Morphological Inflection with Phonological Features

David Guriel, Omer Goldman, Reut Tsarfaty
Bar-Ilan University
{davidgu1312,omer.goldman}@gmail.com,reut.tsarfaty@biu.ac.il

Abstract

Recent years have brought great advances into solving morphological tasks, mostly due to powerful neural models applied to various tasks as (re)inflection and analysis. Yet, such morphological tasks cannot be considered solved, especially when little training data is available or when generalizing to previously unseen lemmas. This work explores effects on performance obtained through various ways in which morphological models get access to sub-character phonological features that are often the targets of morphological processes. We design two methods to achieve this goal: one that leaves models as is but manipulates the data to include features instead of characters, and another that manipulates models to take phonological features into account when building representations for phonemes. We elicit phonemic data from standard graphemic data using language-specific grammars for languages with shallow grapheme-to-phoneme mapping, and we experiment with two reinflection models over eight languages. Our results show that our methods yield comparable results to the grapheme-based baseline overall, with minor improvements in some of the languages. All in all, we conclude that patterns in character distributions are likely to allow models to infer the underlying phonological characteristics, even when phonemes are not explicitly represented.

1 Introduction

In recent years, morphological tasks received much attention in NLP through various tasks such as (re)inflection, lemmatization and others, specifically through the SIGMORPHON shared tasks (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020; Pimentel et al., 2021). State-of-the-art models seem to achieve quite high results in such cross-lingual evaluation campaigns, although recent works showed that there is still room for improvements (Goldman et al., 2022).

Most studies aiming at morphological tasks design models that operate at the character level, without reference to the phonological components that compose the phonemes represented by the characters.[1] This is despite the fact that many morphological processes have distinct phonological features, rather than phonemes, as either the trigger or target of morphological processes. For example, in vowel harmony, a single feature of a vowel in the stem determines the vowels that appear in the affixes added to that stem. Without direct evidence of the phonological features composing every phoneme, models must resort to memorizing groups of phonemes that pattern together for an unobserved reason.

In this work we hypothesize that explicitly inputting models with phonological features will lead to better modelling of morphological tasks. We set out to equip models with two alternative methods for incorporating that information. One method replaces the character-level tokens with phonological feature tokens; and another one equips the model with a self-attention mechanism that learns representation of phonemes from their features.

We implement these methods on the task of morphological reinflection, where forms of the same lemma are inflected from one another. We experiment with 8 languages and 2 models: an LSTM encoder-decoder with global attention by Silfverberg and Hulden (2018); and a transducer by Makarov and Clematide (2018) that predicts edit actions between source and target word-forms and is suitable for lower amounts of training data.

Our experiments show that the proposed methods yield results comparable to the grapheme-based baseline setting for the transducer model. On average across languages, the best phonologically-aware method suffered from a drop of 2.8 accuracy points, although the performance on some individual languages marginally improved. We thus

[1] Some exceptions do exist, like Malouf (2017)'s model that operates over phonemes rather than characters.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 613–622

conjecture that the phonological characteristics are already encoded in the graphemic representations elicited by this model. The results of this work are in line with other works, performed in different settings, investigating the role of phonology in morphological models (see Section 6).

We further note that the LSTM model, unlike the transducer, did not perform well on graphemic data and suffered from a severe drop when applied on phonological data in all tested languages. We attribute this to the transducer's attested ability to perform well particularly in low-resource setting. We subsequently conjecture that, for the phonologically-aware variant of the reinflection task, standard amounts of reinflection data should be effectively considered low-resourced.

