If you can't beat them, join them: the University of Alberta system description

Garrett Nicolai, Bradley Hauer, Mohammad Motallebi, Saeed Najafi, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{nicolai,bmhauer,motalleb,snajafi,gkondrak}@ualberta.ca

Abstract

We describe our approach and experiments in the context of the CoNLL-SIGMORPHON 2017 Shared Task on Universal Morphological Reinflection. We combine a discriminative transduction system with neural models. The results on five languages show that our approach works well in the low-resource setting. We also investigate adaptations designed to handle small training sets.

1 Introduction

In this paper, we describe our system as participants in the CoNLL-SIGMORPHON 2017 Shared Task on Universal Morphological Reinflection (Cotterell et al., 2017). Our focus is on the sub-task of inflection generation under the low-resource scenario, in which the training data is limited to 100 labeled examples, with and without monolingual corpora. Our principal approach follows Nicolai et al. (2015), performing discriminative string transduction with a modified version of the DIRECTL+ program (Jiampojamarn et al., 2008). Taking into account the results of the SIGMORPHON 2016 Shared Task on Morphological Reinflection (Cotterell et al., 2016), we investigate ways to combine the strengths of DIRECTL+ with those of neural models. In addition, we experiment with various adaptations designed to handle small training sets, such as splitting and reordering morphological tags, and synthetic training data.

We derive inflection models for five languages: English, German, Persian, Polish, and Spanish. These languages display varying degrees of inflectional complexity, but are mostly suffixing, fusional languages. We combine three systems for each language: a discriminative transduction system, an ensemble of neural encoder-decoder models, and the affix-matching baseline provided by the task organizers. We test two methods of system combination: linear combination and an SVM reranker. The results demonstrate that our transduction approach is strongly competitive in the low-resource setting. Further gains can be obtained via tag reordering heuristics and system combination.

2 Methods

We follow Nicolai et al. (2015, 2016) in approaching inflection generation as discriminative string transduction. After aligning source lemmas to target word forms, conversion operations are extracted and applied to transform a lemma-tag sequence into an inflected form. In this section, we describe several novel adaptations to the low-resource setting, as well as the system combination methods.

2.1 String transduction

We perform string transduction with a modified version of 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, our version of DIRECTL+ extracts statistically-supported feature templates: source context, target n-gram, and joint

[1] https://github.com/GarrettNicolai/DTL

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 also add an abstract copy feature that corresponds to preserving the source characters unchanged.

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 objective to maximize the conditional 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, which corresponds to the transduction copy feature.

2.2 Tag splitting

Training instances in the inflection generation task consist of a lemma and a tag sequence which specifies the inflection slot. Tag sequences consist of smaller units, which we refer to as *subtags*, that determine specific aspects of the inflection. For example, the tag sequence “V;PTCP;PST;FEM;SG” indicates that the target form is a verbal (V) feminine (FEM) singular (SG) past (PST) participle (PTCP).

In the small training data scenario, it is not practical to treat tag sequences as atomic units, as we did in Nicolai et al. (2016), because many tag sequences may be represented by only a single training instance, or not at all. We follow Kann and Schütze (2016) in separating each tag sequence into its component subtags, in order to share information across inflection slots. Our system treats each subtag as an indivisible atomic symbol. An example is shown in Figure 1.

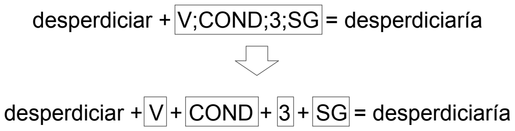
desperdiciar + V;COND;3;SG = desperdiciaría
desperdiciar + V + COND + 3 + SG = desperdiciaría

Figure 1: Splitting a tag into subtags to mitigate data sparsity.

From the linguistic point of view, tag splitting may seem counter-intuitive, as composite inflectional affixes in fusional languages can rarely be separated into individual morphemes. However, on the character level, many affixes share letter substrings across inflection slots. For example, the Spanish word *lavemos* could be analyzed as `lav+e+mos`, where the three substrings correspond to the stem, the subjunctive marker, and the first-person ending, respectively. In the single-tag setting, a model must learn the subjunctive inflection for each person; in the split-tag setting, the model can learn the subjunctive modification separately from the personal suffixes.

After splitting the tags, we perform an additional operation of prepending the part-of-speech symbol to each subtag, in order to distinguish between identically named subtags that correspond to different parts of speech (e.g., V:SG vs. N:SG).

