Modeling Inflectional Complexity in Natural Language Processing

by

Garrett Nicolai

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Inflectional morphology presents numerous problems for traditional computational models, not least of which is an increase in the number of rare types in any corpus. Although few annotated corpora exist for morphologically complex languages, it is possible for lay-speakers of the language to generate data such as inflection tables that describe patterns that can be learned by machine learning algorithms.

We investigate four inflectional tasks: inflection generation, stemming, lemmatization, and morphological analysis, and demonstrate that each of these tasks can be accurately modeled using sequential string transduction methods. Furthermore, expert annotation is unnecessary: inflectional models are learned from crowd-sourced inflection tables.

We first investigate inflection generation: given a dictionary form and a tag representing inflectional information, we produce inflected word-forms. We then refine our predictions by referring to the other forms within a paradigm. Results of experiments on six diverse languages with varying amounts of training data demonstrate that our approach improves the state of the art in terms of predicting inflected word-forms.

We next investigate stemming: the removal of inflectional prefixes and suffixes from a word. Unlike the inflection generation task, it is not possible to use inflection tables to learn a fully-supervised stemming model; however, we exploit paradigmatic regularity to identify stems in an unsupervised manner with over 85% accuracy. Experiments on English, Dutch, and German show that our stemmers substantially outperform rule-based and unsupervised stemmers such as Snowball and Morfessor, and approach the accuracy of a fully-supervised system. Furthermore, the generated stems are more con-

sistent than those annotated by experts. We also use the inflection tables to learn models that generate lemmas from inflected forms. Unlike stemming, lemmatization restores orthographic changes that have occurred during inflection. These models are more accurate than Morfette and Lemming on most datasets.

Finally, we extend our lemmatization methods to produce complete morphological analyses: given a word, return a set of lemma / tag pairs that may have generated it. This task is more ambiguous than inflectional generation or lemmatization which typically produce only a small number of outputs. Thus, morphological analysis involves producing a complete list of lemma+tag analyses for a given word-form. Experiments on four languages demonstrate that our system has much higher coverage than a hand-engineered FST analyzer, and is more accurate than a state-of-the-art morphological tagger.

Preface

The work presented in this dissertation was previously published at computational linguistic conferences and workshops, but modified here to fit the format of the thesis. The author served as primary author on all major work reported here, and was responsible for implementation, experimentation, and analysis. Writing was done in conjunction with the other listed authors.

Chapter 4 was published as Nicolai et al., 2015a)

Chapter 5 was published as Nicolai and Kondrak, 2016).

Chapter 6 was published as Nicolai and Kondrak, 2017).

Chapter 7 was published as Nicolai et al., 2016a). The author was the team lead, as well as responsible for experiments in German, Spanish, Finnish, and Turkish.

You keep using that word. I do not think it means what you think it means.

– Inigo Montoya, *The Princess Bride.*

Caesar non supra grammaticos.

– attributed to the Council of Constance, when Holy Roman Emperor Sigismund inflected a word incorrectly

Acknowledgements

They say that it takes a village to raise a baby, and it's no less true of a PhD. There are many people who helped me get here, and I cannot possibly thank you enough for the support you've provided.

First, thank you, Greg. You went above and beyond the call of duty in ensuring my growth as a researcher. Beyond just providing advice and steering me away from some of my crazier ideas, you gave me the confidence to investigate some really interesting problems, and I'm a much different person than I was when I started.

Secondly, thank you Colin and David. Your comments made my thesis better, and our discussions helped me focus my research.

Thirdly, I would be remiss in neglecting to acknowledge my funding sources at NSERC, AITF, and the University itself. You let me travel to conferences, live near campus, and eat more than Kraft dinner. My mental state-of-mind owes you its thanks.

Thank you to the NLP group: Mohammad, Lei, Bradley, Adam, Ying, Kevin, Simon, Mohammad, and Saeed. I loved the Shared Tasks!

Thank you to neural networks. I'm not quite sure what they did, but they have a hand in everything NLP these days, so I'm sure they did something behind the scenes.

This degree would not have been possible without my family. My parents, George and Karen; my brothers Ed and Dan, and my sister, Tori. You comforted me when I was depressed, shared in my accomplishments, and continue to provide unending love and support. I can't even begin to imagine where I would be without you.

Mike, Jody, Nick, Alex, and auntie Inez: while I've been in Edmonton, you

guys have been my Alberta family. I'm going to miss you guys.

Kyle, when I was searching for grad schools, you convinced me that the U of A would be my best option. I agree whole-heartedly.

Je me souviens d'une classe dans la douxième année. Mme. Belzil nous a demandé pourquoi nous prenions le français. Nous avions tous des réponses typiques: "pour connâitre un deuxième langue"; "pour obentir emploi"; "nous sommes canadiens!"; "pour la santé mentale". Après quelques minutes, Mme. Belzil nous a demandé, plaintivement, s'il n'y avait personne qui aimait la langue. Oui, madame, j'en aime bien.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Like many European languages, the English of a thousand years ago was a highly inflected language: nouns used case to indicate if they were the subject or object of a sentence, adjectives marked the gender and role of the nouns they were modifying, and verbs needed to distinguish between many different tenses and aspects. Consider the first lines from *Beowulf* (Garnett, 1912), given below:

Hwæt! We Gar-Dena in gear-dagum	Lo! We of the Spear-Danes', in days of yore,
þeod-cyninga, þrym gefrunon,	Warrior-kings' glory have heard
hu ðа æþelingas ellen fremedon!	How the princes heroic deeds wrought.
Oft Scyld Scefing sceaþena þreatum	Oft Scyld, son of Scef, from hosts of foes,
monegum mægþum meodo-setla ofteah;	From many tribes, their mead-seats took;

Note that the plural marker is not simply an *-s*, as in modern English, but also marks case: *Gar-Dena* 'Spear-Danes' and *cyninga* 'kings' are inflected with a genitive plural *-a* suffix in lines 1 and 2, as is *meodo-setla*:"mead-seats" in line 5, while *æþelingas* 'princes' is inflected with a nominative plural *-as*. The dative plural *-um* is evident in *dagum* 'days', *þreatum* 'tribes', and *mægþum* 'foes', as well as the modifying adjective *monegum* 'many'. Similarly, Scyld inflects in the 3^{rd} person preterite singular *-ah*: *ofteah* 'took', but the *We* 'we' of line 1, enforce a a 1^{st} person plural preterite *-on* on the verb *gefrunon* 'heard', while the *æþelingas* of line 3 enforces the 3^{rd} person plural preterite *-on* on *fremedon* 'wrought'.

As the language evolved, information that had previously been indicated through the shapes of words began to be replaced by other linguistic cues: nominal cases began to be represented by a strict word order and the increased use of prepositions; verbal tenses that were previously marked by suffixes began to be constructed with auxiliary verbs; adjectival inflection almost completely disappeared. What remains today is typically noted for being inflectionally poor, maintaining only a small subset of the morphology that marked its ancestors.

The same cannot be said for many other modern languages: German marks nouns for 4 different cases; Finnish marks over a dozen. French and Spanish have over fifty different verb forms, depending on tense, mood, aspect, and other features. This large number of types negatively impacts the results of computational methods that rely on the construction of representative statistical models: a Finnish text of billions of words is not sufficiently large to include all of the possible forms of any single word. Conversely, a Finnish speaker may never use a great proportion of the possible forms for any given word, but is able to construct or decipher them, if the need arises.

1.1 Morphology

Broadly speaking, *morphology* is the sub-discipline of linguistics that deals with the systematicity in the relationship between the form and meaning of words (Booij, 2012). *Morpheme-based morphology* posits that words are made up of *morphemes*: the minimal linguistic units with a lexical or a grammatical meaning. It is typically divided into two sub-categories: *derivational morphology* and *inflectional morphology*.

Derivational morphology is a process by which new words are created, forming complex words by means of affixation or non-concatenative morphology, and often alters the part-of-speech of the root word. For example, the adjective *quick* can become the adverb *quickly*: "performed in a quick manner" through the addition of *-ly*. Both *quick* and *quickly* will have entries in the dictionary. Although derivational morphology presents its own set of issues

for computational methods, in this dissertation, we are strictly concerned with inflection.

Inflectional morphology, on the other hand, does not create new *lexemes*. Rather, inflection creates variant instantiations of the same lexeme that often demonstrate grammatical categories such as aspect, mood, tense, etc. Under the morpheme-based paradigm, inflection involves modifying a prototypical word form called the lemma with morphemes that contain grammatical information. Unlike derivation, inflection never changes the part-of-speech of its lemma: *eat*, *eats*, *ate*, *eating*, and *eaten* all describe the act of consuming food; they only differ in whether the act has been completed and who has been doing the consuming.

Inflection is often realized through *affixation*, where inflectional morphemes are bound to the stem. These morphemes do not occur on their own, and as such, are referred to as *bound* morphemes. *Prefixation* binds the morpheme before the stem; *suffixation* attaches the morpheme after the stem; *infixation* interrupts the contiguity of the stem, and *circumfixation* is a combination of prefixation and suffixation. Inflection can also be realized through stem alteration, which modifies the stem itself (cf. *goose* $\rightarrow$ *geese*). The languages in this dissertation largely realize inflection through suffixation, although they also make use of circumfixation and stem alteration.

Throughout this paper, we refer to the lemma as the prototypical form of an inflection table, and describe our algorithms as modifying the lemma, instead of the stem. This may not be linguistically accurate, but within the context of the character-based algorithms we implement, it is an accurate representation of the processes being performed. Realistically, we are inflecting the citation form, which happens to be the lemma for all of the languages we are investigating. The lemma provides a consistent form that we can leverage to learn inflectional patterns in a supervised manner.

infinitive		atmen
present participle		atmend
past participle		geatmet
auxiliary		haben
	indicative	
present	ich atme	wir **atmen**
	du atmest	ihr atmet
	er atmet	sie **atmen**
preterite	ich atmete	wir atmeten
	du atmetest	ihr atmetet
	er atmete	sie atmeten

Figure 1.1: A partial inflection table for the German verb *atmen* "to breathe".

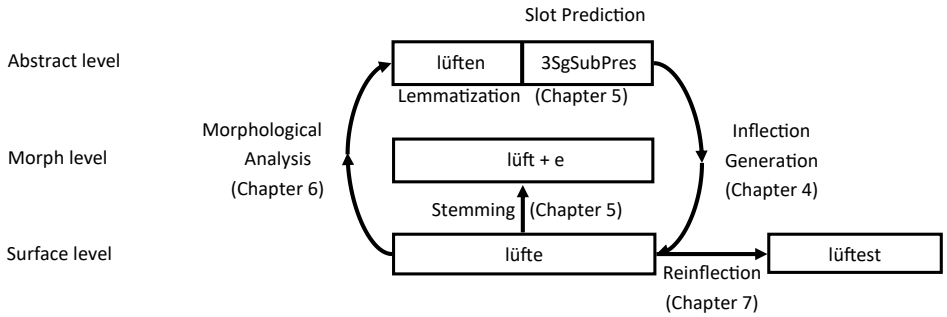

Figure 1.2: The inflectional tasks described in this dissertation.

1.2 Sequential Transduction of Inflection Tables

We propose that inflectional morphology can be effectively modeled as sequential string transduction. Sequential transduction is a deterministic process whereby each character in an input sequence is transformed into an output character via a set of transduction rules. These rules can be learned automatically from data, using computational methods for sequential prediction. Furthermore, we propose that expert annotation is not required to learn inflectional models. Rather, all necessary information is present, or can be inferred, from publicly available inflection tables. These tables can be constructed by moderately-skilled speakers of a language, without the need to consult experts in morphology. We present a sample inflection table in Figure 1.1.

Figure 1.2 outlines the inflectional tasks that we model in this dissertation. If we consider a word as being representable on three separate inflectional

levels, then inflection is the process of moving between these levels. On the surface level, a word appears as it does in text. The morph level represents a word as a sequence of *morphs*: orthographic representations of the stem and affixes. On the abstract level, a word is represented by its lemma and a number of inflectional features. *Inflection generation* is the transformation from the abstract level to the surface level. Inflection generation is unambiguous: except in rare cases, a form on the abstract level will only produce a single form on the surface level. *Re-inflection* is similar to inflection generation, but instead of forcing inflectional features to apply to a lemma, it applies the features to an already inflected form, essentially replacing one set of inflectional features with another one.

The transformation from the surface to the morph level is called *inflectional segmentation*, and subsumes the task of *stemming*. Transforming a word from the surface level to the abstract level is called *morphological analysis*. Morphological analysis consists of two equally important sub-tasks: lemmatization, which restores inflectional changes made to the stem of a word, and slot prediction, which analyzes the inflectional information present in the surface form to identify the inflectional features indicated by the word. Because of the process of *syncretism*, a single surface form may have multiple correct analyses; for example, the form *played* may be either a preterite form, a past participle, or an adjectival form (i.e., a well-played game). Stemming and morphological analysis fall under the umbrella of *inflectional simplification*, which reduces the large number of surface forms to a smaller number of representative forms.

1.3 Summary

We model inflectional morphology as a specific application of sequential string transduction. We model each task in Figure 1.2.

In Chapter 2, we describe the various tools and resources we use to model inflectional morphology.

In Chapter 3, we report related work in inflection, and describe how our work approaches the task differently.

In Chapter 4, we describe our process for modeling inflectional generation, which corresponds with a transformation from the abstract level to the surface level in Figure 1.2. We achieve state-of-the art results, even when restricting the training data to low-resource situations.

In Chapter 5, we reverse the process from Chapter 4, and tackle inflectional simplification; namely, stemming and lemmatization. While continuing to show the suitability of sequential transduction as a method for modeling inflection, we also show that inflection tables are sufficient to learn high-quality stemming annotations, despite the lack of explicit information in the tables.

Chapter 6 extends the work of Chapter 5, modifying the methods to produce full morphological analyses. Unlike previous work, we show that accurate analysis sets can be obtained without expert annotation. We also approach the accuracy of a hand-constructed analyzer, but with much higher coverage.

In Chapter 7, we provide an extrinsic evaluation of our lemmatization and generation methods, presenting our results for the First Shared Task on Morphological Reinflection. We confirm early results in the low-resource setting, but demonstrate that our methods are suitable not only for fusional languages, but for agglutinative and templatic languages as well.

Chapter 8 concludes the dissertation.

Chapter 2

Tools of the Trade

We approach inflection as a specialized form of sequential string transduction. Generally speaking, inflection occurs as one of two processes within a word: either the addition of meaning carrying *affixes*, or as the changing of a root form of a word to represent a desired inflection. For example, English is a largely suffixing language, where inflection is indicated by the addition of inflectional morphemes to the end of a word, such as the plural morpheme *-s*; however, English also occasionally represents inflectional properties through stem changes, such as the past tense of *swim*: "swam".

String transduction is the process of converting one set of characters to another, using contextual cues to establish transduction rules. Characters can be converted via three common operations: insertion, whereby new characters are added to the input sequence to arrive at the output; deletion, where characters in the input are removed from the sequence, and substitution, which can be viewed as a deletion followed by an insertion.

We are not the first to propose that inflection be modeled as transduction: the prevailing methodology of hand-built morphological analyzers is finite-state transducers. These transducers are built upon many contextually-conditioned rules that determine inflectional operations.