2 Morpho-Phonological Processes

Utterances in natural language — sentences and words — are composed of phonemes. Yet, one can further decompose phonemes to their very atomic elements: *phonological distinctive features*. A phonological feature is the minimal unit within a phoneme that distinguishes it from other phonemes. Every phoneme can be described as a unique combination of such features. Vowels, for example, are said to take the features: *backness* of the tongue, *height* of the lower jaw, and *roundness* of the lips; the sound /a/ then has the values *front*, *open* and *unrounded*. Consonants usually take the features: *place of articulation*, *manner of articulation* and *voiceness*, e.g. /g/ has the values *velar*, *plosive* and *voiced*.[2]

Many languages exhibit morphological processes whose target or trigger are phonological features. For instance, Turkish has vowel harmony at the backness feature: the stem's last vowel controls (*harmonizes*) the backness of other vowels in morphemes added to that stem. Table 1 illustrates the alternation for future tense inflection. For *ol*, the future morpheme includes the back vowel /a/, according to the backness of the vowel /o/. In *öl*, however, the vowel /œ/ is front, so the morpheme includes the front vowel /e/.

When a character-level inflection model learns this process, it has to memorize the relation between the letters representing vowels of the same backness (including 4 back vowels and 4 front vowels) instead of aligning vowels explicitly by their backness feature. In general, describing such processes at the grapheme level is often intricate and requires models trained on morphological tasks to put unnecessary effort in learning patterns that are more complicated than their original cause. Because the underlying phonological information is not explicitly shown to them, instead of learning simple rules of phonological features, they memorize groups of characters that pattern together for no observable reason.

	Stem	Future Tense
'be'	*ol*	*olacak*
	/ol/	/ola$\widehat{\text{ʤ}}$ak/
'die'	*öl*	*ölecek*
	/œl/	/œlɛ$\widehat{\text{ʤ}}$ɛk/

Table 1: Vowel harmony in Turkish: the future tense allomorph changes according to the backness of the stem's vowel.

A model that is aware of phonological features would be able to easily learn these relations and treat morpho-phonological processes straightforwardly. In order to construct such a model there is a need for phonologically annotated data or for a tool that converts words to their corresponding sequences of phonemes (their verbal pronunciation) and decomposes the phonemes into their phonological distinctive features. A simple option would be to employ a component that performs grapheme-to-phoneme (G2P) and phoneme-to-grapheme (P2G) conversions for every language, as well as decomposes the phonemes to their corresponding distinctive features. Thus, every character-level model would be able to process phonological data. In the next section we present two ways to incorporate such signals into the data and models for morphological tasks.

3 Modeling Reinflection with Phonology

We set out to re-model morphological tasks by integrating phonological information, in order to make phonological processes explicitly learnable for models. We propose two generic methods that are applicable to any morphological model.

Formally, we denote 3 alphabets, for graphemes Σ_g, phonemes Σ_p and phonological features Σ_f. The first one is language-dependent while the others are universally defined in the IPA list of symbols

[2]The features of vowel and consonants are not unrelated. For example, *place of articulation* and *backness* are essentially aliases for the same physical feature.

and features (Association, 1999).[3] We treat a word as a tuple of its composing graphemes $\mathbf{g} \in \Sigma_g^+$. Correspondingly, the sequence of phonemes that is the result of applying the G2P component to $\mathbf{g}$ is denoted by $\mathbf{p} \in \Sigma_p^+$, and the phonemes' decomposition to a sequence of features is denoted by $\mathbf{f} \in \Sigma_f^+$.

Suppose we have a morphological task T, in which the input is $\mathbf{g}_{src}$ and the output ground truth is $\mathbf{g}_{trg}$. That is

$$\mathbf{g}_{trg} = T\left(\mathbf{g}_{src}; S\right)$$

where S is a set of bundles of morphological features that complement the input form. In standard inflection tasks, for example, $\mathbf{g}_{src}$ is the lemma and $\mathbf{g}_{trg}$ is the inflected output form, where S is the feature bundle to inflect the lemma to. In reinflection, the forms $\mathbf{g}_{src}$ and $\mathbf{g}_{trg}$ are the input and output forms, and S is the feature bundles of the source and target word forms, e.g. $\{(\text{FUT},2,\text{SG}), (\text{FUT},3,\text{PL})\}$.

