from the parallel corpus of Estonian, explores how this inference algorithm performs on a small, real-world corpus, and compares it to existing methods for unsupervised morphological inference.

5.4 Performance & Evaluation

5.4.1 Experiment: Latin

To explore the performance of this induction algorithm on various types of data sets, I created 6 small, targeted subsets of the Whitaker corpus of Latin wordforms. The subsets were chosen to include many inflected forms of a small set of lemmas. These sets are summarized in table 5.2. More information on the Whitaker data can be found in Appendix C. Each set was used to train a paradigmatic morphology, which was evaluated against the gold-standard analyses contained in the corpus.

Each data set was randomly divided into training and test sets, with the constraint

Name	Size	Description
Set1	84	First-declension nouns (8 lemmas).
Set2	86	Second-declension nouns (8 lemmas).
Set3	317	First-conjugation verbs (4 lemmas).
Set4	404	Second-conjugation verbs (4 lemmas).
Set5	882	Set1 through Set4, combined.
Set6	170	Set1 and Set2, combined.

Table 5.2: Summary of the Latin data sets.

that each training set was required to include at least one form of each lemma. (This seems justifiable, since no morphology can reasonably be expected to generate forms of a previously-unseen stem.) For this experiment, I looked at training set sizes of 10%, 50%, and 90% of the original data set. Starting with the six original data sets, this produced 18 paired training and test sets.

For each of the 18 training sets, a morphology was created using the inference method described above, in which clusters of forms are iteratively merged in a greedy search procedure. This morphology was then saved and used to predict forms for all of the tokens in the appropriate test set. (The test set is different in each case, since the assignment of test and training tokens is random. However, in every case the test set consists of forms that were not seen in the training process.)

For evaluation, the learned paradigm is used to predict word forms for each possible inflection of each stem. A predicted form is judged as either correct or incorrect based on whether it exactly matches the form in the original data set. The reported accuracy is the probability that the predicted form is correct. (In the case that the morphology predicts more than one form for a given input, its probability mass is distributed among all the predicted forms, which may independently be either correct or incorrect.) Figure 5.6 plots the accuracy against the size of the training set, illustrating how performance increases with more training data. Figure 5.7 plots the accuracy for the six sets when training is performed using less than the full set. This shows that there is a definite minimum level of data necessary to train an effective

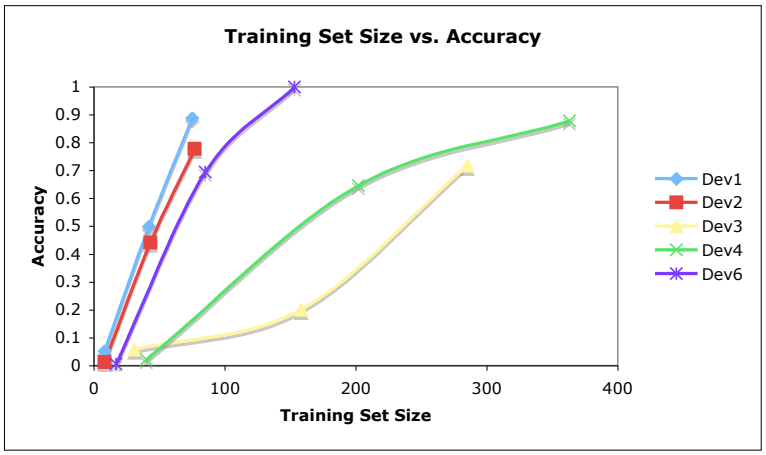


Figure 5.6: Illustration of the effect of training set size (measured by the number of tokens) on generation accuracy for six different data sets.

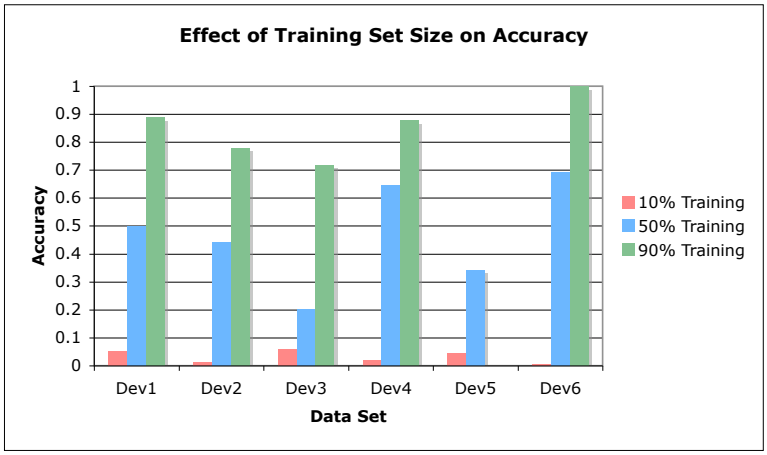


Figure 5.7: Illustration of the effect of training size (as a percentage of the entire data set) on generation accuracy for six different data sets.

model. While the accuracy of the induced morphology depends heavily on the size of the training set, good results are still possible with relatively small sets (hundreds of forms), provided that those data sets contain a good representation of the possible types of wordforms.