2.3 Subtag reordering

Because our alignment and transduction systems are monotonic, tag splitting introduces the issue of subtag ordering. The provided data files are not always consistent in terms of the relative order in which subtags appear in sequence. We enforce the consistency by establishing a global ordering of all subtags in a given language. Our objective is to make as few changes as possible with respect to the original tag sequences. We achieve this by adapting the set ordering algorithm of Hauer and Kondrak (2016), which uses a beam search to minimize the number of subtag swaps within the tag sequences. We then reorder all tag sequences that are inconsistent with the resulting ordering. Our development experiments suggest that the consistent ordering never leads to a decrease in accuracy with respect to the original ordering.

We also investigate ways of optimizing the subtag order. For example, it would make sense for the gender subtag to precede the number subtag in Spanish past participles (e.g., *cortadas*). Since the number of possible orderings is exponential, testing a separate transduction model for each of them is infeasible. Instead, we consider the five orderings with the highest M2M-aligner alignment score on the training set, and select the one that results in the highest accuracy on the development set.

2.4 Particle handling

Some languages, including Spanish, German, and Polish, contain particles that complicate the inflection process. For example, some Spanish verbs contain the reflexive particle *se* (e.g. *levantarse*), which may be detached, inflected, and moved to the front (e.g. *me levanto*). In order to simplify our inflection model, we treat these particles as atomic characters. In this approach, *se* is a single-symbol affix of the lemma which is substituted by *me* and transposed in the output sequence. These particles were identified via language-specific rules, and processed prior to training.

2.5 RNNs and synthetic training data

Recurrent encoder-decoder neural networks (RNNs) can generate a target sequence given an input sequence. Sutskever et al. (2014) introduce this sequence-to-sequence architecture for machine translation. Kann and Schütze (2016) adapt RNNs to perform morphological reinflection by training the models on the character level.

RNNs are sensitive to the amount of training data. In our preliminary experiments, RNNs performed poorly in the low-resource setting. In order to increase the accuracy of the RNNs, we supplement the training data with morphological analyses generated by a DIRECTL+ model trained on the 100 training forms, and applied to randomly-chosen words from an unlabeled corpus using the method of Nicolai and Kondrak (2017). Many of these analyses are incorrect, but overall they provide information to the neural model that enforces inflectional patterns observed in the original training data. This process is shown schematically in Figure 2.

Because RNNs train with a stochastic learning algorithm, they are very dependent upon their initialization method (Goodfellow et al., 2016). In order to improve the stability of the RNNs, we ensemble five distinct models, each initialized with a different random seed. We produce an n-best list from each network, and combine them with equal weighting. This ensembling process is a common technique intended to stabilize neural networks, and lessen the impact of local optima. Our development experiments confirmed that ensembling can reduce the error rate over individual networks by more than 20%, while reducing the variance by half.

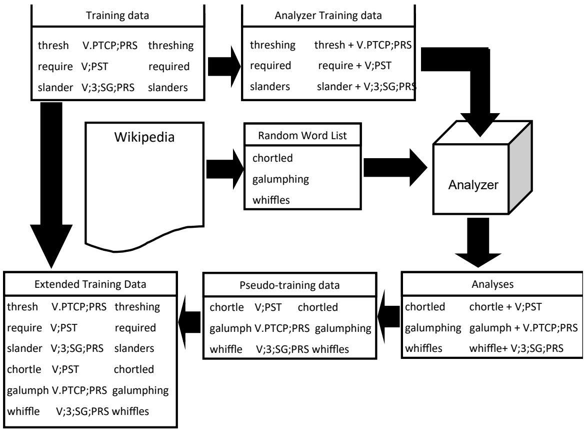
Training data
thresh V.PTCP;PRS threshing
require V;PST required
slander V;3;SG;PRS slanders
Analyzer Training data
threshing thresh + V.PTCP;PRS
required require + V;PST
slanders slander + V;3;SG;PRS
Wikipedia
Random Word List
chortled
galumphing
whiffles
Analyzer
Analyses
chortled chortle + V;PST
galumphing galumph + V.PTCP;PRS
whiffles whiffle+ V;3;SG;PRS
Pseudo-training data
chortle V;PST chortled
galumph V.PTCP;PRS galumphing
whiffle V;3;SG;PRS whiffles
Extended Training Data
thresh V.PTCP;PRS threshing
require V;PST required
slander V;3;SG;PRS slanders
chortle V;PST chortled
galumph V.PTCP;PRS galumphing
whiffle V;3;SG;PRS whiffles

Figure 2: Generation of synthetic training data for RNNs.

2.6 Language models

Transduction models trained on small amounts of data often produce output forms that violate the phonotactic constraints of a language. Character-level language models offer the possibility of reducing the number of implausible outputs. For each language, we produce a list of word types from the first million lines of the provided Wikipedia corpus, and create a 4-gram character language model using the CMU language modeling toolkit.[2] This language model, however, is very noisy, because the corpus contains many hyperlinks and filenames.

