Phonological Features for Morphological Inflection

Adam Wiemerslage and **Miikka Silfverberg** and **Mans Hulden**
Department of Linguistics
University of Colorado
University of Helsinki
first.last@colorado.edu

Abstract

Modeling morphological inflection is an important task in Natural Language Processing. In contrast to earlier work that has largely used orthographic representations, we experiment with this task in a phonetic character space, representing inputs as either IPA segments or bundles of phonological distinctive features. We show that both of these inputs, somewhat counterintuitively, achieve similar accuracies on morphological inflection, slightly lower than orthographic models. We conclude that providing detailed phonological representations is largely redundant when compared to IPA segments, and that articulatory distinctions relevant for word inflection are already latently present in the distributional properties of many graphemic writing systems.

1 Introduction

Models of morphology are important to many tasks in Natural Language Processing, but also present new challenges of their own. Morphologically complex languages require analysis that is often only captured at the morpheme level, but is essential for syntactic or semantic representations. This requires effective morphological analysis, which often receives less attention than other subfields of Natural Language Processing. One relevant task in morphology is that of morphological inflection: automatically generating the inflected form of a lemma according to a given morphological specification. An example of this in English is $''\mathsf{walk}'' + 3 + \mathsf{SG} + \mathsf{PRES} \rightarrow ''\mathsf{walks}''$. There has been recent success in adopting the encoder-decoder architecture (Kann and Schütze, 2016), which has been effective in machine translation (Cho et al., 2014), to this task.

In this work, we explore representing the inputs to such an encoder-decoder model for morphological inflection in two additional ways: IPA segments and bundles of phonological distinctive features. Representing the inputs to an inflection model in phonetic space can unify the character inventory between languages with separate orthographies. The shared character inventory could also enable transfer learning in some instances where it otherwise would be impossible. There are also confusing idiosyncrasies in some orthographies that are not necessarily present in an IPA representation. For example, there are many instances of gemination in English that do not occur in the phonetic realizations of such words, as in ⟨control⟩ → ⟨controlled⟩. English also exhibits several examples of the same sound expressed by completely different orthographic realizations as in ⟨fly⟩ → ⟨flies⟩, or conversely ⟨arch⟩ (/tʃ/) ∼ ⟨monarch⟩ (/k/). Furthermore, a phonetic representation serves as an interface to an even richer representation of characters: phonological distinctive features.

We explore this by representing each IPA segment in a sequence as the combination of its distinctive features. This is potentially useful because (1) a model can learn representations for a fixed set of distinctive features, rather than for each unique IPA segment, and (2) the differences between similar phonemes should be more readily apparent in the distinctive feature representations than the IPA representations. When tasked with generating the past tense of the English verb ”stop”, transcribed as /stɑp/, a model may need to distinguish between both /t/ and /d/ as past tense suffixes, having seen such examples as ”kick”: /kɪk/ → /kɪkt/, or ”rig”: /ɹɪg/ → /ɹɪgd/. Rather than the model needing to learn good representations for both /p/ and /k/ as unrelated segments that precede a /t/ in the past tense, a phonological distinctive feature representation would explicitly capture that they share the feature [−voice]. This encourages a model to more quickly find the parameters that correctly gener-

Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 161–166
Brussels, Belgium, October 31, 2018.
https://doi.org/10.18653/v1/P17

ate this voicing assimilation, and produce the form /stɑpt/. That is, the model that learns from phonological features should quickly be able to generalize that this English past tense is realized as /t/ before voiceless segments. Similarly, in the example of "rob": /ɹɑb/ → /ɹɑbd/, the generated /d/ can be conditioned on [+voice] rather than the individual segment /b/.

An alternative hypothesis is that the proposed distinctive feature representation may, however, not have such a profound effect on the inflection model. This is because distributional representations of IPA segments or phonemic graphemes have been shown to capture good approximations of the distinctive feature space (Silfverberg et al., 2018). In order to test these two hypotheses, we experiment on a subset of data provided by task 1 of the CoNLL-SIGMORPHON 2017 Shared Task on Universal Morphological Reinflection (Cotterell et al., 2017), which introduced 42 more languages than the year before (Cotterell et al., 2016) for a total of 52 languages. We use an existing tool to perform G2P on the data, and, as a second step, to produce distinctive feature vectors from the resulting IPA segments. We evaluate the resulting models on their ability to generate IPA segments.

Related Work Phonetic distributional vectors have been explored for their effectiveness in several NLP applications; especially for informing scenarios that utilize borrowing or transfer learning (Tsvetkov et al., 2016). Phonological distinctive features have also been successfully used to inform NER (Bharadwaj et al., 2016). However, to our knowledge, there does not seem to be work in learning distributional properties of phonological features that compares them directly to vectors of IPA segments.