5.4.2 Experiment 2: Bitext-Aligned Estonian

This experiment combines the morphology induction process with the biform generation process from chapter 4, using a morphologically-tagged parallel corpus of Estonian and English text. Like Latin, Estonian is a suffixing language; however as a non-Indo-European language (Estonian is a Baltic-Finnic language in the Uralic language family), Estonian is a good complement to Latin. Estonian is not a resource-poor language—it is the national language of Estonia, with over a million speakers (Lewis, 2009)—but the existence of a morphologically-tagged corpus is essential to make evaluation of this experiment possible, and the novel-length text used in this experiment is a reasonable approximation of the amount of data available for many RPLs.

Data

This experiment uses data from the morphologically-disambiguated corpus of Estonian produced at the University of Tartu.[6] The corpus consists of several texts which have been manually tagged for rich morphological features (e.g. tense, number, person, case). This experiment uses only the portion of the corpus containing George Orwell's novel *1984*. That text was paired with an English version of the same novel to produce a parallel corpus.

The biforms used as input for the morphological inference algorithm were produced according to the bitext alignment and enrichment procedure described in chapter 4.

[6] `http://www.cl.ut.ee/korpused/morfkorpus/` For more information on this corpus see Appendix D.

For this experiment, only those forms classified as nouns (i.e. Estonian word forms that were aligned with English nouns) are used. The nouns are tagged with the part-of-speech feature, a number feature, a proper/common noun feature, an English gloss feature (the stem of the English word with which the token is aligned), and a dependency feature from the English parse. One of the more interesting aspects of this experiment is how the Estonian case features compare to the dependency features from the parse of the English sentence.

Method

For comparison, I ran the monolingual text through two unsupervised monolingual morphological inference programs, *Linguistica* (Goldsmith, 2001) and *Morfessor 1.0* (Creutz and Lagus, 2005). Both programs were run on the full text of the novel, using the default recommended parameters.

The clustering algorithm described earlier in this chapter is rather slow, and future work will need to address speed optimizations. For this experiment, I ran the algorithm only on the word forms that were predicted to be nouns by the word alignment. To further improve the running time of the algorithm, not all merges were considered (merges were constrained to forms tagged with the same lemma feature), and so the performance is impaired somewhat compared to the full algorithm.

Evaluation and Results

Unlike the Latin dataset, this is a true corpus: the tokens are whatever wordforms happen to be contained in the text, and so it is not possible to directly assess whether a form predicted by the analysis is or is not a valid Estonian word simply by checking whether it exists in the corpus. Therefore, I used the evaluation procedure adopted by the Morpho Challenge (Kurimo et al., 2010), which is designed to estimate precision and recall given the restrictions that not all wordforms are attested in the data and

	Precision	Recall
Morfessor	57.5% (167/290)	34.3% (116/337)
Linguistica	56.6% (176/311)	38.7% (130/337)
Biform (no glosses)	68.4% (266/389)	20.4% (69/337)
Biform (glosses)	74.7% (275/368)	16.3% (55/337)

Figure 5.8: Comparison of morphology induction systems on nouns in the Estonian corpus.

that the predicted and gold morphemes may be labeled differently from one another. To measure precision, shared-morpheme word pairs are drawn from the predicted analyses; for each shared-morpheme pair there must be at least one morpheme that appears in the analysis of both words. Each shared-morpheme pair is evaluated as correct if the gold standard also analyzes these two wordforms as sharing a morpheme, and is evaluated as incorrect otherwise. To measure recall, the shared-morpheme word pairs are sampled from the gold-standard and compared against the predictions; each pair is evaluated as correct if and only if the predicted analysis also shows the two words as both containing a shared morpheme.

Figure 5.8 shows the evaluation results for Morfessor and Linguistica as well as the biform inference method described here. The standard biform method results are given in the row labeled "glosses". The algorithm described in this dissertation outputs morphemes consisting of both its form and its gloss, whereas the morphemes produced by unsupervised, monolingual approaches are unglossed strings. Therefore, I added a "no glosses" analysis that strips the gloss component from the morphemes produced by my algorithm. While this greatly limits the usefulness of the inferred morphemes, it is a better comparison to the monolingual methods, since a string that represented a single morpheme in the monolingual, unsupervised case may actually comprise many unique morphemes in the biform approach.

All evaluation was performed only on wordforms that were tagged as nouns in the gold data. For each data set, 400 forms were sampled, but the actual number of pairs

evaluated is slightly less due to the fact that for some wordforms (e.g. hapax forms), there are no other forms in the corpus that share a morpheme.

One clear trend is that recall is low for all four systems. One likely cause is the fact that noun stems in Estonian exhibit allomorphy, and a purely segmentation-based approach to morphology will end up producing multiple different stems for the same noun. While the error-correcting capability of the biform-based approach mitigates this problem somewhat, the recall is decreased even further because the biform induction method only identifies a single suffix for each form, while many Estonian wordforms exhibit stem-internal derivational morphology in addition to the inflectional suffixing. Also, some forms were inevitably misidentified during the alignment; any form that was not tagged as a noun during the bitext alignment stage will be missing here, and this will have a negative impact on the recall.