2.1 Character Alignment

The training of a transduction model requires a set of aligned source and target strings, which are often of different lengths. Alignment ensures that for each

character in the source string, there is a corresponding character in the target string. Deletion of characters in the source string is easily handled, as the operation is anchored by the sequence being deleted and its context.

However, insertion operations quickly lead to prohibitively expensive transduction models, and are not allowed by our transducer. We mimic insertion by allowing a many-to-many alignment between source and target: several characters in the source can be aligned to one or more characters in the target. The aligned source-target pairs are then given to the transducer as a training set, and atomic character transformations are extracted from these alignments.

Figure 2.1 demonstrates an example alignment between a form on the abstract level, and one on the surface level. Deletion of the lemma-final 'e' is represented with an alignment to an underscore (representing null). Conversely, the insertion of the second 't' on the surface level is accomplished via a one-to-many alignment. The morphological tag, along with a morphemic boundary marker, is aligned to the affix. Individual characters are separated by a space; the morphological tag is treated as a single atomic character.

w	r	i	t	e	+ PastPart
w	r	i	t t	_	e n

Figure 2.1: An aligned source-target pair for the past participle of *write*.

We infer the alignment with a modified version of the M2M aligner of Jiampojamarn et al., 2007). The program applies the Expectation-Maximization algorithm with the objective to maximize the joint likelihood of its aligned source and target pairs. The source and target pairs are task-specific, and described more fully in the respective chapters, but we make several modifications to the core alignment algorithm to be able to handle inflectional morphology.

M2M makes use of an extension of the forward-backward training of a one-to-one stochastic transducer (Ristad and Yianilos, 1998). During the expectation step, the forward part of the algorithm calculates the sums of all

left-to-right paths through the transducer that generate an aligned source-target sequence pair x^s, y^t. Likewise, the backward part of the algorithm calculates the sums of the right-to-left paths through the transducer. An initially uniform probability table is used to transform these sums into partial counts. The maximization step simply normalizes the partial counts by the number of paths that generate the entire sequence. The algorithm proceeds until convergence. Once probabilities are learned, the Viterbi algorithm can be applied to produce the most likely alignment.

In our systems, affixation is modeled as a transduction operation between the word-form of the affix (e.g., *'-s'*), and an atomic morphological tag representing the desired inflectional category (e.g., PLURAL). While stem changes within a word are often confined to sequences of two or fewer characters, affixes often require longer sequences, such as the French suffix for the 3^{rd} Person Plural Conditional Present: *-eraient*.

Representing inflectional classes within M2M requires no modification, as it is capable of accepting multi-character strings as atomic units. However, a modification was required to allow differing alignment lengths between affixes and other characters[1]. Preliminary experiments were conducted that simply increased the maximum alignment length, however it was discovered that this led to a decrease in performance during transduction, as it requires our system to learn transduction rules that are far too precise.

Instead, we mark all inflectional tags with a special marker, and modify the expectation step of M2M. If a character does not contain the tag marker, then M2M proceeds as before. However, if the marker is present on the source side of the alignment, the expectation step is modified to allow an alignment to consist of an alternative maximum number of characters on the target side. Some tasks require the morphological tag to appear on the target side; for these tasks, we simply align the data with the tag on the source side, and reverse the source and target after alignment.

Furthermore, in order to encourage alignments between identical charac-

[1]The modified versions of M2M and DIRECTL+ are available at https://github.com/GarrettNicolai

ters, we modify the aligner to generalize all identity transformations into a single match operation. This feature can be activated via a run-time parameter. After source-target pairs have been aligned, they can be used to train a transducer.

2.2 Transduction

We perform string transduction by adapting DirecTL+, a tool originally designed for grapheme-to-phoneme conversion (Jiampojamarn et al., 2010). DirecTL+ is a feature-rich, discriminative character transducer that searches for a model-optimal sequence of character transformation rules for its input. The core of the engine is a dynamic programming algorithm capable of transducing many consecutive characters in a single operation. Using a structured version of the MIRA algorithm (McDonald et al., 2005), training attempts to assign weights to each feature so that its linear model separates the gold-standard derivation from all others in its search space.

DirecTL+ follows the online training paradigm: during the training step, each instance is evaluated in turn using the current weights. Loss can then be calculated and the weights can be adjusted to make the model favor the correct target over incorrect ones.

At each update, DirecTL+ produces an n-best list of possible targets. The Margin Infused Relaxed Algorithm (MIRA) attempts to update the weights of the model in such a way that the new weights will separate the correct target from each of the incorrect predictions in the n-best list, within an acceptable margin. The loss function is binary: 0 if the target prediction is correct, and 1 otherwise. SVM^{light} is used to approximate the hard margin required by MIRA, and is used to learn an optimum weight update.

The derivation is dependent upon the training alignment: rather than treat each character in a source or target word independently, DirecTL+ learns features on character sequences corresponding to an aligned segment. Only those features that exactly match the focus sequence are allowed to produce a given derivation.

DIRECTL+ uses a number of feature templates to assess the quality of a transduction operation: source context, target n-gram, and joint n-gram features. Context features conjoin the operation with indicators for all source character n-grams within a fixed window of where the operation is being applied. Target n-grams provide indicators on target character sequences, describing the shape of the target as it is being produced, and may also be conjoined with our source context features. Joint n-grams build indicators on operation sequences, combining source and target context, and memorizing frequently-used rule patterns.

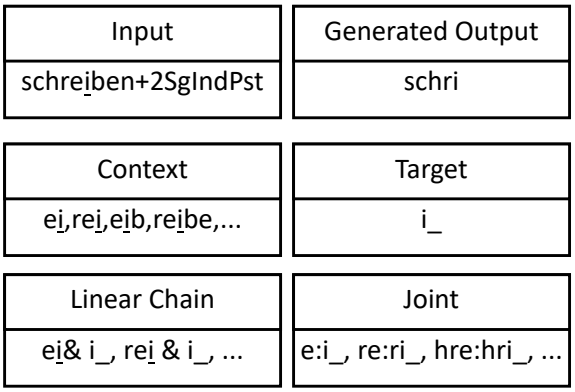

Figure 2.2: An example generation.

An example of the generation process is given in Figure 2.2. The generation proceeds character by character, applying the operations that fit the given context. In this example, the correct output is *schriebst*, and DIRECTL+ has currently processed every source character up to 'e', and is now focused on 'i'. Likewise, every target character up to 'i' has been generated, and DIRECTL+ must transform the 'i' of the lemma into an 'e'. With this focus, the context features look at every n-gram up to length x that apply focus on the segment 'i' on the source side. For example, if $x = 3$, context unigrams, bigrams, and trigrams around the focus character will be used to select appropriate context features. Target features are only applied if they contain an 'i' as a previously generated character. Linear-chain features combine the context and target features, i.e., a linear chain rule must satisfy an appropriate source context

```
1: for each word w in set W do
2:     for each prediction p in n-best list L do
3:         F_p = Extract_Features(p)
4:     for each prediction i in L do
5:         if Training then
6:             for each prediction j != i do
7:                 if i == Gold(w) then
8:                     Class = 1 //i should be ranked > j
9:                 else if j == Gold(w) then
10:                    Class = -1 //j should be ranked > i
11:                else
12:                    continue
13:                RerankInstance = F_i - F_j
14:                Write Class and RerankInstance to TrainFile
15:        else
16:            Write F_i to TestFile
```

Figure 2.3: Algorithm to frame reranking as a classification problem.

and target context. Joint rules look at the aligned source and target to the left of the focus character. Since the alignment is unknown at test time, all reasonable alignments are considered; in this example, it is possible that 'ei' will be a single focus sequence, and thus a different series of rules will apply.

Following Toutanova and Cherry, 2009), we modify the out-of-the-box version of DIRECTL+ by implementing an abstract copy feature that indicates when a rule simply copies its source characters into the target, e.g. $b \rightarrow b$. The copy feature has the effect of biasing the transducer towards preserving the source characters during transduction, and is analogous to the one described in Section 2.1.

2.3 Reranking

Alignment followed by transduction is suitable for learning sequential transduction rules, but DIRECTL+ has no method to incorporate observation statistics into its model. While incorporating such a feature into DIRECTL+ would be useful, it is not clear how to implement word-level features on a character-level transduction model. Furthermore, many of these features are task-dependent, and would need to be re-engineered for each task and lan-

guage, forcing a significant investment for the development of features.

Instead, we make use of the more flexible paradigm of n-best reranking. Under this paradigm, a system first generates a list of prediction *candidates*, which are then re-scored according to separate criteria. DIRECTL+ already provides functionality to output an n-best list, along with confidence scores. This list can then be reranked using features unrelated to the transduction model itself.

At its core, our reranker makes use of a Support Vector Machine (SVM) to classify instances into one of two classes: rerank or not. We rerank according to the method of Joachims, 2002), which converts the reranking problem into a classification one. This process is outlined in Figure 2.3. After generating the n-best list with DIRECTL+, we transform each prediction into a feature vector in lines 2 and 3. This step is often task-dependent, and we leave the details of feature selection to the individual chapters. The transformation from a ranking problem to a classification one occurs in line 13. When creating the training file for the reranker, we compare each pair of predictions from the n-best list, and learn to classify *the difference* between the two. This trick encourages the model to reward features that occur in instances that should be highly ranked, while punishing instances that should be ranked lower. At test time, we no longer classify differences, but instances: we then rank the n-best list based on the scores produced by the SVM.

2.4 CELEX

Many of the morphological tasks that we address in this dissertation traditionally require the careful construction of annotated data sources by experts with years of study in morphological processes. Supervised machine learning algorithms can generalize over new forms, but still require training data to be annotated for morphological phenomena such as stemming, analysis, and lemmatization.

One such resource is CELEX (Baayen et al., 1995). CELEX is a lexicon for English, German, and Dutch, and provides linguistic annotation in several cat-

egories. Morphologically, CELEX provides both derivational and inflectional annotation, and thus provides a high-quality gold-standard for morphological tasks.

The CELEX lexicon covers the full inflection paradigms for more than 50,000 lemmas for English and German, and 120,000 for Dutch, accounting for more than 160,000, 365,000 and 380,000 word-forms, respectively. Along with morphological information, CELEX also provides pronunciation and hyphenation data, orthographic variants, and corpus frequency statistics. Figure 2.1 gives an example of the type of inflectional information stored in CELEX.

Access to this lexicon strongly dictated the languages we worked upon in this dissertation. While we propose that high-quality tools for inflection can be trained on crowd-sourced inflection tables, we required a high-confidence set for the evaluation of our methods. Furthermore, while inflection tables provide lemmatization and analysis information, they do not explicitly provide stems or segmentations; a lexicon such as CELEX is still required for the evaluation of these tasks.

The data in Figure 2.1 illustrates two inflectional issues: first, of the seven different surface forms for *lüften*: "to ventilate", only one was observed more than 5 times in a corpus of more than 5 million tokens. Secondly, if capitalization is ignored, two forms from a completely different lemma (*Luft*: "air") can be confused with inflected forms of *lüften*; this is of particular concern for morphological analysis.

Unfortunately, access to morphological lexicons is limited. Although they are of inestimable value to researchers, they are very expensive to create. Their construction requires an immense effort on the part of a limited set of individuals with the knowledge required to build them. As a result, these lexicons are only available for a small number of languages. Furthermore, although they can be periodically updated, they often suffer from limited coverage. These limitations motivate our search for other morphological sources.

TypeID	Surface Form	Count	Lemma	Morphological Tags
21808	lüfte	3	lüften	1SIE, 13SKE, rS
21809	lüftest	0	lüften	2SIE, 2SKE
21810	lüften	4	lüften	13PIE, 13PKE, i
21811	lüftete	4	lüften	13SIA, 13SKA
21812	lüfteten	1	lüften	13PIA, 13PKA
21813	lüftetet	0	lüften	2PIA, 2PKA
21814	gelüftet	6	lüften	pA
308264	Lüfte	18	Luft	nP, gP, aP
308265	Lüften	5	Luft	dP

Table 2.1: The representation of the forms of lüften, as observed in a morphological lexicon.

2.5 Wiktionary

We claim that expert-annotated data is not required for the training of high-quality inflection tools. To justify this claim, we perform many of our experiments on data acquired from the crowd-sourced Wiktionary (www.wiktionary.org).

Wiktionary follows the typical wiki process of data collection, in that a commonly accessible internet database is modifiable by a large number of anonymous users. As with any crowd-sourced data, Wiktionary has the potential to be noisy, however in practice tends to provide relatively reliable information.

Although its name and project statement both imply that Wiktionary is an online dictionary, it often provides information that is not present in a traditional dictionary, such as inflectional information. Many words are accompanied by complete inflection tables, and many others are accompanied by partially completed tables. The inflection table shown in Chapter 1 was taken from Wiktionary. Projects such as Unimorph (www.unimorph.org) have been active in collecting these tables and adapting them to a format suitable for research.

We distinguish Wiktionary data from CELEX data by noting that it is non-expert data. By non-expert, we mean that the contributors to Wiktionary typically do not need to be trained in morphology to complete the inflection

tables contained on the website: any lay-person with a moderate knowledge of a language can contribute to the inflection tables on Wiktionary.

Chapter 3

Related Work

We are hardly the first to recognize the need for computational processing of inflection, and recently, there has been a significant amount of work that tries to model inflection, particularly inflection generation and morphological analysis.

3.1 Stemming

Stemming is a sub-task of the larger problem of morphological segmentation. Because of the scarcity of morphologically-annotated data, many segmentation algorithms are unsupervised or rule-based.

The Porter stemmer (Porter, 1980) and its derivatives, such as Snowball (snowballstem.org), apply hand-crafted context rules to strip affixes from a word. Creation of such rule-based programs requires significant effort and expert knowledge. We use structured inflection tables to create training data for a discriminative transducer.

Linguistica (Goldsmith, 2001) and Morfessor (Creutz and Lagus, 2002) are unsupervised word segmenters, which divide words into regularly occurring sub-sequences by applying the minimum description length (MDL) principle. While these methods are good at identifying common morphemes, they make no distinction between stems and affixes, and thus cannot be used for stemming. Morfessor Categories-MAP (Creutz and Lagus, 2004; Creutz and Lagus, 2005) distinguishes between stems and affixes, but not between derivational and inflectional affixes. We adapt a more recent version (Grönroos et al., 2014)

to be used as an approximate stemmer for comparison against our stemming methods. Our stemming method differs from the methods of Morfessor in the amount of structure present in the training data. The Morfessor stemmer is completely unsupervised, determining common morphemes from a lexicon. Our method does not require explicit stem annotations, but does require that the data be collected into inflection tables.

Poon et al., 2009) abandons the generative model of Morfessor for a log-linear model that predicts segmentations in sequence. The discriminative approach allows for the incorporation of several priors that minimize over-segmentation. Their unsupervised model outperforms Morfessor, and they are also able to report semi- and fully-supervised results. We also approach the problem using a discriminative method, but by aligning structured inflection tables, we can learn stemming annotations indirectly.

Ruokolainen et al., 2014) obtain further improvements by combining a structured perceptron CRF with letter successor variety (LSV), and the unsupervised features of Creutz and Lagus, 2004). Their system requires that the data be annotated with stem boundaries. While our system requires structured inflection tables, we are able to infer stem boundaries automatically.

