Morphological Reinflection via Discriminative String Transduction

Garrett Nicolai, Bradley Hauer, Adam St Arnaud, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{nicolai,bmhauer,ajstarna,gkondrak}@ualberta.ca

Abstract

We describe our approach and experiments in the context of the SIGMORPHON 2016 Shared Task on Morphological Reinflection. The results show that the methods of Nicolai et al. (2015) perform well on typologically diverse languages. We also discuss language-specific heuristics and errors.

1 Introduction

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 labelled reinflection (Task 2) replaces the input lemma with a morphologically-tagged inflected form. Finally, the task of unlabelled reinflection (Task 3) differs from Task 2 in that the input lacks the inflection tag.

In this paper, we describe our system as participants in the SIGMORPHON 2016 Shared Task on Morphological Reinflection (Cotterell et al., 2016). Our approach is based on discriminative string transduction performed with a modified version of the DIRECTL+ program (Jiampojamarn et al., 2008). We perform Task 1 using the inflection generation approach of Nicolai et al. (2015), which we refer to as the *lemma-to-word* model. We also derive a reverse *word-to-lemma* (lemmatization) model from the Task 1 data. We perform Task 3 by composing the *word-to-lemma* and *lemma-to-word* models. We reduce Task 2 to Task 3 by simply ignoring the input inflection tag.

2 Methods

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

2.1 String Transduction

We perform string transduction by adapting DIRECTL+, a tool originally designed for grapheme-to-phoneme conversion.[1] DIRECTL+ is a feature-rich, discriminative character string 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.

From aligned source-target pairs, DIRECTL+ extracts statistically-supported feature templates: 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. We train separate models for each part of speech in the training data.

We perform source-target pair alignment with a modified version of the M2M aligner (Jiampojamarn et al., 2007). The program applies the Expectation-Maximization algorithm with the ob-

[1] https://code.google.com/p/directl-p

jective to maximize the joint likelihood of its aligned source and target pairs. In order to encourage alignments between identical characters, we modify the aligner to generalize all identity transformations into a single match operation.

2.2 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.

2.3 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.

2.4 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 2.2. We compose 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 (Nicolai and Kondrak, 2016):

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.

However, as the *lemma-based* method obtained the best accuracy during development, we decided to use it for all following experiments.

2.5 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 of Joachims (2002). 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.

3 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.

3.1 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.

3.2 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.

3.3 Georgian Preverbs

Georgian verbs may include *preverb* morphemes, which act more like a derivational affix than an inflectional one. These preverbs primarily distinguish present and future tenses, but can also convey directional meaning. 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.

3.4 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.

4 Experiments

Our transduction models are trained on the pairs of word-forms and their lemmas. The *word-to-lemma* models (Section 2.2), 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 2.4) are derived from the training data of all three tasks, observing the Track 1 stipulations (Section 2.5). 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.[2]

4.1 Development Results

Selected development results are shown in Table 1. 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.

	T1	T2	T3	VB	NN	JJ
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 1: Word accuracy on the development sets.

4.2 Test Results

Table 2 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

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

	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 2: Word accuracy on the test sets.[3]

between development and testing. In Section 5, we describe language specific issues; Arabic and Turkish were particularly affected by less training data.

Table 2 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 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.

5 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 3.3) 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.

[3] The results in italics were obtained after the shared task submission deadline.

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.

6 Conclusion

Previous work in morphological generation was largely limited to a small number of western European languages. The methods proposed by Nicolai et al. (2015) for the task of inflection generation 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.

Acknowledgements

We thank Mohammad Salameh for his help with Arabic. We thank the organizers for their hard work preparing the task, their readiness to answer questions, and their flexibility with regards to complications.

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

References

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

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. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*, pages 905–913.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142. ACM.

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

Garrett Nicolai and Grzegorz Kondrak. 2016. Leveraging inflection tables for stemming and lemmatization. In *ACL*.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *NAACL-HLT*, pages 922–931.