We attempt to improve the quality of the language models using the following two methods. The first method is to disregard the corpus, and instead produce a small language model derived exclusively from the target forms in the training data. The second method, which we refer to as affix-matching, is to use only those word types in the corpus that match the affixes seen in training. We identify the affixes by extracting any character sequence in the training set that is aligned to a subtag by M2M-aligner.

2.7 System combination

In an attempt to leverage their unique strengths, we combine DIRECTL+ with a neural network ensemble. Both approaches produce ranked n-best lists. In addition, we include the provided baseline system, which produces a single output form for each input instance. A diagram of our two system combination methods is shown in Figure 3.

[2] http://www.speech.cs.cmu.edu/SLM/toolkit.html

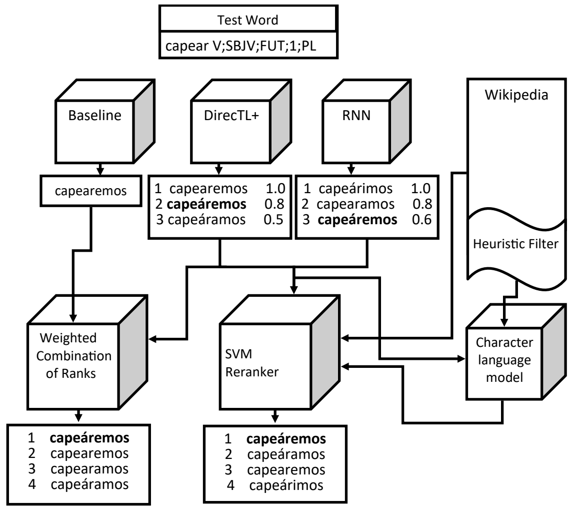

Figure 3: Two methods of system combination. Correct outputs are shown in bold.

The first method is a simple linear combination, which selects the prediction with the highest weighted average of the three ranks. Combining ranks, rather than numerical scores, circumvents issues with scaling, and allows the integration of the baseline, which produces no score.

The second combination method is the reranking of the n-best list produced by DIRECTL+ using other system outputs as features. By framing the reranking of an n-best list as a classification task (Joachims, 2002), we can also leverage other sources of information, such as the language model described in Section 2.6. Our SVM reranker includes four features: (1) the normalized score produced by DIRECTL+, (2) the normalized score produced by the RNN ensemble, (3) a binary indicator of the presence of a prediction in a corpus, and (4) the normalized probability assigned to the prediction by a character language model. The general objective is to promote high-scoring predictions shared by multiple systems that occur in the corpus or look like real words.

3 Experiments

We conduct experiments on five languages: English (EN), German (DE), Persian (FA), Polish (PL), and Spanish (ES). The training data in the low-resource setting of the inflection generation task is limited to 100 instances. The DIRECTL+ models are trained on the subtag sequences made consistent with the method described in Section 2.3. For two languages, we identified best subtag orderings that are different from the initial orderings; the Spanish ordering was found with the alignment-based method. while the Persian ordering was hand-crafted by a native speaker using linguistic analysis.

Our other systems take advantage of the first one million lines of the Wikipedia dumps from 2017/03/01 provided by the task organizers. Our RNN models are trained on the original training set augmented with 16,000 synthetic instances generated by the DIRECTL+ morphological analyzers, as described in Section 2.5. For the language models that inform our SVM reranker, we use the entire Persian corpus, training data only for English and Polish, and the affix-match method for German and Spanish (Section 2.6). The reranker is trained using 2-fold cross-validation on the training data.

3.1 Development results

Our development results are summarized in Table 1. We see that our DIRECTL+ models (DTL) substantially outperform the official baseline (BL). even without subtag reordering. The only exception is Persian, in which the best ordering strategy (BO) makes a dramatic difference. Further, modest gains are obtained via linear combination (LC) and reranking (RR) of the best individual systems.

	BL	RNN	DTL	BO	LC	RR
EN	76.2	76.3	88.0		88.0	88.0
DE	53.7	43.3	66.6		68.6	68.8
FA	27.3	8.1	23.9	40.8	41.4	40.7
PL	41.9	36.0	48.2		49.3	49.0
ES	58.6	38.9	65.8	68.3	68.3	68.4

Table 1: Results on the development sets.

The most striking outcome is the disappointing performance of the RNN ensembles, which in most cases is well below the baseline, even with the addition of the synthetic data.[3] In this context, it is not surprising that system combination only minimally improves over DIRECTL+ by itself.

Based on the development results, we decided to submit 3 versions for each of the 5 languages (DTL, LC, and RR) plus two runs that correspond to the best subtag ordering (BO) for Spanish and Persian.

[3] Without synthetic data, our RNN ensembles completely fail on this task in the low-resource setting.

