Leveraging Inflection Tables for Stemming and Lemmatization

Garrett Nicolai and **Grzegorz Kondrak**
Department of Computing Science
University of Alberta
{nicolai,gkondrak}@ualberta.ca

Abstract

We present several methods for stemming and lemmatization based on discriminative string transduction. We exploit the paradigmatic regularity of semi-structured inflection tables to identify stems in an unsupervised manner with over 85% accuracy. Experiments on English, Dutch and German show that our stemmers substantially outperform Snowball and Morfessor, and approach the accuracy of a supervised model. Furthermore, the generated stems are more consistent than those annotated by experts. Our direct lemmatization model is more accurate than Morfette and Lemming on most datasets. Finally, we test our methods on the data from the shared task on morphological reinflection.

1 Introduction

Many languages contain multiple inflected forms that correspond to the same dictionary word. Inflection is a grammatical procedure that has little impact on the meaning of the word. For example, the German words in Table 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.

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

Table 1: Examples of German word-forms corresponding to the lemma *geben*.

In this paper, 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 2, in an unsupervised manner. We design two transduction models that are trained on such stems, and evaluate them on unseen 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. Finally, we test our methods on the data from the shared task on morphological reinflection.

This paper is organized as follows. In Section 2, we present an overview of prior work on inflectional simplification. In Section 3, we describe our stemming methodology, followed by three types of evaluation experiments in Section 4. In Section 5, we describe our approach to lemmatization, followed by both intrinsic and extrinsic experiments in Section 6. Section 7 concludes the paper.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1138–1147,
Berlin, Germany, August 7-12, 2016.

2 Related Work

In this section, we review prior work on stemming and lemmatization.

2.1 Stemming and Segmentation

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, 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.

Morfessor (Creutz and Lagus, 2002) and Linguistica (Goldsmith, 2001) 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.

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 take advantage of linguistic knowledge, without requiring costly annotation.

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 is inherently supervised, while our stem annotations are derived in an unsupervised manner.

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, effectively combining morphological segmentation and morphological analysis. 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.

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

2.2 Lemmatization

Unlike stemmers, which can be unsupervised, lemmatizers typically require annotated training data. In addition, some lemmatizers assume access to the morphological tag of the word, and/or the surrounding words in the text. Our focus is on context-free lemmatization, which could later be combined with a contextual disambiguation module.

Lemmatization is often part of the morphological analysis task, which aims at annotating each word-form with its lemma and morphological tag. Toutanova and Cherry (2009) learn a joint model for contextual lemmatization and part-of-speech prediction from a morphologically annotated lexicon. Their transduction model is tightly integrated with the POS information, which makes comparison difficult. However, in Section 6, we evaluate our approach against two other fully-supervised morphological analyzers: Morfette (Chrupała et al., 2008) and Lemming (Müller et al., 2015). Both of these systems perform lemmatization and morphological analysis in context, but can be trained to learn non-contextual models. Morfette requires morphological tags during training, while Lemming requires a morphological model constructed by its sister program, Marmot (Müller et al., 2013).

3 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`.

3.1 Character Alignment

The training of a transduction model requires a set of aligned pairs of source and target strings. The alignment involves every input and output character; the insertion and deletion operations are disallowed. Atomic character transformations are then extracted from the alignments.

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. For our task, the source and target strings are nearly identical, except that the target includes stem-affix boundary markers. In order to account for every character in the target, which is usually longer than the source, we allow one-to-many alignment. This has the effect of tying the markers to the edge of a stem or affix. In order to encourage alignments between identical characters, we modify the aligner to generalize all identity transformations into a single match operation.

3.2 Supervised Transduction

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 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.

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.

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

DIRECTL+ uses a number of feature templates to assess the quality of a rule: source context, target n-gram, and joint n-gram features. Context features conjoin the rule with indicators for all source character n-grams within a fixed window of where the rule 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 rule sequences, combining source and target context, and memorizing frequently-used rule patterns.

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.

3.3 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 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 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.

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

Table 4: Alignment of the various representations of the word *gibt*.

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

Table 5: The number of words and distinct inflections for each language in the CELEX datasets.

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 3). We then apply the unsupervised 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 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 3.2. We refer to this unsupervised approach as our basic method (cf. Figure 1).

3.4 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 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 3.3, the stems are replaced with the corresponding stemmas, and the affixes are replaced with the inflection tags. For example, the form *gibt* is paired with the sequence `geb+3SIE`, with the stem and stemma re-aligned at the character level as shown in Table 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.

4 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 af-

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

	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 6: Unsupervised stemming accuracy of the CELEX training set.

fixes.[2] Chipmunk (Cotterell et al., 2015), is a fully-supervised system that represents the current state of the art.

4.1 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). 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).

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.

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.

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

4.2 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 3.3 match the stem annotations in the CELEX database.

The results are presented in Table 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`.

4.3 Intrinsic Evaluation

The results of the intrinsic evaluation of the stemming accuracy on unseen forms in Tables 7-9 demonstrate the quality of our three models. The joint 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 7).[4]

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

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

[4] The decrease in Morfessor accuracy between Tables 6 and 7 can be attributed to a different POS distribution between training and testing.