2 Encoder-Decoder Architecture

Our model is implemented as an RNN Encoder-Decoder with attention, built to imitate the model introduced by Kann and Schütze (2016), variations of which found much success in the 2017 CoNLL-SIGMORPHON shared task. The system, pictured in Figure 1, works by learning an encoder RNN over a sequence of embeddings for the input characters or morphological tags. In practice, the encoder is bidirectional. The decoder RNN is initialized with a sequence boundary token, and each state of the decoder is predicted based on the state of the previous timestep, the previous

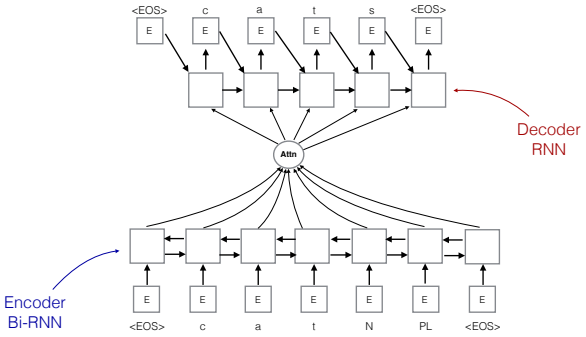

Figure 1: The encoder-decoder with an attention mechanism used for morphological inflection

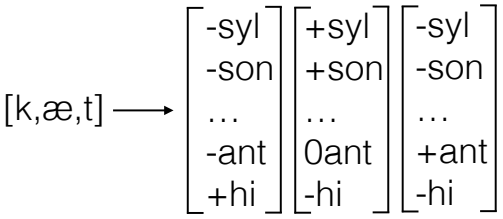

Figure 2: PanPhon transforms a sequence of IPA segments into a matrix of features

output embedding, and all of the encoder states $e_i \in \textit{Encoder}$. We then use an attention mechanism (Bahdanau et al., 2014) to 'attend' over the encoder states, assigning a score to each e_i given the previous decoder state d_{j-1}. The scoring function (Luong et al., 2015) is calculated as

$$score(e_i; d_{j-1}) = tanh([d_{j-1}; e_i] \times W) \quad (1)$$

where W is a parameter matrix that is learned during training, and $[x; y]$ indicates the concatenation of x and y. These scores are then normalized by applying a softmax over all encoder states in *Encoder* to compute each $\epsilon_{i,j-1}$.

Finally, the attention vector is computed as the weighted mean of all encoder states according to their normalized score: $A(d_{j-1}, E) = \sum_{i=0}^{n} \epsilon_{i,j-1} e_i$, which is concatenated to the previously decoded embedding before being passed through the decoder. We implement this model in PyTorch (Paszke et al., 2017), using Gated Recurrent Units (GRU) (Cho et al., 2014) for the encoder and decoder, and optimize with stochastic gradient descent.

3 Embedding Inputs

The inputs to this model are sequences of character and tag embeddings. To this end, each Unicode character codepoint or tag is a one-hot vector c, and an embedding matrix $E \in \mathbb{R}^{|\Sigma| \times n}$ is

	Medium				High			
	Shared task	Text	IPA	Features	Shared task	Text	IPA	Features
English	94.70	89.70	72.50	72.70	97.20	96.60	77.00	76.10
German	80.00	71.80	60.80	59.90	93.00	89.50	82.30	83.20
Hindi	97.40	84.80	89.80	86.50	100.00	99.80	99.60	99.90
Hungarian	75.10	67.10	65.50	63.90	86.80	83.70	82.80	82.60
Persian	91.90	84.10	85.00	82.30	99.90	99.10	99.20	98.70
Polish	79.90	67.10	63.40	64.30	92.80	88.20	88.20	88.90
Russian	84.10	66.90	66.80	67.60	92.80	89.20	86.50	88.90
Spanish	91.70	76.50	82.10	81.30	97.50	95.20	96.40	96.60
AVG	**86.85**	**76.21**	**72.61**	**72.73**	**95.0**	**92.66**	**89.00**	**89.36**

Table 1: Overall accuracy for each model (orthographic, IPA-based, and distinctive feature-based), and comparison with the CoNLL-SIGMORPHON 2017 shared task top performing system for each language.

computed to store the parameters that map the $|\Sigma|$-dimensional one-hot vectors to n-dimensional dense vectors, where Σ is the character and tag vocabulary. Similarly we use a matrix $I \in \mathbb{R}^{|\Sigma_{IPA}| \times n}$ for embedding IPA segments, where Σ_{IPA} is the IPA segment and tag vocabulary. To produce the IPA sequence, we use the Python library Epitran, which performs rule-based G2P on language specific mappings (Mortensen et al., 2018).