Another trend is that the precision of the analyses produced by the algorithm presented in this dissertation is higher than that of the other two systems. It would be expected that precision is higher for the glossed morphemes; variation in the glosses due to imperfect word alignments means that there are likely several forms of each true morpheme, and this reduces the likelihood of any chance collisions between morphemes. However, the precision is higher even for the unglossed version of the output, suggesting that the segmentation accuracy of the biform approach is better than that of the unsupervised methods.

Table 5.3 shows some examples of the Estonian word forms that were analyzed in this experiment. This subset shows three different paradigms (separated by horizontal lines). These examples illustrate some strengths and some weaknesses of this approach to morphology induction. First, it is a good thing that the first paradigm (which contains a form that is mislabeled due to a word-alignment error) is kept separate from the other two even though it shares characteristics of both. The first form and those in the third paradigm are labeled with the same lemma, but they are

Form	Stem	Suffix
tunni	tunni/lemma:pocket	
tunni	tunni/lemma:hour	+∅/dep:DEP/num:SG
tunni	tunni/lemma:hour	+∅/dep:NSUBJ/num:PL
tunni	tunni/lemma:hour	+∅/dep:PREP_IN/num:PL
taskus	taskus/lemma:pocket	+∅/dep:POBJ/num:SG
taskusse	taskus/lemma:pocket	+se/dep:PREP_IN/num:SG
taskusse	taskus/lemma:pocket	+se/dep:PREP_INTO/num:SG
taskust	taskus/lemma:pocket	+t/dep:PREP_OUT_OF/num:PL
taskust	taskus/lemma:pocket	+t/dep:PREP_OUT_OF/num:SG
taskusse	taskus/lemma:pocket	+se/dep:PREP_TO/num:SG

Table 5.3: Example of the morphological analysis produced using Estonian-English bitext as input.

not clustered together due to their differences in form. The first form and those in the second paradigm share identical forms, but different features. In this way, the algorithm handles an alignment error as expected. Furthermore, the algorithm rightly strips the shared features (i.e. these are all common nouns) and uses the remaining features (number and the dependency) to gloss the suffixes.

The actual Estonian suffixes represented in this data subset are the illative *–sse*, inessive *–s*, and elative *–st*. While the morpheme boundary is not perfectly detected (the stem-final *-s* should in fact belong to the suffixes), the algorithm is simply doing its job by minimizing the cost of the paradigm. The dependencies used to gloss the suffixes show how English syntactic features correspond to Estonian morphological features: the inessive case marker appears with the labeled dependencies PREP_INTO, PREP_TO, and PREP_IN, all of which accurately represent the semantic sense of inessive. A further, non-trivial, improvement to this algorithm would be to recognize that these morphemes, which all share the same form and similar, but not identical, features, are in fact one morpheme. The English dependency features do not always align perfectly with the Estonian usage; for example, the PREP_IN feature shows up on one of the *tunni* forms which is not actually inessive case. However, even in

its current form the analysis produced by this approach is linguistically meaningful and is of much greater use from a linguistic standpoint than an unglossed, purely segmentation-based, morphological analysis.

5.5 Conclusion

In this chapter I have presented my algorithm for paradigmatic morphology induction from an input of biform word tokens. The algorithm follows an iterative clustering procedure to find the optimal analysis under an MDL-inspired cost-minimization objective function. The results of the two experiments show that the inference algorithm effectively analyzes biforms into stem and suffix components, and that the paradigms which are inferred are able to accurately analyze and predict unseen biforms. Additionally, the glosses produced by the morpheme induction process contain meaningful representations of the function of the morphemes.

This is a novel form of morphology induction, as previous efforts are either unsupervised methods based on monolingual, unlabeled input, or fully-supervised methods in which the true segmentations are included in the training data. The algorithm described here could be improved in several ways to make it more efficient or robust to noisy data. The search procedure itself considers every possible merge at each iteration; this is inefficient and could be sped up by heuristics that estimate cluster similarity and only attempt to merge clusters that are estimated to have a comparatively high similarity. Speed improvements will be necessary in order to fully apply this algorithm to modestly-sized data sets. However, even in its current state this is an effective method that should prove useful for enriching bitext sources from RPLs through morphological analysis.

CHAPTER 6

Conclusion

The work and experiments described in this dissertation make it clear that electronic texts are rich sources of language data and that it is possible to use existing NLP tools, with appropriate modifications, to leverage that data for grammar induction. In this dissertation, I explored the feasibility and the challenges of extracting bitext from digitally-scanned documents, and I showed that a morphological inference method based on bitext-derived input can be an effective tool for automated linguistic analysis. Such methods to enrich existing electronic texts and produce databases of richly-annotated linguistic data are necessary to enable the future forms of cross-lingual computational linguistics research that will advance our understanding of the nature of language.

However, there exist several obstacles to this ultimate goal, which will need to be addressed by further research. Detailed discussions of directions for future work are given in each of the chapters of this dissertation; here I mention two large-scale areas for future work based on the topics in this dissertation: improving the data collection process and improving grammatical inference as it applies to resource-poor languages.

One obstacle to data collection is the current limitations in access to electronic data. Copyrighted works, proprietary data collections, and non-public data of indi-

vidual researchers, are problematic or impossible to obtain. A second, major obstacle is the OCR quality of electronic data. The OCR text of online collections is error-prone and omits non-Latin scripts entirely. While this can be improved somewhat with commercial OCR software, it remains problematic, particularly since using such software requires identifying the language beforehand and only works for a selection of known languages.