We denote M_T as a model that solves T, i.e. it takes $\mathbf{g}_{src}$ and S, and generates $\hat{\mathbf{g}}_{trg}$, a prediction of the target word:

$$\hat{\mathbf{g}}_{trg} = M_T\left(\mathbf{g}_{src}; S\right)$$

In order to incorporate the phonological information to M_T, its inputs should obviously be changed to include this information — either phonemes or phonological features. However, changes can also be done to M_T itself to treat better the new inputs. We thus propose two methods for inducing phonological information to morphological models: one manipulates only the source and target data to include phonological features, and one adds a learnable layer to the model in order to facilitate better processing of the new input. Both methods leave S untouched, the model processes S in the exact same way as in the graphemic setting.

Data Manipulation In the first method, we propose to train M_T on the *phonological features* of the source and target words, $\mathbf{f}_{src}$ and $\mathbf{f}_{trg}$, instead of their letters. We do not modify M_T or the way it processes S, the model simply operates directly on the modified representations.

$$\hat{\mathbf{f}}_{trg} = M_T\left(\mathbf{f}_{src}; S\right)$$

The network is then optimized with a given loss function ℓ by comparing between the predicted features and the gold target word converted to features:

$$\mathcal{L} = \mathbb{E}\left[\ell\left(\hat{\mathbf{f}}_{trg}, \mathbf{f}_{trg}\right)\right]$$

A clear disadvantage of this method is that the resulting sequences are much longer than the original ones, in practice approximately 3-4 times longer.

Model Manipulation In the second method, we also manipulate the model in accordance with the new data format. We let the model learn a phonemic representation in a way that is aware of the phoneme's phonological features. To this end, we add a self-attention layer (Vaswani et al., 2017) between the embedding matrices to the rest of the network. This layer takes the embeddings of a phoneme $E\left[\mathbf{p}_{src}\right]$ and its features $E\left[\mathbf{f}_{src}\right]$, and learns a single vector per phoneme $\widetilde{\mathbf{p}}_{src}$. The network is then trained to predict the phonemes of the target word:

$$\begin{aligned}
\hat{\mathbf{p}}_{trg} &= M_T\left(\widetilde{\mathbf{p}}_{src}; S\right) \\
\widetilde{\mathbf{p}}_{src} &= \text{SelfAttention}\left(q, K, V\right) \\
K, V &= E\left[\mathbf{f}_{src}\right] \\
q &= E\left[\mathbf{p}_{src}\right],
\end{aligned}$$

where the self-attention is computed as follows (where d is the output dimension and n is the number of heads):

$$\widetilde{\mathbf{p}}_{src} = \text{softmax}\left(\frac{qK^T}{\sqrt{d/n}}\right) \odot V$$

The model is optimized similarly to the first method, except the compared sequences are the predicted phonemes and the gold target word converted to phonemes:

$$\mathcal{L} = \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}_{trg}, \mathbf{p}_{trg}\right)\right]$$

The advantage of this method over the previous one is that the input to the inner network is of the order of magnitude of the number of phonemes, and not the number of features. This leads to more reasonable lengths of the inputs, but it relies more heavily on the model to learn to combine feature representations correctly.

[3] The IPA features we use here may be better described as coarse phonetic features rather than purely phonological, since in some rear language-specific cases there is a mismatch between the phonological behavior of a phoneme and its phonetic properties. However, the scarcity of these cases led to the general usage of IPA features as phonological descriptors and made most linguists consider phonetics and phonology as a unified grammatical mechanism (e.g., Ohala, 1990; Pierrehumbert, 1990).

4 Experiments

Models We applied the described methods to two character-level models.[4] Both were modified to solve reinflection instead of inflection and to handle phonemic symbols and phonological features:

- *LSTM*: a standard LSTM Encoder-Decoder model with global attention[5] as proposed in Silfverberg and Hulden (2018).
- *Transduce*: An LSTM-based model by Makarov and Clematide (2018) predicting edit actions between the source and the target. This model is more robust in low-resource settings.