We then use the Python library PanPhon (Mortensen et al., 2016), which maps IPA segments to features as in Figure 2, to obtain vectors of phonological distinctive features. The features are represented numerically whereby each index of the vector corresponds to a specific feature such as [±coronal] and stores a value from the set {1, 0, -1}. These values correspond to 'exhibits feature', 'unspecified for given class of sounds', and 'does not exhibit feature', respectively. In practice we map -1 to 0 to obtain strictly binary feature vectors. Now, each IPA segment can be represented as a vector v which has a 1 for each feature that it exhibits, and a 0 otherwise. The embedding matrix $F \in \mathbb{R}^{|p| \times n}$, where p is all features, tags, and symbols is no longer just a lookup for IPA segments. Tags are still one-hot vectors, and symbols are one-hot vectors for any character that has no phonological features (e.g. a space, or apostrophe). But the vector for an IPA segment now has a one for each feature that it exhibits, in contrast to the one-hot vectors.

The operation vF is equivalent to summing each F_i for which $v_i = 1$. In this way, an IPA segment is the sum of all of its distinctive feature embeddings. In practice, we can take the matrix-matrix product of the entire sequence of feature vectors and F to calculate the matrix that represents a sequence of embeddings. The overall workflow involves passing from orthographic input sequences, through Epitran, and then PanPhon, and finally to phonological distinctive feature embeddings.

4 Experiments and Results

We evaluate these models on 8 languages that are at the intersection of CoNLL-SIGMORPHON data, Epitran, and PanPhon supported languages, selected to exhibit typological diversity. The languages, split into 2 training settings per the shared task data: Medium (∼1,000 training examples), and High (∼10,000 training examples), and their accuracies are given in Table 1. In the high data setting using orthographic inputs, our implementation performed comparably to the best shared task systems for each language. The slight degradation in performance can be attributed to the fact that we did not use ensemble voting, as the top performing systems in the shared task did (Cotterell et al., 2017), and that this is a comparison to the maximum score of 25 systems per language, which increases the likelihood that the optimal initialization will have been found. In the medium setting, the difference in accuracy is much more apparent. This is due to the fact that all of the top performing systems in the shared task also used either some type of data augmentation method (Zhou and Neubig (2017), Silfverberg et al. (2017), Sudhakar and Kumar (2017), Kann and Schütze (2017), Bergmanis et al. (2017)) a hard alignment method (Makarov et al., 2017), or both (Nicolai et al., 2017). These results illustrate the common observation that neural systems require a large amount of data to be very accurate,

	Ensemble Oracle Results			
	Medium		High	
	All inputs	Text	All inputs	Text
English	94.20	94.20	97.90	97.50
German	79.50	81.20	93.10	94.30
Hindi	94.50	90.30	100.00	100.00
Hungarian	78.60	77.40	90.10	90.10
Persian	91.40	90.40	99.70	99.60
Polish	79.40	78.70	93.70	92.6
Russian	78.30	76.80	94.10	92.50
Spanish	89.2	86.90	98.5	97.0
AVG	**85.64**	**84.49**	**95.89**	**95.45**

Table 2: Ensemble Oracles for each language. If the correct word form is predicted by any of the 3 models, then it is classified as correct. This is compared against 3 text models.

which can be partially addressed by artificially expanding the training data, or enforcing some copy bias into the system.

For both phonetic representation experiments, the decoded outputs are in the inventory of IPA segments, the gold standard of which comes from the deterministic mappings implemented in Epitran. This means that they differ only in terms of the input representation in the encoder. Models trained on both IPA and feature inputs perform comparably to the text model on both the medium and the high setting. There are two main points of interest in the results. (1) The lower performance on average of the IPA and feature models when compared to the text model is almost exclusively due to differences in accuracy for German and English. We attribute this on the one hand to the fact that the orthography of English is often dissimilar to pronunciations and that their orthographies reflect etymological information which is useful in determining a word's inflectional behavior (Scragg, 1974). An example of the discrepancy between spelling and pronunciation is that the English vowel space has about 13 phonetic vowels (Ladefoged and Johnson, 2014), whereas in the orthographic alphabet, there are only 5. Furthermore, the unstressed vowel, schwa (ə), can essentially replace any vowel in an unstressed context. We observe that the majority of inaccuracies in the English predictions are related to vowels, and most commonly to a schwa. This indicates that converting the character space to IPA can introduce some new complications. Regarding German, there is no obvious explanation for the lower accuracy, and we believe that a more detailed analysis of the G2P performance is needed in order to explore this. Experiments on orthographically and morphophonemically similar languages may also be revealing.