There is also room for improvement in terms of the data that is extracted from the documents. The bitext extraction method could be improved via better language ID, both at the word level and at the phrase level. Bitext extraction would also benefit from improved handling of structured text, although this partly is dependent on better OCR methods. Error accumulation from processing stages is also an issue. In the pipeline described at the beginning of this dissertation, bitext is extracted (via a noisy process) from electronic text, then the words within the bitext are aligned (via a noisy process), and features are transferred from English to non-English words, despite a mismatch in the grammatical systems of the two languages. In order for this pipeline to ultimately be successful, it will be necessary to incorporate intelligent error-correcting and noise-reduction methods into the process.

Additionally, there remains the possibility of extracting more of the analysis contained in the grammar by using deeper methods of information extraction. While this dissertation focused on morphology, there are many other types of linguistic knowledge contained within the typical grammar.

While the morphological analyses produced by the methods describe here are an improvement over previous techniques, they still fall short of the quality which would be desired for a linguistic analysis, as any linguist will be quick to recognize. The world's languages abound with any number of diverse morphological processes that are much more fascinating and complex than simple suffixing. String concatenation is a familiar principle to any computer scientist, and it acts well enough as an approx-

imation for the morphological processes common in many languages. However, in order to adequately handle the numerous, diverse, languages that comprise the realm of RPLs, computational linguists need to delve beyond affixation and into the depths of reduplication, templates, vowel harmony, and other morphological processes. Additionally, the paradigm-based inference method I describe only handles the case where a form consists of a single stem and a single suffix. To be more useful, this model will need to be extended to handle stem-internal derivational morphology, as well as cases of multiple affixation.

For linguists, there is a need for usable tools: corpora and software that can easily be obtained and used by researchers who are not necessarily computational linguists. The types of automated processing described in this dissertation could either be added as extensions to existing language documentation tools or could form the basis of new software or web applications. Either way, these methods should prove useful for linguists organizing their own, newly-collected, data, as well as for linguists looking for evidence and examples relevant to their own work from among the quantities of linguistic data that already exists in electronic form. In any case, it is my hope that the work in this dissertation will help to facilitate new types of linguistic research on a wide base of languages.

APPENDICES

APPENDIX A

Software

To complete the experiments described in this dissertation, I used a combination of existing software packages and custom scripts and software I developed myself. The code written specifically for this project consists of Python modules and *bash* shell scripts, so to run this code, all one needs is a bash shell and Python 2.6+ interpreter. The majority of processing was performed on machines running Apple OS X 10.5 and Python version 2.6, although additional processing was run on Linux machines using Python versions 2.6+. Links to download this software and its documentation can be found on my personal web site.

However, several external software packages were used in conjunction with the Python code, and these may need to be built separately. Below is an overview of the external software used. Some of the software packages only require a Python, Java, or Perl interpreter, while other packages must be compiled in order to run.

Package	**Language**	**Summary**
NLTK	Python	Various NLP support libraries
Stanford Parser	Java	Phrase-structure and dependency parsing
PC-KIMMO	C	English morphological analysis
Morfessor	Perl	Unsupervised morphological segmentation
Hunalign	C++	Sentence alignment
GIZA++	C++	Statistical word alignment

APPENDIX B

The Tatoeba Corpus of Sentence Pairs

Tatoeba[1] is a community-driven, open-source database, whose goal is to collect example sentences from as many languages as possible. Because it is user-generated, not all of the content is reliable; however, its growing size and broad coverage of languages make Tatoeba an interesting resource for computational linguistic research.

The Latin dataset consists of 347 English-Latin sentence pairs downloaded from the Tatoeba database, comprising around 2,100 English words and 1,500 Latin words. The sentence pairs are typically fairly short and linguistically diverse, including questions and conversational phrases. In this regard, the data is qualitatively quite different than most corpora, which tend to be genre-focused, typically including large amounts of fiction and news. A few examples of sentence pairs from the Tatoeba dataset are given in table B.1

The French data set used for the statistical translation selection experiments contains 51,129 sentence pairs. Because the database is continually growing, the number of English-French translation pairs in the current version of the Tatoeba database is much larger than that number.

[1] `http://tatoeba.org`

English	Latin
I love you.	Te amo.
Where is the bathroom?	Ubi est lātrīna?
Like father, like son.	Qualis pater, talis filius.
What are you doing?	Quid facis?
What are you doing?	Quid vos facitis?
What do you do?	Quid facis?
I believe you.	Crēdō tibi.
I believe you.	Vōbīs crēdō.

Table B.1: Example sentence pairs from the Tatoeba database.

APPENDIX C

The Whitaker Corpus of Analyzed Latin Wordforms

The dataset referred to as the Whitaker data set is based on morphological analyses produced by the *words* software, written and released for public use by William Whitaker. The software has a public web interface at `http://archives.nd.edu/words.html`, with additional documentation and source code available on Whitaker's website, `http://users.erols.com/whitaker/words.htm`.