Cotterell et al., 2015) introduce Chipmunk, a fully-supervised system for labeled morphological segmentation. Extending the sequence-prediction models, Chipmunk makes use of data that is annotated not only for stem or affix, but also for inflectional role, in a task that the authors refer to as *labeled morphological segmentation*. While highly accurate, Chipmunk is limited in that it requires data that is fully-annotated for both segmentation and inflection. Our system has access to the morphological tags in inflection tables, but segmentation and tag alignment are performed in an unsupervised way.

3.2 Morphological Analysis

Unlike stemmers, which can be unsupervised, morphological analyzers and lemmatizers typically require annotated training data, although like stemmers, rule-based systems also exist.

Traditionally, morphological analysis has been performed by hand-crafted Finite State Transducers (FSTs), such as Morphisto (Zielinski and Simon, 2009) and Omorfi (Pirinen, 2015). FSTs consist of weighted or unweighted transduction rules that provide lists of morphological analyses for given words, with no respect to context. They are also typically bidirectional: rules can be applied in reverse to perform inflection generation. We take significant motivation from FSTs; however, we look to remove the necessity of an expert in the construction of the transduction rules.

Whereas we concentrate on predicting analyses context-free, much of the work in automated morphological analysis provides a single analysis in running text. Toutanova and Cherry, 2009) also learn a joint model for morphological analysis from a morphologically annotated lexicon. Their joint model is very similar to our own, making use of a discriminative lemmatizer and then reranking the results with an unannotated corpus. However, where they have component tagger and lemmatizers, we make use of the transducer to also generate our tags. Without access to their code, it is difficult to make a direct comparison.

Fraser et al., 2012) use CRFs in the context of translation from English to German, but their method assumes the existence of a hand-crafted morphological generator. Unlike our work, which is concentrated on the generation of analyses, Fraser et al., 2012) frame the problem much more like a traditional part-of-speech tagging task: predicting tags based on the tags that have come before. Once the tags are acquired, the lemma and tags are presented to a morphological generator, which re-creates the word-form. Where our methods predict morphological analyses and inflections context-free, Fraser et al., 2012) makes use of the generated analyses to disambiguate in context, and serves as more of a complement to our methods.

Morfette (Chrupala et al., 2008) is a fully-supervised maximum entropy morphological analyzer, which includes a lemmatization module based on the Shortest Edit Script (SES). Unlike our system, Morfette performs analysis along a pipeline, first predicting a likely morphological tag, and then producing a lemma dependent upon the generated tag.

Like Morfette, Marmot (Mueller et al., 2013) and Lemming (Müller et al., 2015) predict morphological analyses in context through the use of a log-linear model; however, Marmot predicts tags and lemmas jointly. We are able to directly compare our method of morphological analysis against a modified, context-free version of Marmot. Furthermore, we investigate the utility of obtaining morphological analyses not from an expert lexicon, but rather from crowd-sourced inflection tables.

3.3 Inflection Generation

In some ways, inflection generation can be seen as the converse operation of morphological analysis, and thus many of the constraints of the latter task also apply to the former: inflection generation is usually a fully-supervised task. However, there is much less ambiguity in inflection generation than morphological analysis, and inflectional regularity means that some latent information can be used in the prediction of inflected forms.

Clifton and Sarkar, 2011) use Conditional Random Fields and a morpheme-based language model to predict the most likely final morpheme of each word in English-to-Finnish translation. Unlike our method for inflection generation, however, Clifton and Sarkar, 2011)'s method produces running text. Furthermore, rather than inflecting lemmas, their method instead learns to predict the correct allomorph from a set of options. This method is meant to capture variations such as consonant gradation and vowel harmony, which we model through target-side context.

Dreyer and Eisner, 2011) use a Dirichlet process mixture model and loopy belief propagation, seeded with a small number of complete inflection tables, to predict inflections of German verbs. They make use of unannotated corpora to aid the prediction task. Like our work, context n-grams are used to determine the appropriate verbal paradigm, but where the authors use Bayesian inference to predict word-forms, we view the problem as a string transduction task. Furthermore, whereas Dreyer and Eisner, 2011) only look at German verbs, we consider verbal inflection for five languages, as well as nominal inflection

for two.

Durrett and DeNero, 2013) align all inflected forms in a paradigm in order to extract string transformation rules that can then be applied to unseen lemmas. After alignment, rule sets are extracted that dictate the morphological changes that occur across an inflection table. Their factored model is similar to our DIRECTL+ model, as it learns rules without access to other forms in the paradigm. The joint model, like our reranked model, uses information from all forms in the paradigm when learning morphological rules.

Ahlberg et al., 2014) also align multiple forms within an inflection table; however, where Durrett & DeNero extract a rule for each changed character span, Ahlberg et al., 2014) replace each *unchanged* span with a variable, allowing rules to apply to an entire inflection table. Both of these methods differ from ours in that we do not enforce any table-level constraint: each rule applies only to a single inflection. By removing this constraint, we extend the flexibility of our rules at the cost of reduced interpretability.

Détrez and Ranta, 2012) focus on the task of determining the correct paradigm for a lemma on the basis of the inflected forms that were seen in training. This is similar to the task of a language learner identifying the correct inflection paradigm for a given word, and then applying the inflection rules of that paradigm to correctly inflect the word. Our work can be seen to mimic this behavior without the linguistic knowledge necessary to build explicit inflection paradigms. DirecTL+ must choose the correct paradigm for a word based on only one form. The reranker then provides further information to correct incorrect forms.

Eskander et al., 2013) construct paradigms from morphologically annotated text by attempting to fill gaps in the inflection tables. Making use of similar morphological operations across tables, they are able to learn *inflectional classes* that are applicable to many lemmas. This system requires a morphologically annotated corpus, from which it can then infer inflectional similarity across tables. Our system does not explicitly look for similarities across inflection tables, although this information is used to derive an alignment before transduction. While our task is significantly different than the

one in this paper, our system would be able to provide the lemma and stem information required by their method.

Similarly, Cotterell et al., 2017) is also concerned with the complete generation of inflection tables, which they refer to as *paradigm completion*. Using a generative neural model, they identify key inflections that are indicative of patterns across entire tables. Inspired by the linguistic theory of principal parts, their method then completes inflection tables through joint consideration of these key inflections. Although DirecTL+ is not capable of sharing information across inflection slots, our reranker takes advantage of paradigmatic regularity.

Neural models have also been adapted for inflection generation. Faruqui et al., 2016) employ a neural encoder-decoder model with attention to inflection generation, with results comparable to our best models from DirecTL+. Likewise, Kann and Schütze, 2016) make use of a bidirectional LSTM encoder-decoder, achieving the best results in the SIGMORPHON Shared Task on Reinflection (Cotterell et al., 2016).

Chapter 4

Inflection Generation as Discriminative String Transduction

Word-forms that correspond to the same lemma can be viewed as paradigmatically related instantiations of the lemma. For example, *take, takes, taking, took,* and *taken* are the word-forms of the lemma *take*. Many languages have complex morphology with dozens of different word-forms for any given lemma: verbs inflect for tense, aspect, person, etc.; nouns can vary depending on their role in a sentence, and adjectives can agree with the nouns that they modify and demonstrate comparative relationships. For such languages, many forms will not be attested even in a large corpus. However, different lemmas often exhibit the same inflectional patterns, called *paradigms*, which are based on morphological criteria. The paradigm of a given lemma can often be identified and applied to generate unseen forms.

Inflection prediction has the potential to improve Statistical Machine Translation (SMT) into morphologically complex languages. In order to address data sparsity in the training bi-text, Clifton and Sarkar, 2011) and Fraser et al., 2012) reduce diverse inflected forms in the target language into the corresponding lemmas. At test time, they predict an abstract inflection tag for each translated lemma, which is then transformed into a proper word-form. Unfortunately, hand-crafted morphological generators such as the ones that they use for this purpose are available only for a small number of languages,

and are expensive to create from scratch. The supervised inflection generation models that we investigate in this chapter can instead be trained on publicly available inflection tables.

The task of an inflection generator is to produce an inflected form given a lemma (e.g., an infinitive) and desired inflection, which can be specified as an abstract inflectional tag. The generator is trained on a number of inflection tables, such as the one previously shown in Figure 1.1, which enumerate inflection forms for a given lemma. At test time, the generator predicts inflections for previously unseen lemmas. For example, given the input *atmen* + *1SIA*, where the tag stands for "first person singular indicative preterite," it should output *atmete*.

Recently, Durrett and DeNero, 2013) and Ahlberg et al., 2014) have proposed to model inflection generation as a two-stage process: an input lemma is first matched with rules corresponding to a paradigm seen during training, which is then used to generate all inflections for that lemma simultaneously. Although their methods are quite different, both systems account for paradigm-wide regularities by creating rules that span all inflections within a paradigm.

In this chapter, we approach the task of supervised inflection generation as discriminative string transduction, in which character-level operations are applied to transform a lemma concatenated with an inflection tag into the correct surface word-form. We carefully model the transformations carried out for a single inflection, taking into account source characters surrounding a rule, rule sequence patterns, and the shape of the resulting inflected word. To take advantage of paradigmatic regularities, we perform a subsequent reranking of the top n word-forms produced by the transducer. In the reranking model, soft constraints capture similarities between different inflection slots within a table. Where previous work leveraged large, rigid rules to span paradigms, our work is characterized by small, flexible rules that can be applied to any inflection, with features determining what rule sequence works best for each pairing of a lemma with an inflection.

Since our target application is machine translation, we focus on maximizing

inflection form accuracy, rather than complete table accuracy. Unlike previous work, which aims at learning linguistically-correct paradigms, our approach is designed to be robust with respect to incomplete and noisy training data. We conduct a series of experiments which demonstrate that our method can accurately learn complex morphological rules in languages with varying levels of morphological complexity. In each experiment we either match or improve over the state of the art reported in previous work. In addition to providing a detailed comparison of the available inflection prediction systems, we also contribute four new inflection datasets composed of Dutch and French verbs, and Czech verbs and nouns, which are made available for future research.

4.1 Inflection Generation

Durrett and DeNero, 2013) formulate the specific task of supervised generation of inflected forms for a given lemma based on a large number of training inflection tables, while Ahlberg et al., 2014) test their alternative method on the same Wiktionary dataset. In this section, we compare their work to our approach with respect to the following three sub-tasks:

1. character-wise alignment of the word-forms in an inflection table (Section 4.1.1),
2. extraction of rules from aligned forms (4.1.2),
3. matching of rules to new lemmas (4.1.3).

4.1.1 Table alignment

The first step in supervised paradigm learning is the alignment of related inflected forms in a table. Though technically a multiple-alignment problem, this can also be addressed by aligning each inflected form to a lemma. Durrett & DeNero do exactly this, aligning each inflection to the lemma with a paradigm-aware, position-dependent edit distance. Ahlberg et al use finite-state-automata to implement a multiple longest-common-sub-sequence (LCS) alignment, avoiding the use of an explicit lemma. Both systems leverage the

intuition that character alignment is mostly a problem of aligning those characters that remain unchanged throughout the inflection table.

Our alignment approach differs from previous work in that we use an EM-driven, many-to-many aligner, as described in Section 2.1. Instead of focusing on unchanged characters within a single paradigm, we look for small multi-character operations that have statistical support across all paradigms. This includes operations that simply copy their source into the target, leaving the characters unchanged.

4.1.2 Rule extraction

The second step involves transforming the character alignments into inflection rules. Both previous efforts begin addressing this problem in the same way: by finding maximal, contiguous spans of changed characters, in the lemma for Durrett & DeNero, and in the aligned word-forms for Ahlberg et al. Given those spans, the two methods diverge quite substantially. Durrett & DeNero extract a rule for each *changed* span, with the rule specifying transformations to perform for each inflection. Ahlberg et al instead replace each *unchanged* span with a variable, creating a single rule that specifies complete inflections for the entire table. The latter approach creates larger rules, which are easier to interpret for a linguist, but are less flexible, and restrict information sharing across paradigms.

We move in the opposite direction by extracting a rule for each minimal, multi-character transformation identified by our aligner, with no hard constraint on what rules travel together across different inflections. We attempt to learn atomic character transformations, which extends the flexibility of our rules at the cost of reduced interpretability.

The differences in rule granularity are illustrated on the German verb *schleichen* "to sneak" in Figure 4.1. The single rule of Ahlberg et al comprises three vertical rules of Durrett & DeNero, which in turn correspond to eleven atomic rules in our system. Note that this is a simplification, as alignments and word boundary markers vary across the three systems.

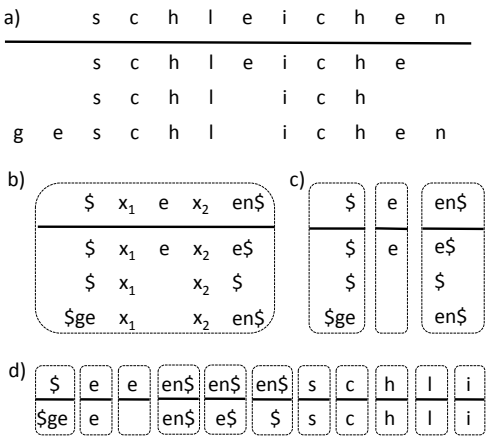

Figure 4.1: Competing strategies for rule extraction: (a) an aligned table; (b) a table-level rule; (c) vertical rules; (d) atomic rules. $ is a word boundary marker.

4.1.3 Rule selection

The final component of an inflection generation system is a mechanism to determine what rules to apply to a new lemma, in order to generate the inflected forms. The strongest signal for this task comes from learning how the training lemmas use the rules. With their highly restrictive rules, Ahlberg et al can afford a simple scheme, keeping an index that associates rules with lemmas, and employing a longest suffix match against this index to assign rules to new lemmas. They also use the corpus frequency of the inflections that would be created by their rules as a rule-selection feature. Durrett & DeNero have much more freedom, both in what rules can be used together and in where each rule can be applied. Therefore, they employ a more complex semi-Markov model to assign rules to spans of the lemma, with features characterizing the n-gram character context surrounding the source side of each rule.

Since our rules provide even greater flexibility, we model rule application very carefully. Like Durrett & DeNero, we employ a discriminative semi-Markov model that considers source character context, and like Ahlberg et al, we use a corpus to re-evaluate predictions. In addition, we model rule sequences, and the character-shape of the resulting inflected form. Note that our rules are much more general than those of our predecessors, which makes it easy to get statistical support for these additional features. Finally, since

our rules are not bound by paradigm structure, we employ a reranking step to account for intra-paradigm regularities.

4.2 Discriminative Transduction

The core of our transduction method is built around discriminative string transduction with DIRECTL+, as described in Section 2.2. In this section, we describe the details of our approach for inflection generation, including the affix representation, the string alignment and transduction, and the paradigm reranking.

4.2.1 Affix representation

Our inflection generation engine is a discriminative semi-Markov model, similar to a monotonic phrase-based decoder from machine translation (Zens and Ney, 2004). This system cannot insert characters, except as a part of a phrasal substitution, so when inflecting a lemma, we add an abstract affix representation to both provide an insertion site and to indicate the desired inflection.

