Improving Low-Resource Morphological Learning with Intermediate Forms from Finite State Transducers

Sarah Moeller and **Ghazaleh Kazeminejad** and **Andrew Cowell** and **Mans Hulden**
Department of Linguistics
University of Colorado
first.last@colorado.edu

Abstract

Neural encoder-decoder models are usually applied to morphology learning as an end-to-end process without considering the underlying phonological representations that linguists posit as abstract forms before morphophonological rules are applied. Finite State Transducers for morphology, on the other hand, are developed to contain these underlying forms as an intermediate representation. This paper shows that training a bidirectional two-step encoder-decoder model of Arapaho verbs to learn two separate mappings between tags and abstract morphemes and morphemes and surface allomorphs improves results when training data is limited to 10,000 to 30,000 examples of inflected word forms.

1 Introduction

A morphological analyzer is a prerequisite for many NLP tasks. A successful morphological analyzer supports applications such as speech recognition and machine translation that could provide speakers of low-resource languages access to online dictionaries or tools similar to Siri or Google Translate and might support and accelerate language revitalization efforts. This is even more crucial for morphologically complex languages such as Arapaho, an Algonquian language indigenous to the western USA. In polysynthetic languages such as Arapaho, inflected verbal forms are often semantically equivalent to whole sentences in morphologically simpler languages. A standard linguistic model of morphophonology holds that multiple morphemes are concatenated together and then phonological rules are applied to produce the inflected forms. The operation of phonological rules can reshape the string of fixed morphemes considerably, making it difficult for learners, whether human or machines, to recreate correct forms (generation) from the morpheme sequence or to analyze the reshaped inflected forms into their individual morphemes (parsing).

In this paper we describe an experiment in training a neural encoder-decoder model to replicate the bidirectional behavior of an existing finite state morphological analyzer for the Arapaho verb (Kazeminejad et al., 2017). When a language is low-resource, natural language processing needs strategies that achieve usable results with less data. We attempt to replicate a low-resource context by using a limited number of training examples. We evaluate the feasibility of learning abstract intermediate forms to achieve better results on various training set sizes. While common wisdom regarding neural models has it that, given enough data (Graves and Jaitly, 2014), end-to-end training is usually preferable to pipelined models, an argument can be made that morphology is an exception to this: learning two regular mappings separately *may* be easier than learning a single complex one. In Liu et al. (2018), adressing a related task, noticeably better results were reached for German, Finnish, and Russian when a neural system was first tasked to learn morphosyntactic tags than when it was tasked to produce an inflected form directly from uninflected forms and context. These three languages are morphologically complex or unpredictable, but marginally better results were achieved for the less complex languages.

2 Arapaho Verbs

Arapaho is a member of the Algonquian (and larger Algic) language family; it is an agglutinating, polysynthetic language, with free word order (Cowell and Moss Sr, 2008). The language has a very complex verbal inflection system, with a number of typologically uncommon elements. A given verb stem is used either with animate or inanimate subjects for intransi-

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 1 Papers, pages 81–86,
Honolulu, Hawai‘i, USA, February 26–27, 2019.

tive verbs (*tei'eihi-* 'be strong.animate' vs. *tei'oo-* 'be strong.inanimate'), and with animate or inanimate objects for transitive verbs (*noohow-* 'see s.o.' vs. *noohoot-* 'see s.t.'). For each of these categories, the pronominal affixes/inflections vary in form. For example, 2SG with intransitive, animate subject is /-*n*/, while for transitive, inanimate object it is /-*ow*/ (*nih-tei'eihi-n* 'you were strong' vs. *nih-noohoot-ow* 'you saw it').

All stem types can occur in four different verbal orders, whose function is primarily modal. These verbal orders each use different pronominal affixes/inflections as well. Thus, with four different verb stem types and four different verbal orders, there are a total of 16 different potential inflectional paradigms for any verbal root, though there is some overlap in the paradigms, and not all stem forms are possible for all roots.

Arapaho also has a proximate/obviative system, which designates pragmatically more- and less-prominent participants. "Direction-of-action" markers included in inflections do not correspond to true pronominal affixes. Thus *nih-noohow-oot* 'more important 3SG saw less important 3S/PL' vs. *nih-noohob-eit* 'less important 3SG/PL saw more important 3S'. The elements *-oo-* and *-ei-* specify direction of action, not specific persons or numbers of participants.