The Whitaker dataset was generated by feeding a long list of Latin wordforms into the *words* software for analysis. The wordforms were taken from the `LISTALL` text file, which Whitaker describes as a list of all forms that can be generated from his dictionary and inflections. The `LISTALL` file contains 1,034,157 inflected forms. Because some forms can be analyzed in more than one way, this resulted in 1,623,818 fully-analyzed tokens, which constitute the Whitaker dataset used in these experiments.

The output of the *words* program includes the segmentation of the wordform into stem and suffix, as well as an specification of the word subclass (noun declension or verb conjugation variety) to which the form belongs. The inflected features are also

Form	Segments	Lemma	POS	Subclass	Additional Features
abaci	abac+i	7	N	2.1	GEN S M
abaci	abac+i	7	N	2.1	NOM P M
abactae	abact+ae	9	VPAR	3.1	GEN S F PERF PASSIVE PPL
abactia	abacti+a	11	ADJ	1.1	NOM S F POS
abactum	abact+um	9	SUPINE	3.1	ACC S N
abacturarum	abact+urarum	9	VPAR	3.1	GEN P F FUT ACTIVE PPL
abarceamur	abarc+eamur	28	V	2.1	PRES PASSIVE SUB 1 P
abarcet	abarc+et	28	V	2.1	PRES ACTIVE IND 3 S

Table C.1: Example tokens from the Whitaker dataset.

given. Table C.1 illustrates the type of output produced by *words* by showing a few different tokens and their features.

The *words* program gives a dictionary definition for each lemma; in my dataset this is replaced by an integer key that uniquely identifies each lemma.

In all, there are 12 part-of-speech tags and 104 subclasses. Table C.2 lists the complete range of part-of-speech subclasses in the dataset, along with a count of the number of tokens belonging to each subclass.

POS	Subclass	Token Count	POS	Subclass	Token Count	POS	Subclass	Token Count
ADJ	0.0	688	N	2.9	14	SUPINE	3.2	44
ADJ	1.1	138226	N	3.1	48765	SUPINE	3.3	32
ADJ	1.2	4663	N	3.2	4137	SUPINE	3.4	561
ADJ	1.3	123	N	3.3	10422	SUPINE	5.1	30
ADJ	1.4	176	N	3.4	3179	SUPINE	6.1	66
ADJ	1.5	72	N	3.6	95	TACKON		25
ADJ	1.6	220	N	3.7	1297	V	1.1	307591
ADJ	1.7	432	N	3.9	738	V	2.1	46291
ADJ	2.1	11	N	4.1	8592	V	3.1	213165
ADJ	2.6	787	N	4.2	91	V	3.2	2455
ADJ	2.7	195	N	4.3	12	V	3.3	628
ADJ	2.8	920	N	4.4	6	V	4.1	42238
ADJ	3.1	9765	N	5.1	1498	V	5.1	1179
ADJ	3.2	19375	N	9.8	63	V	5.2	65
ADJ	3.3	283	N	9.9	142	V	6.1	4697
ADJ	3.6	38	NUM	1.1	66	V	6.2	253
ADJ	9.8	23	NUM	1.2	95	V	7.1	30
ADJ	9.9	24	NUM	1.3	150	V	7.2	16
ADV	COMP	339	NUM	1.4	1188	V	7.3	176
ADV	POS	6666	NUM	2.0	3083	V	9.1	50
ADV	SUPER	363	PREP	ABL	33	V	9.2	11
CONJ		103	PREP	ACC	55	V	9.3	97
INTERJ		104	PRON	1.0	134	V	9.9	15
N	1.1	53401	PRON	3.1	50	VPAR	1.1	276127
N	1.6	2074	PRON	4.1	608	VPAR	2.0	42
N	1.7	2170	PRON	4.2	609	VPAR	2.1	34565
N	1.8	1036	PRON	5.1	11	VPAR	3.1	229513
N	2.1	29675	PRON	5.2	6	VPAR	3.2	2729
N	2.2	31337	PRON	5.3	12	VPAR	3.3	2040
N	2.3	1261	PRON	5.4	6	VPAR	3.4	40475
N	2.4	7839	PRON	6.1	80	VPAR	5.1	1010
N	2.5	669	PRON	6.2	97	VPAR	6.1	3527
N	2.6	3392	SUPINE	1.1	5666	VPAR	6.2	56
N	2.7	45	SUPINE	2.1	568	VPAR	7.2	2
N	2.8	2876	SUPINE	3.1	3078			

Table C.2: List of all part-of-speech subclasses in the Whitaker dataset.

Since the analysis includes not only the segmentation of the form into stem and

suffix, but also a fairly fine-grained classification of word classes, the Whitaker data is particularly useful for evaluating the paradigm-classification subtask of morphology induction.

Whitaker Development Set

The development set consists of the following sets of lemmas, which were chosen to represent certain subclasses of nouns and verbs.