An Ensemble Oracle of all three models is given in Table 2 in order to check if the systems vary in what they learn to predict. The results show that this ensemble outperforms each individual system for any given language. However, when compared to an Ensemble Oracle of three text models, the results are rather similar. The increase in accuracy may simply be due to varying parameters from different random initializations, yielding an effect that is similar to the boosted scores that can be observed in many of the shared task results.

More interesting is the fact that (2) both the IPA and feature representation seem to yield extremely similar accuracies with a paired permutation test p-value of 0.43 over all languages. Even when the training data is rather sparse as in the medium setting, the accuracies remain extremely similar. This suggests that the distributional properties of IPA segments capture the information expressed by distinctive features. Any benefit that representing a segment in terms of its features might have is *already available in the IPA embeddings*. To further compare these representations, we experiment with models that combine the IPA and feature representations. We attempt to simply add a 'feature' to the distinctive feature vectors for each IPA segment. That is, the feature vector for /ə/ would have a 1 for all of its distinctive features, and an additional 1 for that specific segment. We also experiment with concatenation of the embedding found from the feature vector combination and the IPA embedding. The input to the model is a vector of double the embedding size to account for concatenation. The results, given in Table 3, show that neither experiment seems to have much effect, and the accuracies reflect the initial results.

5 Conclusion and Future Work

We have experimented with morphological inflection on 8 different languages and compared results between an input space of IPA segments, and one represented as bundles of phonological distinctive features. The results show that both types of inputs behave similarly. This indicates that the distributional properties of IPA segments align with those found by phonological distinctive features, at least to the extent that articulatory information is relevant to inflection. Furthermore, when compared to a baseline of a purely orthographic space, it is ev-

	Input Combination Results			
	Medium		High	
	Addition	Concat	Addition	Concat
English	71.10	68.70	77.40	77.70
German	56.60	56.40	84.40	84.50
Hindi	93.00	90.80	99.70	99.70
Hungarian	62.90	63.70	83.50	83.40
Persian	84.20	84.10	99.20	99.60
Polish	60.80	66.60	88.90	89.20
Russian	65.90	64.80	90.50	90.00
Spanish	80.80	84.40	97.80	97.50
AVG	**71.91**	**72.44**	**90.18**	**90.20**

Table 3: Results for the combination of feature and IPA embeddings. Addition refers to the inclusion of a specific phoneme 'feature'. Concat refers to concatenating the embedding of both input types.

ident that for many languages the results are still mostly redundant, and if there is a large discrepancy in accuracy it is in favor of the orthographic inputs.

There is still work to be done to explore if there are scenarios where bundles of distinctive features provide an advantage. That is, in the case of transfer learning where the phonology of a language is known, it becomes possible to approximate vector representations for unseen segments. Similarly, distinctive features may be better at representing segments that rarely appear in a training set for a given language.

Acknowledgements

The second author was supported by The Society of Swedish Literature in Finland (SLS). NVIDIA Corp. donated the Titan Xp GPU used for this research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

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, Berlin, Germany. Association for Computational Linguistics.

Akash Bharadwaj, David R. Mortensen, Chris Dyer, and Jaime G. Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1462–1472.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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*, Vancouver, Canada. 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*, Berlin, Germany. Association for Computational Linguistics.

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*, Berlin, Germany. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2017. The LMU system for the CoNLL-SIGMORPHON shared task on universal morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 40–48, Berlin, Germany. Association for Computational Linguistics.

Peter Ladefoged and Keith Johnson. 2014. *A Course in Phonetics*. Nelson Education.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

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, Berlin, Germany. Association for Computational Linguistics.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh Interna-*

tional Conference on Language Resources and Evaluation (LREC 2018), Paris, France. European Language Resources Association (ELRA).

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *The 26th International Conference on Computational Linguistics: Technical Papers*, page 34753484.

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. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 79–84, Berlin, Germany. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Donald George Scragg. 1974. *A history of English spelling*, volume 3. Manchester University Press.

Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL)*, volume 1, pages 136–144.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Berlin, Germany. Association for Computational Linguistics.

Akhilesh Sudhakar and Anil Singh Kumar. 2017. Experiments on morphological reinflection: CoNLL-2017 shared task. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 71–78, Berlin, Germany. Association for Computational Linguistics.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David R. Mortensen, Alan W. Black, Lori S. Levin, and Chris Dyer. 2016. Polyglot Neural Language Models: A Case Study in Cross-Lingual Phonetic Representation Learning. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1357–1366.

Chunting Zhou and Graham Neubig. 2017. Morphological inflection generation with multi-space variational encoder-decoders. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 58–65, Berlin, Germany. Association for Computational Linguistics.

This is investigating G2P transcription and phonemic features as preprocessing, but for in the case of a single learning language, not cross-lingual learning as we do in our paper. Probably should cite in the introduction were we talk about other relevent work.