Abstract tags are separated from their lemmas with a single '+' character. Marking the morpheme boundary in such a way allows the transducer to generalize the context of a morpheme boundary. For example, the third person singular indicative present of the verb *atmen* is represented as *atmen+3SIE*. We use readable tags throughout this dissertation, but they are presented to the transducer as indivisible units; it cannot translate them character-by-character.

German and Dutch past participles, as well as several Czech inflections, are formed by circumfixation. We represent such inflections with separate copies of the circumfix tag before and after the lemma. For example, the past participle *gebracht* "brought" is represented as *PPL+bringen+PPL*. In the absence of language-specific information regarding the set of inflections that involve circumfixation, the system can learn to transduce particular affixes into empty strings.

During development, we experimented with an alternative method, in

which affixes are represented by a default allomorph. Allomorphic representations have the potential advantage of reducing the complexity of transductions by the virtue of being similar to the correct form of the affix. However, we found that allomorphic affixes tend to obfuscate differences between distinct inflections, so we decided to employ abstract tags instead.

In addition to the general model that is trained on all inflected word-forms, we derive tag-specific models for each type of inflection. Development experiments showed the general model to be slightly more accurate overall, but we use both types of models in our reranker.

4.2.2 Reranking

Morphological processes such as stem changes tend to be similar across different word-forms of the same lemma. In order to take advantage of such paradigmatic consistency, we perform a reranking of the n-best word-forms generated by DirecTL+. The correct form is sometimes included in the n-best list, but with a lower score than an incorrect form. We propose to rerank such lists on the basis of features extracted from the 1-best word-forms generated for other inflection slots, the majority of which are typically correct.

We perform reranking using the method described in Section 2.3. An initial inflection table, created to generate reranking features, is composed of 1-best predictions from the general model. For each inflection, we then generate lists of candidate forms by taking the intersection of the n-best lists from the general and the tag-specific models.

In order to generate features from our initial inflection table, we make pairwise comparisons between a prediction and each form in the initial table. We separate stems from affixes using the alignment. Our three features indicate whether the compared forms share the same stem, the same affix, and the same surface word-form, respectively. We generate a feature vector for each aligned pair of related word-forms, such as past participle vs. present participle. In addition, we include as features the confidence scores generated by both models.

Two extra features are designed to leverage a large corpus of raw text.

Language / POS	Set	Lemmas	Infl.
German Nouns	DE-N	2764	8
German Verbs	DE-V	2027	27
Spanish Verbs	ES-V	4055	57
Finnish Nouns	FI-N	6400[1]	28
Finnish Verbs	FI-V	7249	53
Dutch Verbs	NL-V	11200	9
French Verbs	FR-V	6957	48
Czech Nouns	CZ-N	21830	17
Czech Verbs	CZ-V	4435	54

Table 4.1: The number of lemmas and inflections for each dataset.

A binary indicator feature fires if the generated form occurs in the corpus. In order to model the phonotactics of the language, we also derive a 4-gram character language model from the same corpus, and include as a feature the normalized log-likelihood of the predicted form.

4.3 Experiments

We perform five experiments that differ with respect to the amount and completeness of training data, and whether the training is performed on individual word-forms or entire inflection tables. We follow the experimental settings established by previous work, as much as possible.

The parameters of our transducer and aligner were established on a development set of German nouns and verbs, and kept fixed in all experiments. We limit stem alignments to 2-2, affix alignments to 2-4, source context to 8 characters, joint n-grams to 5 characters, and target Markov features to 2 characters.

4.3.1 Inflection data

We adopt the Wiktionary inflection data made available by Durrett and DeNero, 2013), with the same training, development, and test splits. The development and test sets each contain 200 complete inflection tables, and the training sets consist of the remaining data. Table 4.1 shows the total number of tables

Case	Singular	Plural
Nominative	Buch	Bücher
Accusative	Buch	Bücher
Dative	Buch	Büchern
Genitive	Buches	Bücher

Table 4.2: All word-forms of the German noun *Buch*.

in each language set. We convert their inflectional information to abstract tags for input to our transducer.

We augment the original five datasets with four new sets: Dutch verbs from the CELEX lexical database (Baayen et al., 1995), French verbs from Verbiste, an online French conjugation dictionary[2], and Czech nouns and verbs from the Prague Dependency Treebank (Böhmová et al., 2003). For each of these sets, the training data is restricted to 80% of the inflection tables listed in Table 4.1, with 10% each for development and testing. Each lemma inflects to a finite number of forms that vary by part-of-speech and language (Table 4.1); German nouns inflect for number and case (Table 4.2), while French, Spanish, German, and Dutch verbs inflect for number, person, mood, and tense.

We extract Czech data from the Prague Dependency Treebank, which is fully annotated for morphological information. This dataset contains few complete inflection tables, with many lemmas represented by a small number of word-forms. For this reason, it is only suitable for one of our experiments, which we describe in Section 4.3.5.

Finnish has a morphological system that is unlike any of the Indo-European languages. There are 15 different grammatical cases for nouns and adjectives, while verbs make a number of distinctions, such as conditional vs. potential, and affirmative vs. negative. We derive separate models for two noun classes (singular and plural), and six verb classes (infinitive, conditional, potential, participle, imperative, and indicative). This is partly motivated by the number of individual training instances for Finnish, which is much larger than the other languages, but also to take advantage of the similarities within classes.

[2]http://perso.b2b2c.ca/sarrazip/dev/verbiste.html

For the reranker experiments, we use the appropriate Wikipedia language dump. The number of tokens in the corpora is approximately 77M for Czech, 200M for Dutch, 6M for Finnish, 425M for French, 550M for German, and 400M for Spanish.

4.3.2 Individual inflections

In the first experiment, we test the accuracy of our basic model which excludes our reranker, and therefore has no access to features based on inflection tables or corpus counts. Table 4.3 compares our results against the Factored model of Durrett & DeNero(DDN) the neural system of Faruqui et al., 2016)(FTND), and the CRF system of Liu and Mao, 2016)(LM), which also make independent predictions for each inflection. The numbers marked with an asterisk were not reported in the original paper, but were generated by running their publicly-available code on our new Dutch and French datasets. For the purpose of quantifying the effectiveness of our reranker, we also include the percentage of correct answers that appear in our 10-best lists.

Our basic model achieves higher accuracy that Durrett & DeNero on all datasets, which shows that our refined transduction features are consistently more effective than the source-context features employed by their system. We see that our results are also competitive with the neural system of Faruqui et al., 2016). Their system seems to be particularly well suited to Finnish, where their system is able to generalize phenomena such as vowel harmony and consonant gradation. The system of Durrett & DeNero, as well as the system of Ahlberg et al., 2014), is intended for whole-table scenarios, which we test next.

4.3.3 Complete paradigms

In this experiment, we assume the access to complete inflection tables, as well as to raw corpora. We compare our reranking system to the Joint model of Durrett & DeNero (DDN), which is trained on complete tables, and the full model of Ahlberg et al., 2014) (AFH), which is trained on complete tables, and matches forms to rules with aid of corpus counts. To provide a fair comparison

Set	DDN	LM	FTND	Ours	10-best
DE-V	94.8	96.1	96.7	**97.5**	99.8
DE-N	88.3	83.8	88.1	**88.6**	98.6
ES-V	99.6	99.6	**99.8**	**99.8**	100
FI-V	97.2	97.2	97.8	**98.1**	99.9
FI-N	92.1	92.3	**95.4**	93.0	99.0
NL-V	90.5*	NA	**96.7**	96.1	99.4
FR-V	98.8*	NA	98.8	**99.2**	99.7

Table 4.3: Prediction accuracy of models trained and tested on individual inflections.

Set	DDN	AFH14	Ours
DE-V	96.2	**97.9**	**97.9**
DE-N	88.9	**91.8**	89.9
ES-V	99.7	99.6	**99.9**
FI-V	96.4	96.6	**98.1**
FI-N	93.4	**93.8**	93.6
NL-V	94.4*	87.7*	**96.6**
FR-V	96.8*	98.1*	**99.2**

Table 4.4: Individual form accuracy of models trained on complete inflection tables.

with our reranker, which incorporates information from raw data, we compare against systems that also make use of a corpus. Again, we calculated the numbers marked with an asterisk by running the respective implementations on our new datasets.

The results of the experiment are shown in Table 4.4. Our reranking model outperforms the Joint model of DDN on all sets, and the full model of AFH on most verb sets. Looking across tables to Table 4.3, we can see that reranking improves upon our independent model on 5 out of 7 sets, and is equivalent on the remaining two sets. However, according to single-form accuracy, neither our system nor DDN benefits too much from joint predictions. Table 4.5 shows the same results evaluated with respect to complete table accuracy.

Ahlberg et al., 2015) report improved results over the previous paper, but fail to report results augmented with a corpus. To maintain a fair comparison with regards to the available information, we do not report those results here.

Set	DDN	AFH	Ours
DE-V	85.0	76.5	**90.5**
DE-N	79.5	**82.0**	76.5
ES-V	95.0	98.0	**99.0**
FI-V	87.5	92.5	**94.5**
FI-N	83.5	**88.0**	82.0
NL-V	79.5*	37.7*	**82.1**
FR-V	92.1*	96.0*	**97.1**

Table 4.5: Complete table accuracy of models trained on complete inflection tables.

4.3.4 Incomplete paradigms

In this experiment, we consider a scenario where, instead of complete tables, we have access to some but not all of the possible word-forms. This could occur for example if we extracted our training data from a morphologically annotated corpus. We simulate this by only including in our training tables the forms that are observed in the corresponding raw corpus. We then test our ability to predict the same test forms as in the previous experiments, regardless of whether or not they were observed in the corpus. We also allow a small held-out set of complete tables, which corresponds to the development set. For Durrett & DeNero's method, we include this held-out set in the training data, while for our system, we use it to train the reranker.

The Joint method of DDN and the methods of AFH are incapable of training on incomplete tables, and thus, we can only compare our results against the Factored model of DDN. However, unlike their Factored model, we can then still take advantage of paradigmatic and corpus information, by applying our reranker to the predictions made by our simple model.

The results are shown in Table 4.6, where we refer to our independent model as *Basic*, and to our reranked system as *Reranked*. The latter outperforms DDN on all sets. Furthermore, even with only partial tables available during training, reranking improves upon our independent model in every case.

4.3.5 Partial paradigms

We run a separate experiment for Czech, as the data is substantially less comprehensive than for the other languages. Although the number of 13.0% observed noun forms is comparable to the Finnish case, the percentages in Table 4.6 refer only to the training set: the test and held-out sets are complete. For Czech, the percentage includes the testing and held-out sets. Thus, the method of Durrett & DeNero and our reranker have access to less training data than in the experiment of Section 4.3.4.

The results of this experiment are shown in Table 4.7. Our Basic model outperforms DDN for both nouns and verbs, despite training on less data. However, reranking actually decreases the accuracy of our system on Czech nouns. It appears that the reranker is adversely affected by the lack of complete target paradigms. We leave the full investigation into the effectiveness of the reranker on incomplete data to future work.

4.3.6 Seed paradigms

Dreyer and Eisner, 2011) are particularly concerned with situations involving limited training data, and approach inflection generation as a semi-supervised task. In our last experiment we follow their experimental setup, which simulates the situation where we obtain a small number of complete tables from an expert. We use the same training, development, and test splits to test our system. Due to the nature of our model, we need to set aside a hold-out set for reranking. Thus, rather than training on 50 and 100 tables, we train on 40 and 80, but compare the results with the models trained on 50 and 100, respectively. For reranking, we use the same German corpus as in our previous experiments, but limited to the first 10M words.

The results are shown in Table 4.8. When trained on 50 seed tables, the accuracy of our models is comparable to both the basic model of Dreyer and Eisner (DE) and the Factored model of DDN, and matches the best system when we add reranking. When trained on 100 seed tables, our full reranking model outperforms the other models.

Set	% of Total	DDN	Ours	
			Basic	Reranked
DE-V	69.2	90.2	96.2	**97.9**
DE-N	92.7	88.3	88.4	**89.8**
ES-V	36.1	97.1	95.9	**99.6**
FI-V	15.6	73.8	78.7	**85.6**
FI-N	15.2	71.6	78.2	**80.4**
DU-V	50.5	89.8	94.9	**96.0**
FR-V	27.6	94.6	96.6	**98.9**

Table 4.6: Prediction accuracy of models trained on observed forms.

Set	% of Total	DDN	Ours	
			Basic	Reranked
CZ-N	13.0	91.1	**97.7**	93.5
CZ-V	6.8	82.5	83.6	**85.8**

Table 4.7: Prediction accuracy of models trained on observed Czech forms.

4.4 Error analysis

In this section, we analyze several types of errors made by the various systems.

Non-word predictions are marked with an asterisk.

German and Dutch are closely-related languages that exhibit similar errors. Many errors involve the past participle, which is often created by circumfixation. For the German verb *verfilmen* "to film," we predict the correct *verfilmt*, while the other systems have *verfilmen**, and *geverfilmt**, respectively. DDN simply select an incorrect rule for the past participle. AFH choose paradigms through suffix analysis, which fails to account for the fact that verbs that begin with a small set of prefixes, such as *ver-*, do not take a *ge-* prefix. This type of error particularly affects the accuracy of AFH on Dutch because of a number of verbs in our test set that involve infixation for the past participle. Our system uses its source and target-side n-gram features to match these prefixes with their correct representation.

The second type of error is an over-correction by the corpus. The past participle of the verb *dimmen* is *gedimmt*, but AFH predict *dimmt**, and then change it to *dummen* with the corpus. *Dummen* is indeed a valid word in

Seed Tables	DE		DDN		Ours	
	Basic	Full	Factored	Joint	Basic	Full
50	89.9	**90.9**	89.6	90.5	89.7	**90.9**
100	91.5	92.2	91.4	92.3	92.0	**92.6**

Table 4.8: Prediction accuracy on German verb forms after training on a small number of seed inflection tables.

German, but unrelated to the verb *dimmen*. It is also far more common, with 181 occurrences in the corpus, compared with only 28 for *gedimmt*. Since AFH use corpus frequencies, mistakes like this can occur. Our system is trained to balance transducer confidence against a form's existence in a corpus (as opposed to log frequency), which helps it ignore the bias of common, but incorrect, forms.

The German verb *brennen* "to burn" has an irregular past participle: *gebrannt*. It involves both a stem vowel change and a circumfix, two processes that only rarely co-occur. AFH predict the form *brannt**, using the paradigm of the similar *bekennen*. The flexibility of DDN allows them to predict the correct form. Our basic model predicts *gebrennt**, which follows the regular pattern of applying a circumfix, while maintaining the stem vowel. The reranker is able to correct this mistake by relating it to the form *gebrannt* in the corpus, whose stem is identical to the stem of the preterite forms, which is a common paradigmatic pattern.

Our system can also over-correct, such as with the second person plural indicative preterite form for the verb *reisen*, which should be *reistet*, and which our basic model correctly predicts. The reranker, however, changes the prediction to *rist*. This is a nominal form that is observed in the corpus, while the verbal form is not.

An interesting example of a mistake made by the Factored model of DDN involves the Dutch verb *aandragen*. Their model learns that stem vowel *a* should be doubled, and that an *a* should be included as part of the suffix *-agt*, which results in an incorrect form *aandraaagt**. Thanks to the modeling of phonotactics, our model is able to correctly rule out the tripling of a vowel.