Arapaho has both progressive and regressive vowel harmony, operating on /*i*/ and /*e*/ respectively. This results in alternations in both the inflections themselves, and the final elements of stems, such as *noohow-un* 'see him!' vs. *niiteheib-in* 'help him!', or *nih-ni'eeneb-e3en* 'I liked you' vs. *nih-ni'eenow-oot* 'he liked her'.

The Arapaho verb, then, is subject to complicated morphophonological processes. For example, the underlying form of the word 'we see you' concatenates the transitive verb stem with animate object (TA) *noohow* 'see' and the '1PL.EXCL.SUBJ.2SG.OBJ' suffix *-een*. This underlying form undergoes significant transformation after morphophonological rewrite rules are applied. An initial change (IC) epenthesizes *-en* before the first vowel in the verb stem because it is a long vowel and because the verb is affirmative present. Then vowel harmony is at work, changing *n-en-oohow-een* to *n-on-oohow-een*. Finally a consonant mutation rule changes *w* to *b*, producing the surface form *nonoohobeen* (cf. Figure 1).

3 Finite State Model

One of the clear successes in computational modeling of linguistic patterns has been finite state transducer (FST) models for morphological analysis and generation (Koskenniemi, 1983; Beesley and Karttunen, 2003; Hulden, 2009; Lindén et al., 2009). An FST is bidirectional, able to both parse inflected word forms and generate all possible word forms for a given stem (Beesley and Karttunen, 2003). Given enough linguistic expertise and time investment, FSTs provide the capability to analyze any well-formed word in a language.

The Arapaho FST model used in this paper was constructed with the *foma* finite-state toolkit (Hulden, 2009). It used 18,559 verb stems taken from around 91,000 lines of natural discourse in a large transcribed and annotated spoken corpus of Arapaho, parts of which are publicly available in the Endangered Languages Archive (ELAR).[1]. All possible basic inflections occur in the corpus. The FST produces over 450,000 inflected forms from the stems.

The FST is constructed in two parts, the first being a specification of the lexicon and morphotactics using the finite-state lexicon compiler (lexc), a high-level declarative language for effective lexicon creation, where concatenative morphological rules and morphological irregularities are addressed (Karttunen, 1993). The first part produces intermediate, abstract "underlying" forms. These forms concatenate the appropriate morphemes from the lexicon in the correct order, (e.g. *noohoween* in Figure 1) but are not well-formed words in the language.

The second part of the FST implements the morphophonological and phonological rules of the language using "rewrite rules". These rules apply the appropriate phonological changes to the intermediate forms in specified contexts. Thus, in generation, the inflected word is not merely a bundle of morphemes, but the completely correct word form in accord with the morphophonological and phonological rules of the language. By composing, in a particular order (specified in the grammar of the language), the FSTs resulting from these rewrite rules to the parsed forms, the result is a single FST able to both generate and parse as shown in Figure 1.

[1] https://elar.soas.ac.uk/Collection/MPI189644

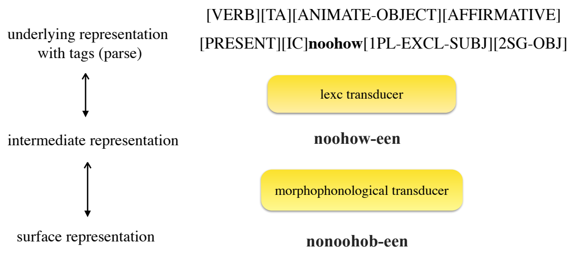
underlying representation with tags (parse)
[VERB][TA][ANIMATE-OBJECT][AFFIRMATIVE]
[PRESENT][IC]noohow[1PL-EXCL-SUBJ][2SG-OBJ]
lexc transducer
intermediate representation
noohow-een
morphophonological transducer
surface representation
nonoohob-een

Figure 1: An example of a parsed form with verb stem and morphosyntactic tags (top left) and inflected surface form (bottom left) for the Arapaho FST. The intermediate underlying phonological forms (middle left) are accessible to the FST before/after applying morphophonological alternations.