3.2 Test results

Our results on the test set are shown in Table 2. The numerical tags of the submitted runs are shown in the top row. In the cases of incorrect files being mistakenly submitted, we provide the actual results, which may differ from the official ones. With the exception of Persian, our results are among the best in the low-resource setting.

			01	02	03	04
	BL	RNN	DTL	BO	LC	RR
EN	80.6	78.4	90.6		90.6	90.3
DE	55.3	57.1	66.0		66.8	66.2
FA	24.5	8.1	*19.5*	*38.3*	*39.0*	*37.7*
PL	42.3	28.2	45.2		45.3	45.9
ES	57.1	37.9	64.6	*68.2*	*68.0*	*67.3*

Table 2: Results on the test sets. Runs corrected after the submission deadline are in italics.

The system combination results largely confirm the development experiments. Notably, the simple linear combination, which has no access to language models, performs slightly better on average than the SVM reranker, and seems to be more stable as well. One possible explanation is the necessary subdivision of an already small training set in order to train the reranker, which further reduces the amount of the training data. The linear combination requires no training, but its weights are tuned on a relatively large development set.

3.3 Error analysis

English is characterized by a relatively simple inflectional morphology, with only 5 verbal inflection slots. Most words are regular, and pose no problem even to an inflection model trained on only 100 instances. The errors tend to reflect irregular verbs, as well as orthographic rules, such as the consonant doubling in *splitting*. The current RNN-based systems are unlikely to achieve significantly better results in the low-resource setting.

A number of German errors can be attributed to implicit information that can only be learned by observing multiple forms. For example, the genitive singular suffix differs depending on the gender of the noun. Certain suffixes, such as `-in`, often indicate the gender of a noun to be feminine. However, the only genitive feminine singular in the training data does not end in `-in`, and thus, our system fails to correctly predict the genitive singular of *Köchin*.

Persian results seem to be affected by subtag orderings to a greater degree than other languages. The verbal morphology demonstrates some agglutinative properties, where individual subtags may match their own affix. One of the authors handcrafted a subtag ordering, which turned out to be much more effective than the orderings derived by our algorithmic methods. The other sources of difficulty that set Persian apart are the differences between formal and colloquial inflectional forms, which are both represented in the training data, as well as the preponderance of multi-word inflection forms (86% of the test instances), which complicates the task of the language model.

Many Polish outputs are non-words, which we expected to be filtered out by the language model. In many cases, the reranker has no chance to succeed, as none of the models includes the correct form in its top-n list. In other cases, the signal from the language model is not strong enough to overrule the top DirecTL+ prediction.

An interesting type of error in Spanish are forms that involve orthographically illegal bigrams like `ze`. DirecTL+ has a set of bigram features on the target side, but their weights are established on the training set, which is too small to learn such constraints. In the future, we would like to investigate ways to integrate the unlabeled corpus information directly into the DirecTL+ generation process.

The languages that we consider in this paper are mostly fusional. Another avenue for future work is adapting our approach to other types of languages.

4 Conclusion

Kann and Schütze (2016) show that the neural network models achieve high accuracy on the morphological reinflection task, given a sufficiently large training set. However, the effectiveness of neural models in the low-resource setting is yet to be demonstrated. In this paper, we have described an attempt to combine our string transduction tool with a reimplementation of the neural approach, which turned out to be largely unsuccessful due to the weakness of the latter. Nevertheless, we are satisfied with several novel ideas that we have developed for the shared task, and with the entire learning experience for the members of our team. The overall results confirm the competitiveness of our string transduction approach in the low-resource setting.

Acknowledgments

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

References

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. The CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*. Association for Computational Linguistics, Vancouver, Canada.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 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. http://www.aclweb.org/anthology/W16-2002.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Bradley Hauer and Grzegorz Kondrak. 2016. Decoding anagrammed texts written in an unknown language and script. *Transactions of the Association of Computational Linguistics* 4:75–86. http://aclweb.org/anthology/Q16-1006.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*. pages 905–913. http://www.aclweb.org/anthology/P/P08/P08-1103.

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. http://www.aclweb.org/anthology/N/N07/N07-1047.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '02, pages 133–142. https://doi.org/10.1145/775047.775067.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. pages 62–70. http://www.aclweb.org/anthology/W16-2010.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, pages 91–98. http://aclweb.org/anthology/P05-1012.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. 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*. Association for Computational Linguistics, pages 922–931. http://aclweb.org/anthology/N15-1093.

Garrett Nicolai, Bradley Hauer, Adam St Arnaud, and Grzegorz Kondrak. 2016. Morphological reinflection via discriminative string transduction. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. pages 31–35. http://www.aclweb.org/anthology/W16-2005.

Garrett Nicolai and Grzegorz Kondrak. 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*. Association for Computational Linguistics, Valencia, Spain, pages 211–216. http://www.aclweb.org/anthology/E17-2034.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.