Finnish errors tend to fall into one of three types. First, words that involve harmonically neutral vowels, such as "e" and "i" occasionally cause errors in vowel harmony. Second, all three systems have difficulty identifying syllable and compound boundaries, and make errors predicting vowels near boundaries. Finally, consonant gradation, which alternates consonants in open and closed syllables, causes a relatively large number of errors; for example, our system predicts **heltempien*, instead of the correct *hellempien* as the genitive singular of the comparative adjective *hellempi* "more affectionate".

4.5 Discussion

We have proposed an alternative method of generating inflected word-forms which is based on discriminative string transduction and reranking. We have conducted a series of experiments on nine datasets involving six languages, including four new datasets that we created. The results demonstrate that our method is not only highly accurate, but also robust against incomplete or limited inflection data.

Now that we are confident that our methods can produce high quality inflections, we reverse the direction of inflectional modeling. In the next chapter, we use our sequential prediction methods and inflection tables to model stemming and lemmatization.

Chapter 5

Leveraging Inflection Tables for Stemming and Lemmatization

Many languages contain multiple inflected forms that correspond to the same dictionary word. Inflection is largely a grammatical procedure that keeps the core meaning of a word. For example, the German words in Table 5.1 all refer to the action of giving. When working with these languages, it is often beneficial to establish a consistent representation across a set of inflections. This is the task that we address here.

There are two principal approaches to inflectional simplification: stemming and lemmatization. Stemming aims at removing inflectional affixes from a word form. It can be viewed as a kind of word segmentation, in which the boundaries of the stem are identified within the word; no attempt is made to restore stem changes that may occur as part of the inflection process. The goal of lemmatization is to map any inflected form to its unique *lemma*, which is typically the word form that represents a set of related inflections in a dictionary. Unlike stemming, lemmatization must always produce an actual word form.

In this chapter, we present a discriminative string transduction approach to both stemming and lemmatization. Supervised stemmers require morphologically annotated corpora, which are expensive to build. We remove this constraint by extracting stems from semi-structured inflection tables, such as the one shown in Table 5.2, in an unsupervised manner. We design two transduction models that are trained on such stems, and evaluate them on unseen

Word form	Meaning	Tag	Stem
geben	"to give"	INF	geb
gibt	"gives"	3SIE	gib
gab	"gave"	1SIA	gab
gegeben	"given"	PP	geb

Table 5.1: Examples of German word-forms corresponding to the lemma *geben*.

	Singular			Plural
	1^{st}	2^{nd}	3^{rd}	1^{st}
Present	*doy*	*das*	*da*	*damos*
Imperfect	*daba*	*dabas*	*daba*	*dábamos*
Preterite	*di*	*diste*	*dio*	*dimos*
Future	*daré*	*darás*	*dará*	*daramos*

Table 5.2: A partial inflection table for the Spanish verb *dar* "to give".

forms against a supervised model. We then extend our stemming models to perform the lemmatization task, and to incorporate an unannotated corpus. We evaluate them on several datasets. Our best system improves the state of the art for Dutch, German, and Spanish.

5.1 Stemming Methods

We approach stemming as a string transduction task. Stemming can be performed by inserting morpheme boundary markers between the stem and the affixes. For example, the German verb form *gegeben* is transduced into `ge+geb+en`, which induces the stem `geb`.

5.1.1 Supervised Transduction

Using the methods described in Section 2.1, we align source and target pairs to train a stemming model. The source consists of a word-form, while the target is identical, except that it also includes boundary markers between the stem and affixes. For example, one pair might be *geschrieben*, *ge+schrieb+en*.

Once we have aligned the source and target pairs, we proceed to train a *word-to-stem* transduction model for stemming unseen test instances. The

STEM|INF	**geb**|en	**setz**|en	**tu**|n
STEM|1SIA	gab|-	setz|te	tat|-
STEM|2SIE	gib|st	setz|t	tu|st
PP|STEM|PP	ge|geb|en	ge|setz|t	ge|ta|n

Table 5.3: Stemming of the training data based on the patterns of regularity in inflectional tables. Stemmas are shown in bold.

word-to-stem model learns where to insert boundary markers. We refer to a model that is trained on annotated morphological segmentations as our supervised method.

5.1.2 Unsupervised Segmentation

In order to train a fully-supervised model for stemming, large lists of morphologically-segmented words are generally required. While such annotated corpora are rare, semi-structured, crowd-sourced inflection tables are available for many languages on websites such as Wiktionary (Table 5.2). In this section, we introduce an unsupervised method of inducing stems by leveraging paradigmatic regularity in inflection tables.

Sets of inflection tables often exhibit the same inflectional patterns, called paradigms, which are based on phonological, semantic, or morphological criteria (cf. Table 5.3). Each table consists of lists of word forms, including the lemma. The number of distinct stems, such as *'geb'* and *'gib'* for the verb *geben*, is typically very small, averaging slightly over two per German verb inflection table. The number of distinct affix forms corresponding to the same inflectional form across different lemmas is also small, averaging below three for German verbs. For example, the second person singular indicative present suffix is always either *-st*, *-est*, or *-t*.

We take advantage of this relative consistency to determine the boundaries between the stems and affixes of each word form in an unsupervised manner. We first associate each word form in the training data with an abstract tag sequence, which is typically composed of the `STEM` tag and a suffix tag representing a given inflection slot (Table 5.3). We then apply the unsupervised

Source	g	i	b	t
Target	g	i	b	+t
Tags	STEM			3SIE
Joint	g	**e**	b	+3SIE

Table 5.4: Alignment of the various representations of the word *gibt*.

aligner to determine the most likely alignment between the character sequences and the tags, which are treated as indivisible units. The aligner simultaneously learns common representations for stems within a single inflection table, as well as common representations for each affix across multiple tables.

Some inflections, such as the German past participle (`PP` in Table 5.3) involve a *circumfix*, which can be analyzed as a prefix-suffix combination. Prior to the alignment, we associate all forms that belong to the inflection slots involving circumfixation with tag sequences composed of three tags. Occasionally, a word form will only have a suffix where one would normally expect a circumfix (e.g. *existiert*). In order to facilitate tag alignment in such cases, we prepend a dummy null character to each surface word form.

After the stem-affix boundaries have been identified, we proceed to train a *word-to-stem* transduction model as described in Section 5.1.1. We refer to this unsupervised approach as our basic method (cf. Figure 5.1).

5.1.3 Joint Stemming and Tagging

The method described in the previous section fails to make use of a key piece of information in the inflection table: the lemma. The stem of an inflected form is typically either identical or very similar to the stem of its lemma, or *stemma* (Table 5.3). Our joint method takes advantage of this similarity by transducing word-forms into stemmas with tags.

The format of the training data for the *word-to-stemma* model is different from the *word-to-stem* model. After the initial segmentation of the source word-forms into morphemes by the unsupervised aligner, as described in Section 5.1.2, the stems are replaced with the corresponding stemmas, and the affixes are replaced with the inflection tags. For example, the form *gibt* is

	Words	Noun	Verb	Adj
English	50,155	2	5	3
Dutch	101,667	2	9	3
German	96,038	8	27	48

Table 5.5: The number of words and distinct inflections for each language in the CELEX datasets.

paired with the sequence `geb+3SIE`, with the stem and stemma re-aligned at the character level as shown in Table 5.4.

Unlike the basic method, which simply inserts morpheme breaks into word-forms, the joint method uses the tags to identify the boundaries between stems and affixes. At test time, the input word-form is transduced into a stemma and tag sequence. The character string that has generated the tag is then stripped from the input word-form to obtain the stem. By making use of both the tags and the stemma, the *word-to-stemma* model jointly optimizes the stem and affix combination. We refer to this unsupervised approach as our joint method.

5.2 Stemming Experiments

Precise evaluation of stemming methods requires morphologically annotated lexicons, which are rare. Unlike lemmas, stems are abstract representations, rather than actual word forms. Unsurprisingly, annotators do not always agree on the segmentation of a word. In this section, we describe three experiments for evaluating stem extraction, intrinsic accuracy, and consistency.

We evaluate our methods against three systems that are based on very different principles. Snowball[1] is a rule-based program based on the methodology of the Porter Stemmer. Morfessor FlatCat (Grönroos et al., 2014) performs unsupervised morphological segmentation, and approximates stemming by distinguishing stems and affixes.[2] Chipmunk (Cotterell et al., 2015), is a fully-supervised system that represents the current state of the art.

[1] http://snowball.tartarus.org

[2] Morfessor is applied to the union of the training and test data.

	EN	NL	DE
Our method	85.9	88.0	85.7
Snowball	48.2	58.8	49.5
Morfessor	61.4	71.4	61.4

Table 5.6: Unsupervised stemming accuracy of the CELEX training set.

5.2.1 Data

We perform an evaluation of stemming on English (EN), Dutch (NL), and German (DE) lexicons from CELEX (Baayen et al., 1995). The three languages vary in terms of morphological complexity (Table 5.5). We use the morphological boundary annotations for testing all stemming systems, as well as for training our supervised system.

For both unsupervised systems, we could build training sets from any inflection tables that contain unsegmented word-forms. However, in order to perform a precise comparison between the supervised and unsupervised systems, we extract the inflection tables from CELEX, disregarding the segmentation information. Each system is represented by a single stemming model that works on nouns, verbs, and adjectives. Due to differences in representation, the number of training instances vary slightly between models, but the number of words is constant (Table 5.5).

In order to demonstrate that our unsupervised methods require no segmentation information, we create additional German training sets using the inflection tables extracted from Wiktionary by Durrett and DeNero, 2013). The sets contain 18,912 noun forms and 43,929 verb forms. We derive separate models for verbs and nouns in order to compare the difficulty of stemming different parts of speech.

The test sets for both CELEX and Wiktionary data come from CELEX, and consist of 5252, 6155, and 9817 unique forms for English, Dutch, and German, respectively. The German test set contains 2620 nouns, 3837 verbs, and 3360 adjectives.

	EN	NL	DE
Supervised	98.5	96.0	91.2
Basic	82.3	89.1	80.9
Joint	94.6	93.2	86.0
Snowball	50.0	58.4	48.2
Morfessor	65.2	60.9	51.8

Table 5.7: Stemming accuracy of systems trained and tested on CELEX datasets.

Chipmunk[3] requires training data in which every morpheme of a word is annotated for morphological function. Since this information is not included in CELEX, we train and test Chipmunk, as well as a version of our supervised model, on the data created by Cotterell et al., 2015), which is much smaller. The English and German segmentation datasets contain 1161 and 1266 training instances, and 816 and 952 test instances, respectively.

5.2.2 Stem Extraction Evaluation

First, we evaluate our unsupervised segmentation approach, which serves as the basis for our basic and joint models, on the union of the training and development parts of the CELEX dataset. We are interested how often the stems induced by the method described in Section 5.1.2 match the stem annotations in the CELEX database.

The results are presented in Table 5.6. Our method is substantially more accurate than either Snowball or Morfessor. Snowball, despite being called a stemming algorithm, often eliminates derivational affixes; e.g. `able` in *unbearable*. Morfessor makes similar mistakes, although less often. Our method tends to prefer longer stems and shorter affixes. For example, it stems *verwandtestem*, as `verwandte`, while CELEX has `verwandt`.

5.2.3 Intrinsic Evaluation

The results of the intrinsic evaluation of the stemming accuracy on unseen forms in Tables 5.7-5.9 demonstrate the quality of our three models. The joint

[3]http://cistern.cis.lmu.de/chipmunk

	Noun	Verb
Basic	76.8	90.3
Joint	85.2	91.1
Snowball	55.5	39.8
Morfessor	61.9	34.9

Table 5.8: German stemming accuracy of systems trained on Wiktionary data, and tested on the CELEX data.

	EN	DE
Supervised	94.7	85.1
Chipmunk	94.9	87.4

Table 5.9: Stemming accuracy of systems trained and tested on the Chipmunk data.

model performs better than the basic model, and approaches the accuracy of the supervised model. On the CELEX data, our unsupervised joint model substantially outperforms Snowball and Morfessor on all three languages (Table 5.7).[4] These results are further confirmed on the German Wiktionary data (Table 5.8). Our supervised model performs almost as well as Chipmunk on its dataset (Table 5.9).

A major advantage of the joint model over the basic model is its tag awareness (cf. Table 5.4). Although the tags are not always correctly recovered on the test data, they often allow the model to select the right analysis. For example, the basic model erroneously segments the German form *erklärte* as `erklärt+e` because `+e` is a common verbal, adjectival and nominal suffix. The joint model, recognizing `er` as a verbal derivational prefix, predicts a verbal inflection tag (`+1SIA`), and the correct segmentation `erklär+te`. Verbal stems are unlikely to end in `ärt`, and `+te`, unlike `+e`, can only be a verbal suffix.

5.2.4 Consistency Evaluation

When stemming is used for inflectional simplification, it should ideally produce the same stem for all word-forms that correspond to a given lemma. In

[4]The decrease in Morfessor accuracy between Tables 5.6 and 5.7 can be attributed to a different POS distribution between training and testing.

	EN	NL	DE
Gold	1.10	1.17	1.30
Supervised	1.13	1.64	1.50
Basic	1.06	1.21	1.25
Joint	1.09	1.08	1.20
Snowball	1.03	1.45	2.02
Morfessor	1.11	1.68	3.27

Table 5.10: Average number of stems per lemma.

many cases, this is not an attainable goal because of internal stem changes (cf. Table 5.1). However, most inflected words follow regular paradigms, which involve no stem changes. For example, all forms of the Spanish verb *cantar* contain the substring `cant`, which is considered the common stem. We quantify the extent to which the various systems approximate this goal by calculating the average number of unique generated stems per inflection table in the CELEX test sets.[5]

The results are presented in Table 5.10. The stems-per-table average tends to reflect the morphological complexity of a language. All systems achieve excellent consistency on English, but the Dutch and German results paint a different picture. The supervised system falls somewhat short of emulating the gold segmentations, which may be due to the confusion between different parts of speech. In terms of consistency, the stems generated by our unsupervised methods are superior to those of Snowball and Morfessor, and even to the gold stems. We attribute this surprising result to the fact that the EM-based alignment of the training data favors consistency in both stems and affixes, although this may not always result in the correct segmentation.

5.3 Lemmatization Methods

In this section, we present three supervised lemmatization methods, two of which incorporate the unsupervised stemming models described in Section 5.1.

[5]Chipmunk is excluded from the consistency evaluation because its dataset is not composed of complete inflection tables.

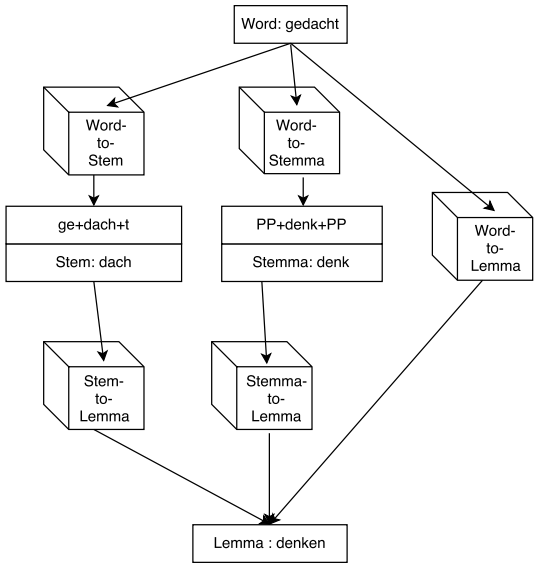

Figure 5.1: Three lemmatization methods.