4 Training the LSTM

Although an extensive finite state morphological analyzer is an extremely useful resource, neural models are much better able to analyze to unseen forms than finite state machines are. However, neural models are hampered in low-resource contexts by their data greediness. In order to see whether this limitation could be addressed we simulated training a neural model in low-resource contexts using output from the Arapaho FST. Since the currently strongest performing models for morphological inflection (Cotterell et al., 2017; Kann and Schütze, 2016; Makarov et al., 2017) use an LSTM-based sequence-to-sequence (seq2seq) model (Sutskever et al., 2014), we follow this design in our work. We implement the seq2seq model with OpenNMT's (Klein et al., 2018) default parameters of 2 layers for both the encoder and decoder, a hidden size of 500 for the recurrent unit, and a maximum batch size of 64.

Training corpora of various sizes are created by randomly selecting examples of inflected word forms and their corresponding intermediate and parsed forms from the bidirectional output of the Arapaho FST. This results in triplets like in Figure 1. The triplets are arranged into three pairs—inflected "surface" forms (SF) & intermediate forms (IF), IF & parsed forms (PF), and SF & PF. Re-using the pairs for both parsing and generation gives six data sets. For simplicity's sake, since the primary aim is to compare the two strategies' performance and not to measure accuracy, forms with ambiguous inflected forms, parses, or intermediate forms were filtered. Other experiments (Moeller et al., 2018) indicate that pre-processing the data to account for ambiguous forms would not greatly affect accuracy.

We treat the intermediate strategy of parsing as a translation task of input character sequences from the fully-inflected surface forms to an output of character sequences of the intermediate forms, and then from the intermediate forms to a sequence of morphosyntactic tags plus the character sequences of the verbal root. Generation follows the same model in the opposite direction.

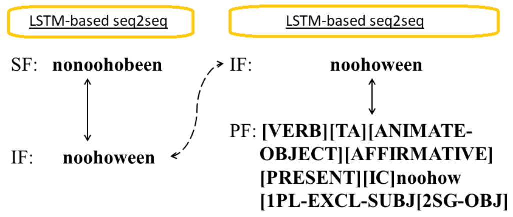
LSTM-based seq2seq
LSTM-based seq2seq
SF: nonoohobeen
IF: noohoween
IF: noohoween
PF: [VERB][TA][ANIMATE-OBJECT][AFFIRMATIVE][PRESENT][IC]noohow[1PL-EXCL-SUBJ[2SG-OBJ]

Figure 2: An example from training/test sets. In parsing, surface forms (SF) predict intermediate forms (IF). The output trains another encoder-decoder to predict parsed forms (PF). Generation follows the same steps but proceeding from the PF instead.

The selected data is divided roughly in half. The first half serves as training and development and the second half as testing data in the first step of the intermediate training strategy (SF⇔IF or PF⇔IF). In order to compare the two training strategies, the output of this intermediate step trains and tests the second step of the intermediate strategy. The original second half also serves to train and test the direct strategy (SF-PF or PF-SF). Symbol prediction degrades at the end of a sequence. Best results are achieved when each character/tag sequence is doubled on its line for training and testing (Moeller et al., 2018) and trimmed for evaluation. So, for example, *nonoohobeen* becomes *nonoohobeennonoohobeen* during training and testing but predicted symbols that exceed the length of the original string are deleted and the first half of the predicted string is evaluated against the original string.

5 Experiment and Results

We compare two strategies to train a neural model to generate inflected verbs from morphosyntactic tags with verb stem or to parse inflected verb forms. First, we train the neural model to learn correct output forms directly from the parsed or inflected input. Second, we added an intermediate step where the model first learns the mapping to intermediate forms and, from there, the mapping to the correct parsed or inflected form. We measure the final accuracy score and the average Levenshtein distance and compare the performance of the two strategies in generation and in parsing. Accuracy is measured as the fraction of correct generated/parsed forms in the output compared to complete gold inflected or parsed forms.

