Combining Neural and Non-Neural Methods for Low-Resource Morphological Reinflection

Saeed Najafi, Bradley Hauer, Rashed Rubby Riyadh, Leyuan Yu, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{snajafi,bmhauer,riyadh,leyuan,gkondrak}@ualberta.ca

Abstract

We describe our systems and results in the type-level low-resource setting of the CoNLL–SIGMORPHON 2018 Shared Task on Universal Morphological Reinflection. We test non-neural transduction models, as well as more recent neural methods. We also investigate the effect of leveraging unannotated corpora to improve the performance of selected methods. Our best system obtains the highest accuracy on 34 out of 103 languages.

1 Introduction

In this system paper, we discuss our submissions to the CoNLL–SIGMORPHON 2018 Shared Task on Universal Morphological Reinflection (Cotterell et al., 2018). We focus on the sub-task of type-level inflection under the low-resource scenario, in which the training data is limited to 100 labelled examples. Because of the sheer number of tested languages, we attempted no language-specific modifications. The results demonstrate that our non-neural transduction models perform better on average than our neural models. However, combining neural and non-neural models yields the highest accuracy.

In addition to standard submissions, we test novel methods of leveraging additional monolingual corpora, from which we derive target language models and/or word lists. We show that substantial gains in accuracy can be obtained in the way. Again, a combination of neural and non-neural systems produces the best non-standard results.

The paper has the following structure. In Section 2, we describe four standard systems, as well as our weighted-voting method of combining them. Our two non-standard systems and their linear combination are introduced in Section 3. Section 4 discusses the results.

2 Standard Systems

In this section, we briefly describe the four individual standard systems that we experimented with, followed by our voting method for combining them.

2.1 BASELINE (UA-01)

The shared task organizers have provided a baseline system for the type-level sub-task.[1] For each training instance, the baseline system aligns the input and output forms, and uses leading and trailing null alignments to identify prefix and suffix boundaries. Thus, the input and output are each divided into a prefix, stem, and suffix, with the prefix and suffix possibly being empty. The pairs of aligned characters from the suffix, and optionally a trailing substring of the stem, are recorded as suffixing rules for the morphological tag of the instance in question. Prefixing rules are identified in an analogous way. In this way, a series of inflection rules are generated from aligned training pairs for each morphological tag attested in the training data.

To perform reinflection on an unseen instance, the longest applicable suffixing rule for the given tag is selected and applied, as is the most frequent prefixing rule. Since some languages tend to prefer prefixing over suffixing, a heuristic is used to detect which of the two types of affixation is predominant. If a preference for prefixing is detected, all input and output strings for that language are reversed. During development, we found that the output files produced by the BASELINE system for these languages had the lemmas reversed; we rectified this issue for our experiments and submissions.

[1] https://github.com/sigmorphon/conll2018/tree/master/task1/baseline

Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, pages 116–120,
Brussels, Belgium, October 31, 2018.

2.2 HAEM (UA-02)

The hard attention model over edit actions (HAEM) of Makarov et al. (2017) performed very well in the low-resource setting of the 2017 edition of the shared task. We use the implementation made available by the authors.[2] The method learns a neural state-transition model with hard monotonic attention. It produces sequences of insertion and deletion operations on the lemma that transduce it into the appropriate inflected form. The system that achieved the top results in the 2017 shared task was an ensemble of up to 15 different models, each trained with multiple seeds. Because of time constraints, and the difficulties with using the implementation, we derive only a single transition inflector model for each language, eschewing the complex ensemble procedures described in the original paper.

The process of compiling and running the provided code was non-trivial. In particular, libraries required by the provided code had been supplanted by newer versions, which lacked backwards compatibility. Since the versions used in 2017 are no longer readily available, we had to adapt the code to the new versions. Further modifications were necessary to account for the different format of the test data this year. Even with these modifications, the code failed to run properly on several languages, resulting in 0% accuracy.

2.3 DIRECTL+ (UA-03)

We perform string transduction with a modified version[3] of DIRECTL+, a tool originally designed for grapheme-to-phoneme conversion (Jiampojamarn et al., 2008). 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. We perform source-target pair alignment with a modified version[4] of the M2M aligner (Jiampojamarn et al., 2007), which applies the EM to maximize the conditional likelihood of its aligned source and target pairs.

M2M-aligner		
source side	target side	maximum tag
{1-2}	{1-2}	{2-4}
DIRECTL+		
n-gram	context size	joint m-gram
{1-5}	{3-11}	{1-10}

Table 1: The tuning ranges for hyper-parameters.

We apply the tag splitting and particle handling techniques described in our 2017 system paper Nicolai et al. (2017). In particular, we split the tags into subtags, and append them at both the beginning and end of the lemma. We decided not apply any subtag reordering techniques this year, due to the large number of languages.