The different approaches are presented schematically in Figure 5.1, using the example of the German past participle *gedacht*.

5.3.1 Stem-based Lemmatization

Our stem-based lemmatization method is an extension of our basic stemming method. We compose the *word-to-stem* transduction model from Section 5.1 with a *stem-to-lemma* model that converts stems into lemmas. The latter is trained on character-aligned pairs of stems and lemmas, where stems are extracted from the inflection tables via the unsupervised method described in Section 5.1.2. The inflection tables in our training data always contain the lemma, making the creation of stem / lemma pairs a trivial task.

5.3.2 Stemma-based Lemmatization

Our stemma-based lemmatization method is an extension of our joint stemming method. We compose the *word-to-stemma* transduction model described in Section 5.1.3 with a *stemma-to-lemma* model that converts stems into lemmas. The latter is trained on character-aligned pairs of stemmas and lemmas, where stemmas are extracted via the method described in Section 5.1.3. Typ-

ically, the model simply appends a lemmatic affix to the stemma, as all stem changes are handled by the *word-to-stemma* model.

5.3.3 Direct Lemmatization

Our final lemmatization method is a *word-to-lemma* transduction model that directly transforms word-forms into lemmas and tags. The model is trained on word-forms paired with their lemmas and inflectional tags. If a word-form has multiple tags, we present all of them to the algorithm, shuffling them in training to prevent a single tag from having any bias over others, other than biases incurred through frequency of affix / tag alignments. A potential advantage of this direct method lies in removing the possibility of error propagation that is inherent in pipeline approaches. However, it involves a more complex transduction model that must simultaneously apply both stem changes, and transform inflectional affixes into lemmatic ones.

5.3.4 Reranking

Intuitively, lemmatization accuracy could be improved by leveraging large, unannotated corpora. After generating n-best lists of possible lemmas, we rerank them using the method described in Section 2.3. We employ four features of the prediction:

1. normalized score from DirecTL+,
2. rank in the n-best list
3. presence in the corpus,
4. normalized likelihood from a 4-gram character language model derived from the corpus.

5.4 Lemmatization Experiments

Unlike stemming, lemmatization is a completely consistent process: all word-forms within an inflection table correspond to the same lemma. In this section, we describe intrinsic and extrinsic experiments to evaluate the quality of the

	Wiki	CELEX			CoNLL		
	ES	EN	NL	DE	EN	DE	ES
Stem-based	97.1	89.1	82.3	76.3	90.2	71.1	83.2
Stemma-based	94.5	96.4	85.2	85.8	**92.5**	75.9	91.2
Direct	**98.8**	96.4	**89.5**	**88.7**	**92.5**	**80.1**	**91.5**
Morfette	98.0	96.0	80.2	81.3	**92.5**	73.5	**91.5**
Lemming	98.6	**96.7**	86.6	88.2	**92.5**	77.9	90.4

Table 5.11: Lemmatization results without the use of a corpus.

lemmas generated by our systems, and compare the results against the current state of the art.

5.4.1 Data

As in our stemming experiments, we extract complete English, Dutch, and German inflection tables from CELEX. We use the same data splits as in Section 5.2.1. We also evaluate our methods on Spanish verb inflection tables extracted from Wiktionary by Durrett and DeNero, 2013), using the original data splits. Spanish is a Romance language, with a rich verbal morphology comprising 57 inflections for each lemma.

A different type of dataset comes from the CoNLL-2009 Shared Task (Hajič et al., 2009). Unlike the CELEX and Wiktionary datasets, they are extracted from an annotated text, and thus contain few complete inflection tables, with many lemmas represented by a small number of word-forms. We extract all appropriate parts-of-speech from the test section of the corpus for English, German, and Spanish. This results in a test set of 5165 unique forms for English, 6572 for German, and 2668 for Spanish.

For reranking, we make use of a word list constructed from the first one million lines of the appropriate Wikipedia dump.[6] A character language model is constructed using the CMU Statistical Language Modeling Toolkit.[7] 20% of the development set is reserved for the purpose of training a re-ranking model. For Lemming and Morfette, we provide a lexicon generated from the corpus.

[6] All dumps are from November 2, 2015.

[7] http://www.speech.cs.cmu.edu

Recall that these systems were constructed for contextual lemmatization; we remove context to make their results comparable to our own.

Spanish marks unpredictable stress by marking a stressed vowel with an acute accent (e.g. *cantó* vs. *canto*). In order to facilitate generalization, we perform a lossless pre-processing step that replaces all accented vowels with their unaccented equivalent followed by a special stress symbol (e.g. `canto'`). For consistency, this modification is applied to the data for each system.

5.4.2 Intrinsic Evaluation

We evaluate lemmatization using word accuracy. In cases where a surface word-form without a morphological tag may correspond to multiple lemmas, we judge the prediction as correct if it matches any of the lemmas. For example, both the noun *Schrei* and the verb *schreien* are considered to be correct lemmas for the German word *schreien*.[8]

The results without the use of a corpus are shown in Table 5.11. Thanks to its tag awareness, the stemma-based method is more accurate than the stem-based method, except on the verb-only Spanish Wiktionary dataset. However, our best method is the direct *word-to-lemma* model, which outperforms both Morfette and Lemming on most datasets.

We interpret the results as the evidence for the effectiveness of our discriminative string transduction approach. The direct model is superior to the stemma-based model because it avoids any information loss that may occur during an intermediate stemming step. However, it is still able to take advantage of the tag that it generates together with the target lemma. For example, Lemming incorrectly lemmatizes the German noun form *Verdienste* "earnings" as `verdien` because `+ste` is a superlative adjective suffix. Our direct model, however, considers `dien` to be an unlikely ending for an adjective, and instead produces the correct lemma *Verdienst*.

The results with the use of a corpus are shown in Table 5.12. We omit the results on Spanish Wiktionary and on both English datasets, which are almost identical to those in Table 5.11. We observe that both the stemma-based and

[8]The capitalization of German nouns is ignored.

	CELEX		CoNLL	
	NL	DE	DE	ES
Stem-based	82.3	76.9	71.9	90.6
Stemma-based	87.3	88.4	79.0	**93.3**
Direct	**92.4**	**90.0**	**81.3**	91.9
Lemming	86.9	88.5	77.9	90.6

Table 5.12: Lemmatization results boosted with a raw corpus.

direct methods achieve a substantial error rate reduction on the Dutch and German datasets, while Lemming improvements are minimal.[9] The Spanish CoNLL results are different: only the stem-based and stemma-based methods benefit noticeably from re-ranking.

Error analysis indicates that the re-ranker is able to filter non-existent lemmas, such as `wint` for *Winter*, and `endstadie` for *Endstadien*, instead of *Endstadium*. In general, the degree of improvement seems to depend on the set of randomly selected instances in the held-out set used for training the re-ranker. If a base model achieves a very high accuracy on the held-out set, the re-ranker tends to avoid correcting the predictions on the test set.

5.5 Discussion

We have presented novel methods that leverage readily available inflection tables to produce high-quality stems and lemmas. In the next chapter, we describe the logical extension of our lemmatization method: the production of complete morphological analyses.

[9] We were unable to obtain any corpus improvement with Morfette.

Chapter 6

Morphological Analysis without Expert Annotation

The task of morphological analysis is to annotate a given word-form with its lemma and morphological tag. Since word-forms are often ambiguous, the goal is to produce a complete list of correct analyses, which may involve not only multiple inflections, but also distinct lemmas and parts of speech (c.f. Table 6.1). Hand-built lexicons, such as CELEX (Baayen et al., 1995), contain this kind of information, but they exist only for a small number of languages, are expensive to create, and have limited coverage. Finite-state analyzers, such as Morphisto (Zielinski and Simon, 2009) and Omorfi (Pirinen, 2015), provide an alternative to lexicons, but their construction also requires expert knowledge and substantial engineering effort. However, they are often more general than lexicons, although they may require a lemmatic lexicon to ensure high precision.

Morphological tagging is a distinct but related task, which aims at determining a single correct analysis of a word-form within the context of a sentence. Machine learning taggers, such as Morfette (Chrupala et al., 2008) and Marmot (Mueller et al., 2013), are capable of achieving high tagging accuracy, but they need to be trained on morphologically annotated corpora, which are unavailable for most languages. Often, morphological tagging can be performed as a downstream application of morphological analysis: tools such as Marmot and the Zurich Dependency Parser (Sennrich et al., 2009) have the functionality to incorporate the output of a morphological analyzer

Lemma	POS	Inflection	Tag
luft	Noun	Nom. Pl.	NP
luft	Noun	Acc. Pl.	AP
luft	Noun	Gen. Pl.	GP
lüften	Verb	1^{st} Sg. Ind. Pres.	1SIE
lüften	Verb	1^{st} Sg. Subj. Pres.	1SKE
lüften	Verb	3^{rd} Sg. Subj. Pres.	3SKE
lüften	Verb	Sg. Imperative	RS

Table 6.1: An example of morphological analysis: multiple correct interpretations of the German word-form *lüfte*.

to perform morphological tagging.

In this chapter, we propose a novel discriminative string transduction approach to morphological analysis, which is designed to be trained on plain inflection tables, thus obviating the need for expert rule engineering or morphologically annotated corpora. In addition, our system is capable of leveraging raw unannotated corpora to refine its analyses by re-ranking. The accuracy of the system on German approaches that of a hand-engineered FST analyzer, while having much higher coverage. The experimental results on English, Dutch, German, and Spanish demonstrate that it is also more accurate than the analysis module of a state-of-the-art morphological tagger.

6.1 Methods

Our approach to morphological analysis is based on string transduction between a word-form (e.g. *lüfte*) and an analysis composed of a lemma and a tag (e.g. *lüften*+1SIE), where the tag corresponds to the predicted inflection slot. Our system consists of four modules: alignment, transduction, re-ranking, and thresholding.

We perform alignment and transduction using the modified versions of M2M and DirecTL+ described in Chapter 2. For the analysis task, the source side consists of the surface form of the word, while the target side is a lemma + tag combination. This input format is identical to that of the direct model from Chapter 5. Whereas in that chapter, we only used the tag

s	c	h	r	e	i	b	et	
s	c	h	r	e	i	b	en+2PKA	✓
s	c	h	r	e	i	b	**en+2PKE**	✓
s	c	h	r	e	i	b	en+3SIA	×
s	c	h	r	e	i	b	en+3PIE	×
s	c	h	r	e	i	b	en+2PIA	✓

Table 6.2: Example alignments of hypothetical analyses of the German word-form *schreibet*. The check marks indicate which of the analyses satisfy the affix-match constraint.

to provide information for the lemmatization task, here we are interested in the ability of our method to produce accurate tags, as well.

Unlike lemmatization or inflection generation, a single word may have multiple correct analyses. This could prove to be a problem for DIRECTL+, which was designed with the assumption that there is one correct target form for each input form. We randomly shuffle the order of tags for words with multiple analyses, to prevent a single tag from biasing the model. In practice, we find that this method allows DIRECTL+ to produce multiple correct analyses that are only biased by the frequency of affix / tag pairs in training.

We use the induced alignment as an additional constraint for the generation of morphological tags, which we call the *affix-match constraint*. During training, we record all substring alignments that involve affixes and tags. At test time, the source-target alignment is implied by the substring transduction sequence. We say that a lemma+tag analysis generated from a word-form satisfies the affix-match constraint if and only if the resulting affix-tag pair occurs in the alignment of the training data. Table 6.2 shows the alignments of five possible analyses to the corresponding word-form *schreibet*, of which three satisfy the affix-match constraint. Only analysis #2 (in bold) is correct.

Likewise, we introduce a new constraint to the transduction process, which we call the *mirror constraint*. In addition to training an *analyzer* model that transforms a word-form into an analysis, we also train an *inflector* model that converts an analysis back into a word-form. This opposite transformation corresponds to the task of morphological inflection (i.e, Chapter 4). By de-

Source	Target	
schreiben + 2PKA	schr<u>ie</u>bet	×
schreiben + 2PKE	**schreibet**	✓
schreiben + 3SIA	schr<u>ie</u>b	×
schrieben + 2PKE	schr<u>ie</u>bet	×
schreiben + 2PIA	schr<u>ie</u>bt	×

Table 6.3: Example source-target pairs of the inflector model. The check marks indicate which of the analyses of the German word-form *schreibet* satisfy the mirror constraint.

riving two complementary models from the same training set, we attempt to mimic the functionality of a genuine finite-state transducer. We say that a lemma+tag analysis generated by the analyzer model satisfies the mirror constraint if and only if the inflector model correctly reconstructs the original word-form from the analysis by returning it as its top-1 prediction. Table 6.3 shows five possible analyses of the word-form *schreibet*, of which only one satisfies the mirror constraint. Only analysis #2 (in bold) is correct.

6.1.1 Reranking

In order to produce multiple morphological analyses, we take advantage of the capability of DirecTL+ to output n-best lists of candidate target strings. To promote the most likely lemma+tag combinations, we re-rank the n-best lists using the Liblinear SVM tool (Fan et al., 2008), converting the classification task into a ranking task with the method described in Section 2.3.

The reranker employs several features, which are enumerated in Table 6.4. The first three features consider the form of the predicted lemma. Feature 1 indicates whether the lemma occurs at least once in a text corpus. Feature 2 is set to the normalized likelihood score of the lemma computed with a 4-gram character language model that is derived from the corpus. Feature 3 is the normalized confidence score assigned by DirecTL+.

Features 4-6 refer to the *affix-match* constraint defined in Section 6.1, in order to promote analyses that involve correct tags. Features 4 and 5 are complementary and indicate whether the alignment between the affix of the

	Description	Type
1	lemma in Corpus	binary
2	LM score	real
3	DIRECTL+ score	real
4	affix match	binary
5	no affix match	binary
6	no affix match, top-1	binary
7	mirrored	binary
8	not mirrored	binary
9	not mirrored, top-1	binary

Table 6.4: Features of the re-ranker.

given word-form and the tag of the predicted analysis was generated at least once in the training data. Feature 6 accounts for unusual affix-tag pairs that are unattested in the training data: it fires if the affix-match constraint is not satisfied but the analysis is deemed the most likely by DIRECTL+.

Features 7-9 refer to the *mirror* constraint defined in Section 6.1, in order to promote analyses that the inflector model correctly transduces back into the initial word-form. These three features follow the same pattern as the affix-match features.

6.1.2 Thresholding

Each word-form has at least one analysis, but the number of correct analyses varies; for example, *lüfte* has seven (Table 6.1). The system needs to decide where to "draw the line" between the correct and incorrect analyses in its n-best list. Apart from the top-1 analysis, the candidate analyses are filtered by a pair of thresholds which are defined as percentages of the top analysis score. The thresholds aim at reconciling two types of syncretism: one that involves multiple inflections of the same lemma, and the other that involves inflections of different lemmas. The first threshold is unconditional: it allows any analysis with a sufficiently high score. The second, lower threshold is conditional: it only allows a relatively high-scoring analysis if its lemma occurs in one of the analyses that clear the first threshold. For example, the fourth analysis in Table 6.3, *schrieben* + 2PKE, needs to clear both thresholds, because its

lemma differs from the lemmas of the analyses that clear the first threshold. Both thresholds are tuned on a development set.