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

Table 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 9: Stemming accuracy of systems trained and tested on the Chipmunk data.

These results are further confirmed on the German Wiktionary data (Table 8). Our supervised model performs almost as well as Chipmunk on its dataset (Table 9).

A major advantage of the joint model over the basic model is its tag awareness (cf. Table 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.

4.4 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 many cases, this is not an attainable goal because of internal stem changes (cf. Table 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]

	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 10: Average number of stems per lemma.

The results are presented in Table 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 Lemmatization Methods

In this section, we present three supervised lemmatization methods, two of which incorporate the unsupervised stemming models described in Section 3. The different approaches are presented schematically in Figure 1, using the example of the German past participle *gedacht*.

5.1 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 3 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 3.3.

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

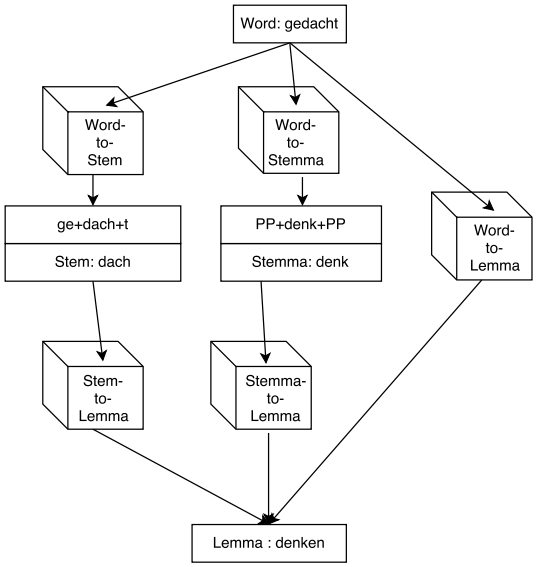

Figure 1: Three lemmatization methods.

5.2 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 3.4 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 3.4. Typically, the model simply appends a lemmatic affix to the stemma, as all stem changes are handled by the *word-to-stemma* model.

5.3 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, which are easily obtained from the inflection tables. A potential advantage of this 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.

5.4 Re-ranking

Intuitively, lemmatization accuracy could be improved by leveraging large, unannotated corpora. After generating n-best lists of possible lemmas, we re-rank them using the method of Joachims (2002) implemented with the Liblinear SVM tool (Fan et al., 2008). 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.

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

6.1 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 4.1. 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 re-ranking, 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.

Spanish marks unpredictable stress by marking a stressed vowel with an acute accent (e.g. *cantó*

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

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

	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 11: Lemmatization results without the use of a corpus.

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.

6.2 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 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 12. We omit the results on Spanish Wiktionary and on both English datasets, which are almost identical to those in Table 11. We observe that both the stemma-based and 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.

	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 12: Lemmatization results boosted with a raw corpus.

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.

6.3 Extrinsic Evaluation

We perform our final evaluation experiment on the German dataset[10] from the SIGMORPHON shared task on morphological reinflection (Cot-

[8] The capitalization of German nouns is ignored.

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

[10] http://sigmorphon.org/sharedtask

	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 13: Accuracy on the German dataset from the shared task on morphological reinflection.

terell et al., 2016).[11] The task of inflection generation (Task 1) is to produce a word-form given a lemma and an abstract inflectional tag. The task of unlabeled reinflection (Task 3) takes as input an unannotated inflected form instead of a lemma.

We evaluate four different methods that combine the models introduced in this paper. For Task 1, the stem-based method composes a *lemma-to-stem* and a *stem-to-word* models; 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 *word-to-lemma* model is added in front of both the stem-based and stemma-based methods; the lemma-based method composes a *word-to-lemma* and a *lemma-to-word* models; and the source-target method is a *word-to-word* model. 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 13. 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.

[11] We use the development sets for this evaluation because the target sides of the test sets have not been publicly released.

7 Conclusion

We have presented novel methods that leverage readily available inflection tables to produce high-quality stems and lemmas. In the future, we plan to expand our method to predict morphological analyses, as well as to incorporate other information such as parts-of-speech.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the Alberta Innovates Technology Futures.

References

Harald R. Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.

Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. 2008. Learning morphology with Morfette. In *LREC*.

Ryan Cotterell, Thomas Müller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. *CoNLL 2015*, page 164.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *SIGMORPHON*.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30.

Mathias Creutz and Krista Lagus. 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.

Mathias Creutz and Krista Lagus. 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 (AKRR05)*, volume 1(106-113), pages 51–59.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *HLT-NAACL*, pages 1185–1195.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

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

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *COLING*, pages 1177–1185.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *CoNLL*, pages 1–18.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *NAACL-HLT*, pages 372–379.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training network. In *NAACL-HLT*.

Thorsten Joachims. 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.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *EMNLP*, pages 322–332.

Thomas Müller, Ryan Cotterell, and Alexander Fraser. 2015. Joint lemmatization and morphological tagging with LEMMING. In *EMNLP*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL-HLT*, pages 209–217.

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

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. *EACL*, page 84.

Kristina Toutanova and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *ACL*, pages 486–494.