We tune hyper-parameters for each language using grid search. Table 1 specifies the tuning ranges for both the aligner and the transducer. The list of the actual hyper-parameter settings for each language is available on request.

2.4 AC-RNN (UA-04)

AC-RNN is our novel implementation of the encoder-decoder RNN model, which is specialized to the sequence-labelling task, and trains with an Actor-Critic reinforcement-learning objective (Najafi et al., 2018a). The implementation is further modified to incorporate soft-general attention mechanism, and adapted to the task of morphological reinflection.[5] In an initial experiment, we validated AC-RNN using the high-resource French dataset from the 2017 shared task, obtaining the test accuracy of 89.7%, compared to 89.5% of the best-performing 2017 ensemble system.

2.5 Standard Combination (UA-05)

In our development experiments, we observed that no system described in this section strictly dominates the others in terms of accuracy on every language; rather, different systems perform well on different languages. Furthermore, we often found instances where incorrect predictions were made by the top-performing system for the language in question, but the correct output was produced by other systems. These observations motivated our attempt to combine the strengths of the four systems.

[2] https://gitlab.cl.uzh.ch/makarov/sigmorphon2017
[3] https://github.com/GarrettNicolai/DTL
[4] https://github.com/GarrettNicolai/m2m
[5] https://gitlab.com/SaeedNajafi/ac-morph

Our standard combination approach is based on *weighted voting*. The top prediction from each of the four individual systems[6] is assigned a score equal to the system's accuracy on the development set for that language. The prediction with the highest total score is returned.

This system favors predictions from the top-performing system on a given language, while allowing errors to be corrected when other systems agree on a different prediction. If one system achieves an accuracy greater than the sum of the accuracies of all other systems, it dominates the voting, and the output of the combination is identical to the output of that system. This scenario occurred for only seven languages.

3 Non-standard Systems

Large monolingual raw text corpora, which are freely available for a wide variety of languages, offer the possibility of improving the accuracy of transduction models trained on small amounts of source-target pairs. Many of the target forms are observed in raw text corpora. In addition, character-level language models derived from monolingual corpora can reduce the number of output forms that violate the phonotactic constraints of a language. Target language modelling is particularly important in low-data scenarios, where the limited transduction models often produce many ill-formed output candidates. In this section, we describe the sources of the text corpora, and two novel methods that attempt to leverage the additional information.

3.1 Additional Data

The monolingual corpora come from one of two sources. The UniMorph project (Kirov et al., 2018) contains corpora for 46 out of 103 languages.[7] For 42 languages that are not represented in Unimorph, we instead use the target side of the high-resource training data in this shared task. For the 15 remaining languages that lack either of these resources, we simply back off to the standard version of each system. Note that we use only the target-side forms of the high-resource training data (if applicable), so that there is no overlap between the training and testing sets.

The principal use of the additional data is to construct a list of all word types, with counts, into a *target word list*. The idea is to bias the system predictions towards forms that are actually observed in a monolingual corpus. In this shared task, our word list sizes vary between 115 for Arabic and 22,371 for Slovene.

The second use of the unannotated corpora is to derive a target character-level n-gram language model. For this purpose, we employ the CMU language modeling toolkit.[8]

3.2 DTLM (UA-06)

Nicolai et al. (2018) present DTLM, a new system that combines discriminative transduction with character and word language models derived from large unannotated corpora. DTLM is an extension of DIRECTL+ (Section 2.3), whose target language modeling is limited to a set of binary n-gram features, which are based exclusively on the target sequences from the parallel training data. DTLM avoids the error propagation problem that is inherent in pipeline approaches by incorporating the language-model features directly into the transducer.

In addition, DTLM bolsters the quality of transduction by employing a novel alignment method, which is referred to as precision alignment. The idea is to allow null substrings on the source side during the alignment of the training data, and then apply a separate aggregation algorithm to merge them with adjoining non-empty substrings. This alignment method results in substantially higher transduction accuracy.[9]

3.3 AC-RNN with Word Lists (UA-07)

We also indirectly leverage the target word lists (ignoring the counts) in the AC-RNN model (Section 2.4). The neural network is trained with each of these external words as both input and output. We pre-train AC-RNN with this copying procedure for 50 epochs. (Bergmanis et al. (2017) use a similar technique with randomly-generated sequences.) We then fine-tune the model on the actual low-resource dataset. This approach is helpful in a several different ways: it biases the network towards copying input characters in the output, guides the attention parameters towards learning a monotonic alignment, and improves the randomly

[6] We had no access to additional top-n predictions from the BASELINE and HAEM systems.

[7] https://unimorph.github.io

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

[9] DTLM was also succesfully used in the NEWS 2018 shared task on transliteration (Najafi et al., 2018b).