6.2 Experiments

In this section, we evaluate our morphological analyzer on English, German, Dutch, and Spanish, and compare our results to two other systems.

6.2.1 Data

	English			German			Dutch			Spanish		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DirecTL+	**93.5**	88.9	**91.2**	**87.3**	**88.7**	**88.0**	**87.3**	**90.3**	**88.8**	**99.3**	**99.5**	**99.4**
Marmot	87.5	**94.3**	90.8	85.3	88.5	86.9	81.3	84.7	82.9	99.2	98.9	99.1

Table 6.5: Macro-averaged results on four languages.

We extract complete inflection tables for English, German, and Dutch from the CELEX lexical database (Baayen et al., 1995). The number of inflectional categories across verbs, nouns, and adjectives is 16, 50, and 24, respectively, in the three languages. However, in order to test whether an analyzer can handle arbitrary word-forms, the data is not separated according to distinct POS sets. For consistency, we ignore German noun capitalization.

CELEX data already exists in a format conducive to creating source / target pairs. We first collect all word-forms that share a lemma. The word-form itself forms the source form in the training data, with the lemma and provided inflectional tag forming the target. Training, development, and testing sets are constructed such that there is no lemma overlap between them.

The Spanish data is from Wiktionary inflection tables, as provided by Durrett and DeNero, 2013). and includes 57 inflectional categories of Spanish verbs. We convert accented characters to their unaccented counterparts followed by a special symbol (e.g. *cantó* → `canto'`), with no loss of information.

The data is split into 80/10/10 train/dev/test sets; for Spanish, we use the same splits as Durrett and DeNero, 2013). We eliminate duplicate identical word-forms from the test data, and hold out 20% of the development data to

System	P	R	F1
DirecTL+	**78.7**	**92.6**	**85.1**
Morphisto	65.1	52.7	58.2

Table 6.6: Micro-averaged results on German.

train the re-ranker. The training instances are randomly shuffled to eliminate potential biases.

For reranking, we extract word-form lists from the first one million lines of the November 2, 2015 Wikipedia dump for the given language, and derive our language models using the CMU Statistical Language Modeling Toolkit.[1]

6.2.2 Comparison to Morphisto

We first compare our German results against Morphisto (Zielinski and Simon, 2009), an FST analyzer. Beyond morphological analysis, Morphisto also performs some derivational analysis, converting compound segments back into lemmas. For a fair comparison, we exclude compounds from the test set. In addition, because the lexicon of Morphisto has a limited coverage, we report micro-averaged results in this section.

Table 6.6 shows that overall our system performs much better on the test sets than the hand-engineered Morphisto, which fails to analyze 43% of the word-forms in the test set. If we disregard the word-forms that Morphisto cannot handle, its F-score is actually higher: 89.5% vs. 84.0%.

6.2.3 Comparison to Marmot

Marmot (Mueller et al., 2013) is a state-of-the-art, publicly available morphological tagger[2], augmented with a lemmatizing module (Müller et al., 2015), which can also take advantage of unannotated corpora. In order to make a fair comparison, we train Marmot on the same data as our system, with default parameters. Because Marmot is a morphological tagger, rather than an analyzer, we provide the training and test word-forms as single-word sentences. In

[1] *http://www.speech.cs.cmu.edu*
[2] *http://cistern.cis.lmu.de/marmot*

addition, we have modified the source code to output a list of n-best analyses instead of a single best analysis. No additional reranking of the results is performed, as Marmot already contains its own module for leveraging a corpus, which is activated in these experiments. Separate thresholds for each language are tuned on the development sets. (c.f. Section 6.1.2).

Table 6.5 presents the results. We evaluate the systems using macro-averaged precision, recall, and F-score. Our system is consistently more accurate, improving the F-score on each of the four languages. Both systems make few mistakes on Spanish verbs.

The English results stand out, with Marmot achieving a higher recall at the cost of precision. English contains more syncretic forms than the other three languages: 3 different analyses per word-form on average in the test set, compared to 1.9, 1.3, and 1.1 for German, Dutch, and Spanish, respectively. Marmot's edit-tree method of candidate selection favors fewer lemmas, which allows the lemmatization module to run efficiently. On the other hand, DIRECTL+ has no bias towards lemmas or tags. This may be the reason of the substantial difference between the two systems on Dutch, where nearly a quarter of all syncretic test word-forms involve multiple lemmas.

An example of an incorrect analysis is provided by Spanish *lacremos*. Both systems correctly identify it as a plural subjunctive form of the verb `lacrar`. However, Marmot also outputs an alternative analysis that involves a bizarre lemma `lacr`. Our system is able to exclude this word-form thanks to a low score from the character language model, which is taken into consideration by the re-ranker.

6.3 Discussion

We have presented a transduction-based morphological analyzer that can be trained on plain inflection tables. Our system is highly accurate, and has a much higher coverage than a carefully-crafted FST analyzer. By eliminating the necessity of expert-annotated data, our approach may lead to the creation of analyzers for a wide variety of languages.

Now that we've shown that our methods are highly accurate modeling inflection in both directions, we combine our generation and lemmatization tools for the task of reinflection. The next chapter discusses our participation in the first Shared Task on Morphological Reinflection.

Chapter 7

Morphological Reinflection via Discriminative String Transduction.

Many languages have complex morphology with dozens of different word-forms for any given lemma. It is often beneficial to reduce the data sparsity introduced by morphological variation in order to improve the applicability of methods that rely on textual regularity. The task of inflection generation (Task 1) is to produce an inflected form given a lemma and desired inflection, which is specified as an abstract tag. The task of labeled reinflection (Task 2) replaces the input lemma with a morphologically-tagged inflected form. Finally, the task of unlabeled reinflection (Task 3) differs from Task 2 in that the input lacks the inflection tag. Reinflection removes a dependence on the lemma, generalizing the task to inflection from any form into any form. Such a generalization may prove beneficial in languages where the lemma is only used for traditional reasons, and bears little resemblance to many inflected forms.

In this chapter, we describe our system as participants in the SIGMORPHON 2016 Shared Task on Morphological Reinflection (Cotterell et al., 2016). We perform Task 1 using the inflection generation approach of Chapter 4, which we refer to as the *lemma-to-word* model. For Task 3, we first determine the best way to perform reinflection by evaluating the models from Chapter 5 on German, and using the best system to perform reinflection. We reduce Task 2 to Task 3 by simply ignoring the input inflection tag.

7.1 Methods

In this section, we describe the application of our string transduction and reranking approaches to the three shared tasks.

7.1.1 Task 1: Inflection

For Task 1, we derive a *lemma-to-word* model, which transforms the lemma along with an inflection tag into the inflected form. Our method models affixation with atomic morphological tags. For example, the training instance corresponding to the past participle *dado* of the Spanish verb *dar* "to give" consists of the source `dar+PP` and the target `dado`. The unsupervised M2M aligner matches the `+PP` tag with the `do` suffix on the basis of their frequent co-occurrence in the training data. DIRECTL+ then learns that the `PP` tag should be transduced into `do` when the lemma ends in `ar`. Similarly, prefixes are represented by a tag before the lemma. The transducer can also memorize stem changes that occur within the context of a tag. For example, the training pair `PP+singen+PP` → `gesungen` can inform the transduction `PP+ringen+PP` → `gerungen` at test time.

7.1.2 Task 2: Labeled Reinflection

Task 2 is to generate a target inflected form, given another inflected form and its tag. Since our current approach is not able to take advantage of the tag information, we disregard this part of the input, effectively reducing Task 2 to Task 3.

7.1.3 Task 3: Unlabeled Reinflection

In general, Task 3 appears to be harder than Tasks 1 and 2 because it provides neither the lemma nor the inflection tag for the given word-form. In essence, our approach is to first *lemmatize* the source word, and then proceed as with Task 1 as described in Section 7.1.1. We chain the *lemma-to-word* model from Task 1 with a *word-to-lemma* model, which is derived from the same data, but with the source and target sides swapped. The *word-to-lemma* model

transforms the inflected word-forms into sequences of lemmas and tags; e.g. `dado` → `dar+PP`.

The only difference between the two models involves empty affixes (e.g. the plural of *fish* in English). The *lemma-to-word* model can simply delete the tag on the source side, but the *word-to-lemma* model would need to insert it on the target side. In order to avoid the problem of unbounded insertions, we place a dummy *null* character at the boundaries of the word, effectively turning insertion into substitution.

Lemmatization is not the only method of inflection simplification; we experimented with three alternative approaches, as first described in Chapter 5:

1. *stem-based* approach, which is composed of the *word-to-stem* and *stem-to-word* models;

2. *stemma-based* approach, which instead pivots on stemmed lemmas;

3. *word-to-word* model, which directly transduces one inflected form into another.

We experiment with all three simplification methods before determining which one to use in our final system.

7.1.4 Corpus Reranking

The shared task is divided into three tracks that vary in the amount of information allowed to train reinflection models. Track 1 ("Standard") allows the training data from the corresponding or lower-numbered tasks. We did not participate in Track 2 ("Restricted") because it was formulated after the release of the training data. For Track 3 ("Bonus"), the shared task organizers provided unannotated text corpora for each language.

Our Track 3 approach is to rerank the n-best list of predictions generated by DirecTL+ for each test word-form using the method described in Section 2.3. For each language, we take the first one million lines from the corresponding Wikipedia dump as our corpus, removing the XML markup with the `html2text` utility. Our reranker contains three features:

1. normalized score of the prediction generated by DirecTL+;

2. presence in the corpus;

3. normalized log likelihood of the prediction given a 4-gram character language model derived from the corpus.

7.2 Language-Specific Heuristics

Each language has its own unique properties that affect the accuracy of reinflection. While our approach is designed to be language-independent, we also investigated modifications for improving accuracy on individual languages.

7.2.1 Spanish Stress Accents

In Spanish, vowels are marked to indicate irregular stress (e.g. *á* in *darás*). This introduces several additional characters that are phonetically related to their unaccented counterparts. In an attempt to generalize unstressed and stressed vowels, we represent each stressed vowel as a pair of an unaccented vowel and the stress mark. (e.g. *darás* becomes *dara's*). After inflecting the test word-forms, we reverse this process: any vowel followed immediately by a stress mark is replaced with the corresponding accented vowel; stress marks not following a vowel are deleted.

7.2.2 Vowel Harmony

In agglutinative languages such as Finnish, Turkish, and Hungarian, vowels in stems and suffixes often share certain features such as *height*, *backness*, or *rounding*. We augment DirecTL+ with features that correspond to vowel harmony violations. Since our development experiments demonstrated a substantial (13%) error reduction only for Turkish verbs, the vowel harmony features were restricted to that subset of the data.

7.2.3 Georgian Preverbs

Georgian verbs may include *preverb* morphemes, which act more like a derivational affix than an inflectional one. We observed that the Georgian training data contained many preverbs *da* and *ga*, but only some of the instances included the preverb on the lemma. This forced the models to learn two separate sets of rules. Removing these preverbs from the training word-forms and lemmas led to an 8% error reduction on the development set.

7.2.4 Arabic Sun Letters

In Arabic, consonants are divided into two classes: *sun* letters (i.e. coronal consonants) and *moon* letters (all others). When the definite article *al-* is followed by a sun letter, the letter *lām* assimilates to the following letter. Thus, *al+shams* "the sun" is realized as *ash-shams*. We observed that almost half of the errors on the adjectives could be attributed to this phenomenon. We therefore enforce this type of assimilation with a post-processing script.

7.3 Experiments

Our transduction models are trained on the pairs of word-forms and their lemmas. The *word-to-lemma* models (Section 7.1.1), are trained on the Task 1 training dataset, which contains gold-standard lemmas. These models are then employed in Tasks 2 and 3 for lemmatizing the source word-forms. The *lemma-to-word* models (Section 7.1.3) are derived from the training data of all three tasks, observing the Track 1 stipulations (Section 7.1.4). For example, the *lemma-to-word* models employed in Task 2 are trained on a combination of the gold-standard lemmas from Task 1, as well as the lemmas generated by the *word-to-lemma* models from the source word-forms in Task 2. Our development experiments showed that this kind of self-training approach can improve the overall accuracy.[1]

[1]Because of time constraints, we made an exception for Maltese by training on the gold lemmas from Task 1 only.

	Task 1	Task 3
Baseline	89.4	81.5
Chipmunk	82.0	88.3
Stem-based	86.9	89.3
Stemma-based	84.0	89.5
Lemma-based	n/a	90.7
Source-Target	94.8	88.2

Table 7.1: Accuracy on the German dataset using alternative methods of morphological simplification.

7.3.1 Lemmatization method

In order to determine the best morphological simplification method for reinflection, we evaluate four different methods that combine the models introduced in Chapter 5. For Task 1, the stem-based method chains a *lemma-to-stem* and a *stem-to-word* model; the stemma-based method is similar, but pivots on stemmas instead; and the source-target method is a *lemma-to-word* model. For Task 3, a *lemma-to-stem* and *lemma-to-stemma* models are replaced by *word-to-stem* and *word-to-stemma* models, respectively. The *lemma-based* method chains a *word-to-lemma* and a *lemma-to-word* model; and the source-target method is a *word-to-word* model with no simplification. In addition, we compare with a method that is similar to our *stem-based* method, but pivots on Chipmunk-generated stems instead. As a baseline, we run the transduction method provided by the task organizers.

The results are shown in Table 7.1. On Task 1, none of the stemming approaches is competitive with a direct *lemma-to-word* model. This is not surprising. First, the lemmatic suffixes provide information regarding part-of-speech. Second, the stemmers fail to take into account the fact that the source word-forms are lemmas. For example, the German word *überhitzend* "overheated" can either be an adjective, or the present participle of the verb *überhitzen*; if the word is a lemma, it is obviously the former.

The *lemma-based* method is the best performing one on Task 3. One advantage that it has over the *word-to-word* model lies in the ability to reduce the potentially quadratic number of transduction operations between various

related word-forms to a linear number of transduction operations between the word-forms and their lemmas, and vice-versa. For the remaining experiments in this chapter, we use the *lemma-based* method.

7.3.2 Development Results

Selected development results are shown in Table 7.2. The Task 1 results are broken down by part-of-speech. Because of an ambiguity in the initial shared task instructions, all development models were trained on a union of the data from all three tasks.

	Task 1	Task 2	Task 3	VB	NN	ADJ
ES	98.0	96.3	96.3	96.0	95.9	100
DE	94.4	92.2	92.2	90.5	88.6	97.7
FI	90.0	88.4	88.4	92.1	89.7	63.9
RU	89.5	86.3	86.3	81.9	91.7	96.7
TR	78.6	74.9	74.9	78.8	78.5	n/a
KA	96.8	95.5	95.5	62.9	99.0	99.2
NV	91.3	90.0	90.0	88.5	99.1	n/a
AR	81.1	76.2	76.2	85.7	61.2	84.6

Table 7.2: Word accuracy on the development sets.