5.1 Generation

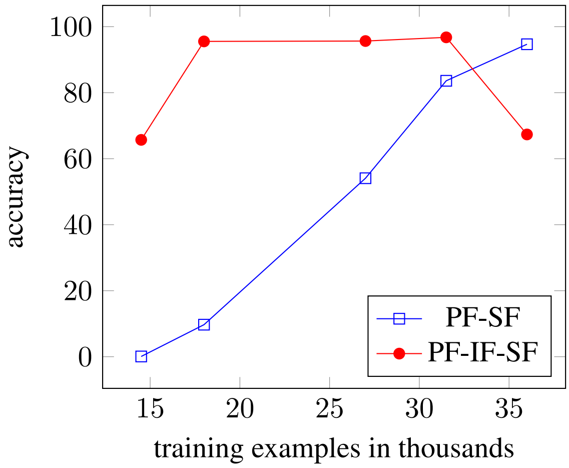

Figure 3: Generation - accuracy scores

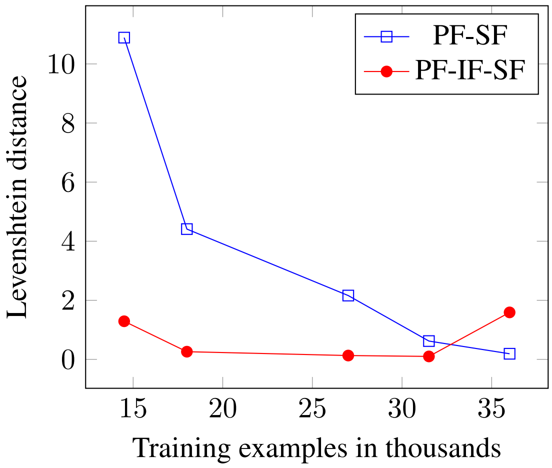

Figure 4: Generation - average Levenshtein distances

We trained a bidirectional LSTM encoder-decoder with attention (Bahdanau et al., 2015) to generate Arapaho verbs using five training sets with approximately 14.5, 18, 27, 31.5, and 36 thousand examples. The direct strategy trains on the morphosyntactic tags and verb stem. Each tag occurs in the same order as its corresponding morpheme appears in the intermediate form. Only "direction-of-action" tags/morphemes come after the stem.

The accuracy scores in Figure 3 and the Levenshtein distance measures in 4 show that the intermediate strategy performs better than the direct strategy in low-resource settings. Starting at about 14,500 training examples, where the direct strategy produces barely any inflected forms (SF) correctly, the intermediate strategy achieves nearly 69% accuracy. As the training size approaches 36,000, the advantage of the intermediate step is lost. Indeed, the intermediate strategy begins to perform worse while the direct strategy continues to improve. The intermediate strategy seems to peak at 30,000 training examples.