First-Declension Nouns

(All are of the first subclass variety)

Lemma	ID	gloss
abra	188	"maid"
bucca	6894	"jaw"
cartula	8196	"scrap of paper"
eminentia	17737	"eminence"
incola	20995	"inhabitant"
maxilla	23576	"jaw"
phantasia	26467	"imagination"
spira	30378	"coil"

Second-Declension Nouns

(All are of the first subclass variety)

Lemma	ID	gloss
abacus	7	“abacus”
baetulus	5903	“sacred stone”
capillus	7856	“hair”
haedus	20020	“kid (young goat)”
ornus	25442	“ash tree”
pessulus	26422	“bolt”
spondeus	30406	“spondee”
vitricus	33036	“stepfather”

First-Conjugation Verbs

Lemma	ID	gloss
incolo	20998	“inhabit”
orno	25438	“equip”
scrutino	29492	“investigate”
vexo	32804	“shake”

Second-Conjugation Verbs

Lemma	ID	gloss
adluceo	1188	“shine upon”
fulgeo	19451	“gleam”
seneo	29713	“be old”
video	32864fg	“see”

APPENDIX D

The Estonian Morphologically Disambiguated Corpus

The Morphologically Disambiguated Corpus, available for download at the University of Tartu website,[1] consists of 513,000 words of Estonian text. Each wordform is tagged with a number of morphologically features as well as a lemma+suffix segmentation. The corpus includes the text George Orwell's novel *1984*, as well as a collection of articles from mainly news and legal sources, which form a subset of the Corpus of Written Estonian. The project originated as part of the Multext-East project,[2] which has produced a variety of NLP resources for Eastern European languages (Dimitrova et al., 1998).

The website states that the corpus was manually tagged, suggesting that morphological analysis software for Estonian either does not exist or was unavailable to the researchers at the time when the corpus was created (2002–2003). This is a true corpus and not a complete wordlist: although the attested forms only constitute a small fraction of all wordforms possible in the Estonian language, each form is disambiguated and analyzed appropriately for the given context in the corpus. Figure D.1

[1] `http://www.cl.ut.ee/korpused/morfkorpus/`
[2] `http://nl.ijs.si/ME/`

gives an example of a few lines from the corpus, showing how the forms, segmentations, and feature are presented.

<s>		
Oli	ole+i	_V_ main indic impf ps3 sg ps af
külm	külm+0	_A_ pos sg nom
selge	selge+0	_A_ pos sg nom
aprillipäev	aprilli_päev+0	_S_ com sg nom
,	,	_Z_ Com
kellad	kell+d	_S_ com pl nom
lõid	löö+id	_V_ main indic impf ps3 pl ps af
parajasti	parajasti+0	_D_
kolmteist	kolm_teist+0	_N_ card sg nom l
.	.	_Z_ Fst
</s>		

Figure D.1: Example file contents of the Estonian morphologically-disambiguated corpus.

The documentation for the corpus lists 579 different rich POS tags. However, as each tag is composed of individual features, this decomposes to a smaller set of individual morphological features. Figure D.2 lists these features (omitting the features for punctuation), organized by their grammatical function.

[3]This appears to be a compound class for genitive adjectives. Not knowing Estonian, I'm not sure why this exists as a separate part-of-speech class.

Abbreviation	*English Explanation*
Parts of Speech	
_A_	Adjective
_D_	Adverb
_G_	Genitive Singular Positive Adjective[3]
_I_	Interjection
_J_	Conjunction
_K_	Adposition
_N_	Numeral
_P_	Pronoun
_S_	Noun
_T_	Foreign
_V_	Verb
_X_	Adverb
_Y_	Abbreviation
_Z_	Punctuation
Number	
sg	singular
pl	plural
Case	
abes	abessive
abl	ablative
ad	adessive
adit	aditive
all	allative
el	elative
es	essive
gen	genitive
ill	illative
in	inessive
kom	comitative
nom	nominative
part	partitive
term	terminative
tr	translative
Person	
ps1	first person
ps2	second person
ps3	third person
Noun Type	
com	common noun
prop	proper noun

Abbreviation	*English Explanaiton*
Adjective Degree	
pos	positive
comp	comparative
super	superlative
Verb Type	
aux	auxiliary
main	main verb
mod	modal
Verbal Forms	
ger	gerund
partic	participle
inf	infinitive
sup	supine
Verb Mood	
cond	conditional
imper	imperative
indic	indicative
quot	quotative
Verb Tense/Aspect	
impf	imperfect
past	past
pres	present
Verb Negation	
af	afirmative
neg	negative
Verb Voice	
imps	passive
ps	active
Adposition Types	
post	postposition
pre	preposition
Conjunction Types	
crd	coordinating
sub	subordinating
Numeral Features	
card	cardinal
digit	digit
l	letter
ord	ordinal
roman	roman
Abbreviation Types	
adjectival	adjectival
adverbial	adverbial
nominal	nominal
verbal	verbal

Figure D.2: Features used in the Estonian morphologically-disambiguated corpus.

APPENDIX E

Universal Part-Of-Speech Tag Set

This is a "universal" set of twelve part-of-speech tags as proposed by Petrov et al. (2012). The motivation of this tagset is to provide a minimal, basic set of tags that will be useful and applicable to a wide set of languages. Table E.1 below lists the twelve tags in this set. Table E.2 lists how these tags map to the Penn Treebank tagset, and table E.3 lists how these tags map to the Estonian tagset.