Method	Dev	Test
Standard		
BASELINE	39.3	38.2
HAEM	40.5	39.2
DIRECTL+	47.2	44.8
AC-RNN	21.4	21.3
Combination	**52.5**	**50.5**
Non-Standard		
AC-RNN + WL	38.7	38.0
DTLM	51.4	49.7
Combination	**54.4**	**53.2**

Table 2: The average accuracy across all languages.

initialized character embeddings by pre-training them on external data.

We also experimented with two different ideas to re-rank predictions of AC-RNN. The first idea was to train a separate RNN-based language model to re-score predictions. The second idea was to learn a reverse model that would generate the input lemma from the inflected form and tag, for the purpose of re-scoring the n-best lists of AC-RNN. Unfortunately, neither of these approaches outperformed the copying procedure outlined in the previous paragraph.

3.4 Non-standard Combination (UA-08)

We take advantage of the ability of both DTLM and AC-RNN to produce n-best lists of predictions by combining the lists via a linear combination of their confidence scores. The scores from each model are normalized, and the linear coefficients are tuned separately for each language on the provided development sets. The top scoring output for each input instance is returned.

4 Results

Table 2 shows the average accuracy over 103 languages for our eight submitted systems in the low-resource setting. The ranking of the systems is the same for both the development and test sets.[10] The best performing individual standard system is DIRECTL+, followed by HAEM, BASELINE, and AC-RNN. We conclude that 100 training instances are insufficient for the soft-attention based neural models like AC-RNN. Moreover, we were not able to replicate the superior results of HAEM reported in the 2017 shared task, which we attribute to the reasons outlined in Section 2.2. Our weighted-voting combination of all four systems substantially improves over each individual system. In the development experiments, we observed that all individual systems, including AC-RNN, contributed to the accuracy of the combination system.

[10] Detailed results on all languages are available on request.

	None	+LM	+LM +WL
High-Resource	38.6	42.6	*28.6*
Unimorph	38.6	45.4	49.7

Table 3: The average accuracy of DTLM on the development sets of 46 languages with additional data.

Among the non-standard systems, the DTLM model easily outperforms DIRECTL+. The copy pre-training approach on the target word lists almost doubles the accuracy of AC-RNN, but it is not sufficient to even reach the BASELINE. Nevertheless, the linear combination of the two non-standard systems is clearly the best of our submissions, obtaining the highest accuracy on 34 languages in the shared task.

In order to shed light on the effect of additional data on the DTLM results, we ran experiments on 46 languages that have both the Unimorph and high-resource data (Table 3). It is clear that incorporating a target language model from either data source improves the overall accuracy. The results also suggest that the Unimorph corpora are better for the purpose of deriving the language models than the high-resource training data. The addition of the target word lists from Unimorph further improves the results. However, the word lists from the high-resource data are detrimental. Since there is no overlap between the training and development data, there are no useful targets in the word lists to help guide the model outputs.

5 Conclusion

We described the details of the systems that we tested on 103 languages in the low-resource setting of the shared task. In particular, we experimented with combining diverse systems, applying reinforcement learning to neural models, and leveraging target corpora for reinflection. Our results suggest that these techniques lead to improvements in accuracy with respect to the base systems. We hope that this report will serve as a useful reference for future experiments involving the datasets from this shared task.

Acknowledgements

We thank Garrett Nicolai for the assistance with DTLM. We thank the shared task organizers for preparing a large number of datasets, and smoothly executing the evaluation process. This research was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and Alberta Advanced Education.

References

Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, Brussels, Belgium. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. *Proceedings of ACL-08: HLT*, pages 905–913.

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.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Graldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian Mielke, Arya D. McCarthy, Sandra Kbler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: Uzh at sigmorphon 2017 shared task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57. Association for Computational Linguistics.

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)*, pages 91–98. Association for Computational Linguistics.

Saeed Najafi, Colin Cherry, and Grzegorz Kondrak. 2018a. Sequence labeling and transduction with output-adjusted actor-critic training of RNNs. In preparation.

Saeed Najafi, Bradley Hauer, Rashed Rubby Riyadh, Leyuan Yu, and Grzegorz Kondrak. 2018b. Comparison of assorted models for transliteration. In *Proceedings of the Seventh Named Entities Workshop*, pages 84–88. Association for Computational Linguistics.

Garrett Nicolai, Bradley Hauer, Mohammad Motallebi, Saeed Najafi, and Grzegorz Kondrak. 2017. If you can't beat them, join them: the university of alberta system description. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 79–84.

Garrett Nicolai, Saeed Najafi, and Grzegorz Kondrak. 2018. String transduction with target language models and insertion handling. In *Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Brussels, Belgium.