5.2 Parsing

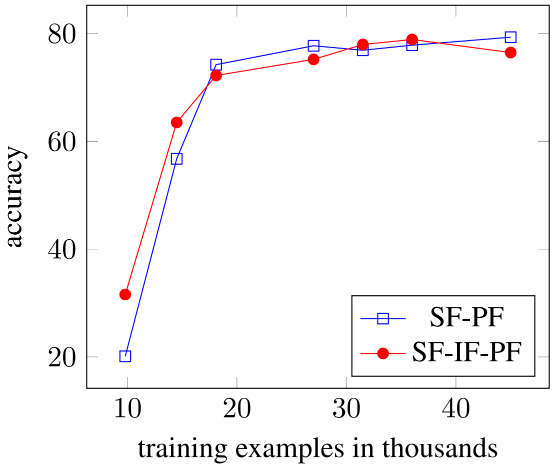

Figure 5: Parsing - accuracy scores

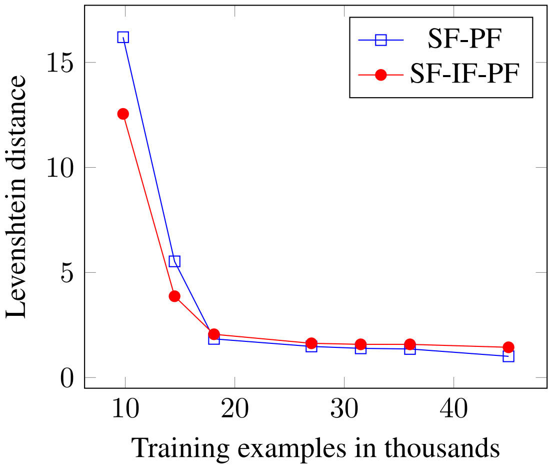

Figure 6: Parsing - average Levenshtein distances

The parsing trend is less clear when compared to morphological generation. We compare seven training sets of approximately 10, 14.5, 18, 31.5, 36, and 45 thousand examples. As in morphological generation, at the lowest data settings the intermediate learning strategy is preferable to the direct strategy, though it has a less dramatic performance difference. The accuracy scores in Figure 5 show that with 14,000 training examples, the intermediate strategy performs only about 10 points higher. The advantage of the intermediate strategy is less noticeable in parsing, nor does its advantage decrease as quickly. With 36,000 examples barely one point separates the two strategies and the intermediate strategy performs slightly better. The intermediate strategy performance does not begin to reduce until 45,000 examples. The average Levenshtein distances in Figure 6, however, show that the direct strategy improves more consistently, though it is still only slightly better as training size increases.

5.3 Discussion

An end-to-end neural model demands quite a bit of data in order to learn patterns. It appears that, for languages with complicated morphophonological alternations, if an intermediate model is trained on a simple concatenation of morphemes, these disadvantages may be counterbalanced. The morpheme substrings in the intermediate forms correspond predictably to morphosyntactic tags in the parsed form. Subsequent alternations are less predictable. This may explain the intermediate strategy's difference in performance between parsing and generation. A pipelined approach with intermediate training is generally not preferable to end-to-end training. The intermediate step inevitably introduces errors into the training of the second neural model. The intermediate strategy's performance degradation beyond 35 or 40 thousand examples might indicate that the errors become too dominant.

Comparing our results to the recent CoNLL-SIGMORPHON shared tasks (Cotterell et al., 2016, 2017, 2018), it is surprising that the Arapaho direct generation results at 10,000 examples are so low. However, polysynthetic languages are rare in the shared task—only one, Navajo, was available in 2016 and 2017–making it difficult to compare results on such complicated and varied morphologies. In addition, our data included phenomena which could be considered derivational, such as verbal stems signaling animacy and modality (cf. Sect. 2). Also, since the data was selected randomly from the full FST output, the neural model may simply have not seen enough repeated stems in the low settings. Our results are not very good at the lowest settings but, in future, more in-depth pre-processing and filtering of the data could improve overall performance.

The varying results from morphological parsing shown in Figures 5 and 6 demonstrate the preliminary nature of this study. The trend between the two strategies seems indicative but several more comparisons should be conducted on similar languages. We hope to conduct a similar study on other low-resource languages for which an FST exists in order to determine whether the trend will reappear.

6 Conclusion

A sweet spot exists between 10,000 and 30,000 randomly selected training examples of Arapaho verbs where better results are achieved in morphological generation by first training an encoder-decoder to produce the intermediate forms from an FST than by learning the inflected or parsed form directly. For generation, the intermediate strategy achieves the strongest results around 30,000 examples. The results of morphological parsing vary, with the intermediate strategy outperforming the direct strategy at very low settings but achieving similar results with 18,000 and 36,000 training examples. Overall, the intermediate strategy appears to produce reliably better results at low-resource settings than the direct strategy.

Acknowledgements

NVIDIA Corp. donated the Titan Xp GPU used in this research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.

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*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Gèraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30. Association for Computational Linguistics.

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, Berlin, Germany. Association for Computational Linguistics.

Andrew Cowell and Alonzo Moss Sr. 2008. *The Arapaho Language*. University Press of Colorado.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Lauri Karttunen. 1993. *Finite-state lexicon compiler*. Xerox Corporation. Palo Alto Research Center.

Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2017. Creating lexical resources for polysynthetic languages—the case of Arapaho. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 10–18, Honolulu. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184. Association for Machine Translation in the Americas.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and production*. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. 2009. HFST tools for morphology—an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47. Springer.

Ling Liu, Ilamvazhuthy Subbiah, Adam Wiemerslage, Jonathan Lilley, and Sarah Moeller. 2018. Morphological reinflection in context: CU Boulder's submission to CoNLL-SIGMORPHON 2018 shared task. In *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 86–92, Brussels. Association for Computational Linguistics.

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, Vancouver. Association for Computational Linguistics.

Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2018. A neural morphological analyzer for Arapaho verbs learned from a finite state transducer. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 12–20. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.