7.3.3 Test Results

Table 7.3 shows our test results. In most cases, these results are close to our development results. One exception is Navajo, where the test sets were significantly harder than the development sets. We also note drops in accuracy from Task 1 to Task 2 and 3 that were not evident in development, particularly for Arabic and Turkish. The drops can be attributed to the different training conditions between development and testing. In Section 7.4, we describe language specific issues; Arabic and Turkish were particularly affected by less training data.

Table 7.3 also contains the results for the "Bonus" track (RR). The reranking yields an improvement in almost all cases. Arabic is a clear exception. The data provided for the task was presented in a transliterated Latin script, while the Wikipedia corpus was in the original Arabic text. While a transliterated

	Task 1		Task 2		Task 3	
	ST	RR	ST	RR	ST	RR
ES	97.8	98.0	96.2	96.4	96.5	96.6
DE	94.1	93.8	91.1	91.6	91.1	91.6
FI	88.5	88.7	85.6	85.7	85.8	85.9
RU	88.6	89.7	85.5	86.6	85.5	86.6
TR	82.2	87.5	62.5	59.2	63.1	59.2
KA	96.1	96.3	94.1	94.2	94.1	94.4
NV	60.3	60.3	50.4	50.8	48.8	49.1
AR	82.1	53.1	71.8	44.1	72.2	58.5
HU	86.7	89.6	86.3	88.8	86.4	88.9
MT	42.0	42.5	37.5	37.8	37.5	37.8

Table 7.3: Word accuracy on the test sets.

version of the text was eventually provided, it was not a complete transliteration: certain vowels were omitted, as they are difficult to recover from standard Arabic. This affected our reranker because it depends on correct forms in the corpus and a character language model.

7.4 Error Analysis

In this section, we discuss a few types of errors that we observed on the development sets for each language.

Spanish The highest overall accuracy among the tested languages confirms its reputation of morphological regularity. A handful of verb errors are related to the interplay between orthography and phonology. Our models appear to have difficulty generalizing the rigid rules governing the representation of the phonemes [k] and [θ] by the letters *q*, *c* and *z*. For example, the form *crucen*, pronounced [kruθɛn], is incorrectly predicted with *z* instead of *c*, even though the bigram *ze* is never observed in Spanish. This demonstrates that the character language model feature of the reranker is not able to completely prevent orthographically-invalid predictions.

German Nouns and verbs fall into several different inflectional classes that are difficult to predict from the orthography alone. For example, the plural

of *Schnurrbart*, "moustache", is *Schnurrbärte*. Our system incorrectly misses the umlaut, applying the pluralization pattern of the training form *Wart*, "attendant", which is indeed pluralized without the umlaut.

Finnish A phenomenon known as consonant gradation alternates variants of consonants depending on their context. Given the amount of the training data, our method is unable to learn all of the appropriate gradation contexts.

Russian The results indicate that verbs are substantially more challenging than nouns and adjectives. Most of the errors involve vowel changes. The reranker reduces the error rate by about 10% on Task 1. In particular, it filters out certain predictions that appear to violate phonotactic constraints, and reduces the number of errors related to lexically-conditioned prefixes in the perfective forms.

Turkish Occasionally, the forms in crowd-sourced data are incorrect, which can lead to spurious transduction rules both during lemmatization and inflection. For example, the form *çıkaracağım* of the verb *çıkarmak* "to subtract" is erroneously associated in the training data with the lemma *toplamak* "to add", which causes the *word-to-lemma* model to learn a spurious `çı` → `to` rule. At test time, this leads to incorrect lemma predictions, which in turn propagate to multiple inflected forms.

Georgian The highly unpredictable preverbs (Section 7.2.3) were the cause of a large number of errors on verbs. On the other hand, our system did very well on nouns and adjectives, second only to Spanish.

Arabic Errors were mainly constrained to irregular forms, such as the nominal *broken plurals*. Unlike *sound plurals* that inflect via suffixation, broken plurals involve consonantal substitution. This is a difficult transduction to learn, given its low frequency in training. Another type of errors involves *weak roots*, which contain semi-vowels rather than full consonants.

Navajo In contrast with the test results, our development results were very promising, with near-perfect performance on nouns. After the submission deadline, we were informed that the test set differed in significant ways from the training and development sets, which lead to increased difficulty for this language.

Hungarian As it was one of the surprise languages, we applied no language-specific techniques. Nevertheless, the test results were on par with the other agglutinative languages. We speculate that adding customized vowel harmony features could further improve the results.

Maltese A complicated morphology is represented by an extremely large tag set (3184 distinct tags). For nouns and adjectives, the number of tags is very close to the number of training instances, which precludes any meaningful learning generalization. While many features within tags are repeated, taking advantage of this regularity would require more development time, which was unavailable for the surprise languages. The results highlight a limitation of the atomic tags in our method.

7.5 Discussion

Previous work in morphological generation was largely limited to a small number of western European languages. The methods proposed in Chapter 4 were originally developed on such languages. The results on the shared task data show that those methods can be adapted to the task of reinflection, and perform well on various morphologically-complex languages. On the other hand, there is room for improvement on languages like Maltese, which provides motivation for future work.

Chapter 8

Conclusions

In this dissertation, we investigate several problems concerning inflection, and propose methods to model them computationally using sequential string transduction.

In Chapter 4, we achieve state-of-the-art results in inflection generation by training our transducer on inflection tables. The results reported in Chapter 4 remain among the state-of-the-art: Faruqui et al., 2016) compare these results against a neural model, and are unable to better our results on five of seven languages sets.

In Chapter 5, we turn our transduction methods to inflectional simplification. We show that with a small number of modifications, the methods that produce state-of-the-art inflection generation models can also produce high quality lemmatizers. We also demonstrate that although inflection tables include no explicit stemming annotation, there is enough implicit information to obtain accurate, consistent stems, using the same tools used to produce inflections and lemmas.

Chapter 6 goes beyond lemmatization; we demonstrate that our transduction methods can also predict inflectional categories with high accuracy, producing analyses that improve upon the previous state of the art. Like our generation and stemming methods, we do not require expertly-annotated training corpora, but instead analyze words after training on corpora that can be constructed by speakers with little explicit knowledge.

In Chapter 7, we show that our lemmatization and generation methods are

suitable across a variety of languages, while also demonstrating how explicit linguistic knowledge can be used to modify methods for individual languages.

At this point, we make no effort to combine our methods with neural sequential models, such as (Bahdanau et al., 2014) or Kann and Schütze, 2016), which have shown large improvements in similar tasks. As a future direction for study, we propose a combination of our methods with neural methods, which can take advantage of the positive aspects of both systems. Many of the gains seen in neural models can be attributed to an increased context provided by attention mechanisms, an ability to share information across inflectional classes, and the ability to ensemble multiple systems. However, what these systems gain in accuracy, they lose in transparency. By combining neural models with our own, we can preserve the transparency of our system, while gaining access to the benefits of the neural model.

We live in an increasingly international community that will require more and more processing of languages with inflectional complexity that far outstrips that of English. The problem of what to do with inflectional morphology is far from solved, but we have shown that it is possible to use sequential prediction tools to accurately model inflectional processes such as those given in Figure 1.2: inflection generation, stemming, lemmatization, and analysis can be modeled via sequential string prediction, and require no more training data than publicly available inflection tables.

Bibliography

[Ahlberg et al., 2014] Ahlberg, M., Forsberg, M., and Hulden, M. (2014). Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden. Association for Computational Linguistics.

[Ahlberg et al., 2015] Ahlberg, M., Forsberg, M., and Hulden, M. (2015). Paradigm classification in supervised learning of morphology. In *HLT-NAACL*, pages 1024–1029.

[Baayen et al., 1995] Baayen, H. R., Piepenbrock, R., and Gulikers, L. (1995). *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Böhmová et al., 2003] Böhmová, A., Hajič, J., Hajičová, E., and Hladká, B. (2003). The Prague dependency treebank. In *Treebanks*, pages 103–127. Springer.

[Booij, 2012] Booij, G. (2012). *The grammar of words: An introduction to linguistic morphology*. Oxford University Press.

[Chrupala et al., 2008] Chrupala, G., Dinu, G., and van Genabith, J. (2008). Learning morphology with Morfette. In *LREC 2008*.

[Clifton and Sarkar, 2011] Clifton, A. and Sarkar, A. (2011). Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 32–42.

[Cotterell et al., 2016] Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., and Hulden, M. (2016). The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22. Association for Computational Linguistics.

[Cotterell et al., 2015] Cotterell, R., Müller, T., Fraser, A., and Schütze, H. (2015). Labeled morphological segmentation with semi-Markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174. Association for Computational Linguistics.

[Cotterell et al., 2017] Cotterell, R., Sylak-Glassman, J., and Kirov, C. (2017). Neural graphical models over strings for principal parts morphological paradigm completion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 759–765. Association for Computational Linguistics.

[Creutz and Lagus, 2002] Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30.

[Creutz and Lagus, 2004] Creutz, M. and Lagus, K. (2004). Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 43–51.

[Creutz and Lagus, 2005] Creutz, M. and Lagus, K. (2005). Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, volume 1(106-113), pages 51–59.

[Détrez and Ranta, 2012] Détrez, G. and Ranta, A. (2012). Smart paradigms and the predictability and complexity of inflectional morphology. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–653. Association for Computational Linguistics.

[Dreyer and Eisner, 2011] Dreyer, M. and Eisner, J. (2011). Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 616–627. Association for Computational Linguistics.

[Durrett and DeNero, 2013] Durrett, G. and DeNero, J. (2013). Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195. Association for Computational Linguistics.

[Eskander et al., 2013] Eskander, R., Habash, N., and Rambow, O. (2013). Automatic extraction of morphological lexicons from morphologically annotated corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1032–1043, Seattle, Washington, USA. Association for Computational Linguistics.

[Fan et al., 2008] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

[Faruqui et al., 2016] Faruqui, M., Tsvetkov, Y., Neubig, G., and Dyer, C. (2016). Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643. Association for Computational Linguistics.

[Fraser et al., 2012] Fraser, A., Weller, M., Cahill, A., and Cap, F. (2012). Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674. Association for Computational Linguistics.

[Garnett, 1912] Garnett, J. (1912). *Beowulf, an Angol-Saxon poem, and The Fight at Finnsburg; translated by James M. Garnett*. Boston Ginn, Boston, Massachusetts.

[Goldsmith, 2001] Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

[Grönroos et al., 2014] Grönroos, S.-A., Virpioja, S., Smit, P., and Kurimo, M. (2014). Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185. Dublin City University and Association for Computational Linguistics.

[Hajič et al., 2009] Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, A. M., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štepánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18. Association for Computational Linguistics.

[Hauer et al., 2017] Hauer, B., Nicolai, G., and Kondrak, G. (2017). Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 619–624. Association for Computational Linguistics.

[Jiampojamarn et al., 2010] Jiampojamarn, S., Cherry, C., and Kondrak, G. (2010). Integrating joint n-gram features into a discriminative training framework. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 697–700. Association for Computational Linguistics.

[Jiampojamarn et al., 2007] Jiampojamarn, S., Kondrak, G., and Sherif, T. (2007). Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *NAACL-HLT*, pages 372–379.

[Joachims, 2002] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

[Kann and Schütze, 2016] Kann, K. and Schütze, H. (2016). *MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection*, pages 62–70. Association for Computational Linguistics.

[Liu and Mao, 2016] Liu, L. and Mao, J. L. (2016). *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, chapter Morphological reinflection with conditional random fields and unsupervised features, pages 36–40. Association for Computational Linguistics.

[McDonald et al., 2005] McDonald, R., Crammer, K., and Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98. Association for Computational Linguistics.

[Mueller et al., 2013] Mueller, T., Schmid, H., and Schütze, H. (2013). Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332. Association for Computational Linguistics.

[Müller et al., 2015] Müller, T., Cotterell, R., Fraser, A., and Schütze, H. (2015). Joint lemmatization and morphological tagging with Lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274. Association for Computational Linguistics.

[Nicolai et al., 2015a] Nicolai, G., Cherry, C., and Kondrak, G. (2015a). Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931. Association for Computational Linguistics.

[Nicolai et al., 2015b] Nicolai, G., Cherry, C., and Kondrak, G. (2015b). Morpho-syntactic regularities in continuous word representations: A multilingual study. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 129–134. Association for Computational Linguistics.

[Nicolai et al., 2015c] Nicolai, G., Hauer, B., Salameh, M., St Arnaud, A., Xu, Y., Yao, L., and Kondrak, G. (2015c). Multiple system combination for transliteration. In *Proceedings of the Fifth Named Entity Workshop*, pages 72–77. Association for Computational Linguistics.

[Nicolai et al., 2013] Nicolai, G., Hauer, B., Salameh, M., Yao, L., and Kondrak, G. (2013). Cognate and misspelling features for natural language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 140–145. Association for Computational Linguistics.

[Nicolai et al., 2016a] Nicolai, G., Hauer, B., St Arnaud, A., and Kondrak, G. (2016a). Morphological reinflection via discriminative string transduction. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 31–35. Association for Computational Linguistics.

[Nicolai and Kondrak, 2014] Nicolai, G. and Kondrak, G. (2014). Does the phonology of L1 show up in L2 texts? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 854–859. Association for Computational Linguistics.

[Nicolai and Kondrak, 2015] Nicolai, G. and Kondrak, G. (2015). English orthography is not "close to optimal". In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 537–545. Association for Computational Linguistics.

[Nicolai and Kondrak, 2016] Nicolai, G. and Kondrak, G. (2016). Leveraging inflection tables for stemming and lemmatization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1138–1147. Association for Computational Linguistics.

[Nicolai and Kondrak, 2017] Nicolai, G. and Kondrak, G. (2017). Morphological analysis without expert annotation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 211–216. Association for Computational Linguistics.

[Nicolai et al., 2016b] Nicolai, G., Yao, L., and Kondrak, G. (2016b). Morphological segmentation can improve syllabification. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 99–103. Association for Computational Linguistics.

[Pirinen, 2015] Pirinen, A. T. (2015). Omorfi — free and open source morphological lexical database for finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 313–315. Linköping University Electronic Press, Sweden.

[Poon et al., 2009] Poon, H., Cherry, C., and Toutanova, K. (2009). Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217. Association for Computational Linguistics.

[Porter, 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

[Ristad and Yianilos, 1998] Ristad, E. S. and Yianilos, P. N. (1998). Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532.

[Ruokolainen et al., 2014] Ruokolainen, T., Kohonen, O., Virpioja, S., and kurimo, m. (2014). Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84–89. Association for Computational Linguistics.

[Sennrich et al., 2009] Sennrich, R., Schneider, G., Volk, M., and Warin, M. (2009). A new hybrid dependency parser for German. *Proceedings of the German Society for Computational Linguistics and Language Technology*.

[Toutanova and Cherry, 2009] Toutanova, K. and Cherry, C. (2009). A global model for joint lemmatization and part-of-speech prediction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 486–494. Association for Computational Linguistics.

[Zens and Ney, 2004] Zens, R. and Ney, H. (2004). Improvements in phrase-based statistical machine translation. In *HLT-NAACL 2004: Main Proceedings*, pages 257–264, Boston, USA.

[Zielinski and Simon, 2009] Zielinski, A. and Simon, C. (2009). Morphisto-an open source morphological analyzer for German. In Piskorski, J., Watson, B., and Yli-Jyrä, A., editors, *Finite-state Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP*, volume 191, page 224. IOS Press.