Tag	**Description**
VERB	verbs (all tenses and modes)
NOUN	nouns (common and proper)
PRON	pronouns
ADJ	adjectives
ADV	adverbs
ADP	adpositions (prepositions and postpositions)
CONJ	conjunctions
DET	determiners
NUM	cardinal numbers
PRT	particles or other function words[1]
X	other: foreign words, typos, abbreviations
.	punctuation

Table E.1: The set of twelve universal part-of-speech tags

WSJ	Universal	WSJ	Universal	WSJ	Universal	WSJ	Universal
!	.	FW	X	NN|VBG	NOUN	UH	X
#	.	IN	ADP	NP	NOUN	VB	VERB
$	.	IN|RP	ADP	PDT	DET	VBD	VERB
''	.	JJ	ADJ	POS	PRT	VBD|VBN	VERB
(	.	JJR	ADJ	PRP	PRON	VBG	VERB
)	.	JJRJR	ADJ	PRP$	PRON	VBG|NN	VERB
,	.	JJS	ADJ	PRP|VBP	PRON	VBN	VERB
-LRB-	.	JJ|RB	ADJ	PRT	PRT	VBP	VERB
-RRB-	.	JJ|VBG	ADJ	RB	ADV	VBP|TO	VERB
.	.	LS	X	RBR	ADV	VBZ	VERB
:	.	MD	VERB	RBS	ADV	VP	VERB
?	.	NN	NOUN	RB|RP	ADV	WDT	DET
CC	CONJ	NNP	NOUN	RB|VBG	ADV	WH	X
CD	NUM	NNPS	NOUN	RN	X	WP	PRON
CD|RB	X	NNS	NOUN	RP	PRT	WP$	PRON
DT	DET	NN|NNS	NOUN	SYM	X	WRB	ADV
EX	DET	NN|SYM	NOUN	TO	PRT	``	.

Table E.2: Mapping of Penn Treebank tags to universal part-of-speech tags.

Tag	Description	Universal
_A_	Adjective	ADJ
_D_	Adverb	ADV
_G_	GenitiveSingularPositiveAdjective	X
_I_	Interjection	PRT
_J_	Conjunction	CONJ
_K_	Adposition	ADP
_N_	Numeral	NUM
_P_	Pronoun	PRON
_S_	Noun	NOUN
_T_	Foreign	X
_V_	Verb	VERB
_X_	Adverb	ADV
_Y_	Abbreviation	X
_Z_	Punctuation	PUNCT

Table E.3: Estonian corpus part-of-speech tags and mapping to universal tags.

APPENDIX F

Skrefsrud's Santhal Grammar

One of the texts downloaded and used extensively in our development is *A grammar of the Santhal language* by L. O. Skrefsrud, published in 1873 and digitized into the University of Michigan Digital General Collection in 2006 (available at: `http://name.umdl.umich.edu/AAU0434.0001.001`). This text is a typical example of a grammar in our collection.

From this text, pages 3, 30, 33, 34, and 35 were annotated by multiple annotators, allowing us to measure inter-annotator agreement. From these annotations, we created a gold set "best" annotations, which were used for training the language ID models.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abney, Steven. 2011. Data-intensive experimental linguistics. *Linguistic Issues in Language Technology* 6.

Abney, Steven and Steven Bird. 2010. The human language project: Building a universal corpus of the world's languages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 88–97. Uppsala, Sweden.

Avramidis, Eleftherios and Philipp Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 763–770. Columbus, OH.

Berg-Kirkpatrick, Taylor and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297. Uppsala, Sweden.

Berkowitz, Luci and Karl A. Squitier. 1986. *Thesaurus Linguae Graecae: Canon of Greek Authors And Works*. New York, NY: Oxford University Press, 2nd edition.

Bird, Steven. 2009. Natural language processing and linguistic fieldwork. *Computational Linguistics* 35:469–474.

Bow, Cathy, Baden Hughes, and Steven Bird. 2003. Towards a general model of interlinear text. In *Proceedings of the EMELD Workshop on Digitizing and Annotating Texts and Field Recordings*. LSA Institute, Michigan State University.

Brown, Peter F., John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, Robert L. Mercer, and Paul S. Roossin. 1988. A statistical approach to language translation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 71–76. Budapest, Hungary.

Carletta, Jean. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics* 22:249–254.

Creutz, Mathias and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. *Publications in Computer and Information Science* Report A81.

Daumé III, Hal and Lyle Campbell. 2007. A Bayesian model for discovering typological implications. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 65–72. Prague, Czech Republic.

de Marcken, Carl. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, MIT.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Dimitrova, Ludmila, Tomaz Erjavec, Nancy Ide, Heiki Jaan Kaalep, Vladimir Petkevic, and Dan Tufis. 1998. Multext-East: Parallel and comparable corpora and lexicons for six Central and Eastern European languages. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics*, pages 315–319. San Francisco, CA.

Dryer, Matthew S. 2011. Prefixing vs. suffixing in inflectional morphology. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Munich: Max Planck Digital Library.

Durgar-El-Kahlout, İlknur and Kemal Oflazer. 2006. Initial explorations in English to Turkish statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 7–14. New York, NY.

Francis, W. Nelson and Henry Kučera. 1979. *Manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers*. Brown University, revised and amplified edition.

Gale, William A. and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics* 19:177–184.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27:153–198.

Grenager, Trond, Dan Klein, and Christopher D. Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 371–378. Morristown, NJ.

Grünwald, Peter D. 2005. Minimum description length tutorial. In Peter D. Grünwald, In Jae Myung, and Mark A. Pitt, editors, *Advances in Minimum Description Length*, chapter 2, pages 22–80. MIT Press.

Hammarström, Harald. 2009. *Unsupervised Learning of Morphology and the Languages of the World*. Ph.D. thesis, Chalmers University of Technology and University of Gothenburg.

Harris, Zellig S. 1951. *Structural Linguistics*. Chicago: The University of Chicago Press.

Iwata, Tomoharu, Daichi Mochihashi, and Hiroshi Sawada. 2010. Learning common grammar from multilingual corpus. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 184–188. Uppsala, Sweden.

Karttunen, Lauri. 1983. Kimmo: A general morphological processor. *Texas Linguistic Forum* 22:165–186.

Katamba, Francis and John Stonham. 2006. *Morphology*. Houndmills: Palgrave Macmillan, second edition.

Kiss, Tibor and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics* 32:485–525.

Klein, Dan. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Klein, Dan and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Klein, Dan and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language processing. *Advances in Neural Information Processing Systems 15 (NIPS)* .

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876.

Kruengkrai, Canasai, Prapass Srichaivattana, Virach Sornlertlamvanich, and Hitoshi Isahara. 2006. Language identification based on string kernels. In *Proceedings of the First International Conference on Knowledge, Information and Creativity Support Systems*. Ayutthaya, Thailand.

Kurimo, Mikko, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95. Uppsala, Sweden: Association for Computational Linguistics.

Lewis, M. Paul, editor. 2009. *Ethnologue: Languages of the World*. Online version: `http://www.ethnologue.com/`. Dallas, TX: SIL International, sixteenth edition.

Lewis, William D. and Fei Xia. 2009. Parsing, projecting & prototypes: repurposing linguistic data on the web. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 41–44. Morristown, NJ.

Lewis, William D. and Fei Xia. 2010. Developing ODIN: A multilingual repository of annotated langauge data for hundreds of the world's languages. *Literary and Linguistic Computing* .

Li, Peng, Maosong Sun, and Ping Xue. 2010. Fast-Champollion: A fast and robust sentence alignment algorithm. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters Volume*, pages 710–718.

Ma, Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Mansuri, Imran R. and Sunita Sarawagi. 2006. Integrating unstructured data into relational databases. In *Proceedings of the 22nd International Conference on Data Engineering*, pages 29–41. Washington, DC: IEEE Computer Society.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19:313–330.

Maxwell, Mike and Baden Hughes. 2006. Frontiers in linguistic annotation for lower-density languages. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pages 29–37. Morristown, NJ.

Minkov, Einat, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 128–135. Prague, Czech Republic.

Moon, Taesun and Katrin Erk. 2008. Minimally supervised lemmatization scheme induction through bilingual parallel corpora. In *Proceedings of the International Conference on Global Interoperability for Language Resources*, pages 179–186.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, pages 135–144. Tiburon, CA: Springer.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29:19–51.

Palmer, Alexis, Taesun Moon, Jason Baldridge, Katrin Erk, Eric Campbell, and Telma Can. 2010. Computational strategies for reducing annotation effort in language documentation: A case study in creating interlinear texts for Uspanteko. *Linguistic Issues in Language Technology* 3.

Petrov, Slav, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*.

Resnik, Philip and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics* 29:349–380.

Rissanen, Jorma. 1989. *Stochastic complexity in statistical inquiry*. Singapore: World Scientific.

Saffran, Jenny R., Richard N. Aslin, and Elissa L. Newport. 1996. Statistical learning by 8-month-old-infants. *Science* 274:1926–1928.

Scannell, Kevin P. 2007. The Crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*. Louvain-la-Neuve, Belgium.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK.

Sereewattana, Siriwan. 2003. *Unsupervised Segmentation for Statistical Machine Translation*. Master's thesis, School of Informatics, University of Edinburgh.

Sharif-Razavian, Narges and Stephan Vogel. 2010. Fixed length word suffix for factored statistical machine translation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 147–150. Uppsala, Sweden.

Shi, Lei and Ming Zhou. 2008. Improved sentence alignment on parallel web pages using a stochastic tree alignment model. In *Proceedings of the 2008 Conference on Empircal Methods in Natural Language Processing*, pages 505–513.

Snyder, Benjamin and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 737–745. Columbus, OH.

Varga, Dániel, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of Recent Advances in Natural Language Processing 2005*, pages 590–596. Borovets, Bulgaria.

Virpioja, Sami, Jaakko J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of Machine Translation Summit XI*, pages 491–498. Copenhagen, Denmark.

Xia, Fei, William Lewis, and Hoifung Poon. 2009. Language ID in the context of harvesting language data off the web. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 870–878. Athens, Greece.

Xia, Fei and William D. Lewis. 2007. Multilingual structural projection across interlinearized text. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, NY.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8. Morristown, NJ.

Yeniterzi, Reyyan and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 454–464. Uppsala, Sweden.