Data We experimented with eight languages: Swahili, Georgian, Albanian, Bulgarian, Latvian, Hungarian, Finnish and Turkish, in three part-of-speech types. All of these languages have shallow orthography, i.e., nearly one-to-one G2P and P2G mappings. We purposefully selected such languages to be able to disentangle the effects of convoluted orthographies from the potential benefits of phonetic decomposition to features, and to avoid the use of trainable G2P and P2G models that would inevitably propagate errors and serve as a confounding factor. We compared the two proposed methods to the baseline where the models take letters as the source and target tokens.

We randomly sampled 10,000 reinflection samples from the UniMorph 3.0 repository (McCarthy et al., 2020) for train, validation and test sets, with 80%-10%-10% splits ratios. The split was done such that the sets would have no overlapping lemmas, following Goldman et al. (2022). The models were trained separately for each language and POS.

Preprocessing Due to the orthographic shallowness of the selected languages we were able to implement for each language a rule-based component for G2P and P2G conversions.

Evaluation Two evaluation metrics are reported: exact match accuracy and averaged edit distance. For comparability, all predictions were measured at the grapheme level, by converting the predictions back to graphemes using the P2G component.[6]

[4]All our code is available at https://github.com/OnlpLab/InflectionWithPhonology

[5]Not to be confused with the self-attention layer applied in the *model manipulation* method.

[6]In case the conversion component could not find a matching phoneme to the sequence of features, it used an out-of-vocabulary token '#'.

Model	Method			Average
	Baseline	Data Manipulation	Model Manipulation	
LSTM	**46.5%±0.8%**	26.4%±0.5%	10.9%±2.2%	27.9%±0.5%
Transduce	**83.6%±0.2%**	80.3%±0.2%	80.8%±0.9%	81.6%±0.2%

Table 2: Graphemic Accuracy of all systems, averaged on all language-POS datasets, and averaged over 3 seeds. Highest value per row is **bold**.

5 Results and Analysis

Table 2 shows the results of the two systems across the two methods, compared to the graphemic baseline, averaged over languages. The *LSTM* model performs poorly, with 46 accuracy points at the baseline, and less than 30 points in the novel methods. The *Transduce* model performed much better in general, with more than 80 points in all 3 settings. On average over the 15 language-POS combinations, training on our methods resulted in a slight drop of 2.8 points, which makes them comparable with the baseline. These results may imply that our methods fit better to stronger models, and that this setting and quantities may be considered as low-resource, at least without hallucination methods like that of Anastasopoulos and Neubig (2019).

Table 3 breaks down the results of the *Transduce* model per language. In 7 out of 15 datasets, at least one of our methods outperformed the baseline. The difference varies from 0.9 up to 11.7 accuracy points. All in all, it seems that there is no connection between the relative success of the phonologically-aware methods and the abundance of morpho-phonological processes in a language. In Turkish, for instance, that has vowel harmony and additional phonological processes, the baseline performed much better, while in Swahili and Georgian (which barely exhibit such processes) there were clear improvements.

To provide insights into the sufficiency of the data and the richness of the signal, we plot on figure 1 (in appendix A) learning curves for the *Transduce* model per language. We trained each model over an increasing number of train samples from 1,000 to 8,000 and evaluated them on the same test sets for each language. The general trends show that the amount of data is indeed sufficient for the model and the signal is not richer, as in most cases the test accuracy with 8,000 samples is similar to the one with 3,000 samples. Moreover, the graphs show that our methods have no clear advantage over the baseline even in as few as 1,000 training examples.

Language	POS	Method		
		Baseline	Data Manipulation	Model Manipulation
Bulgarian	Adj	**96.6%±0.4%**	95.5%±1.2%	95.7%±2.4%
Bulgarian	V	**89.0%±1.1%**	87.6%±1.0%	88.0%±1.5%
Finnish	Adj	**94.2%±0.5%**	92.8%±0.2%	92.8%±0.1%
Finnish	N	82.3%±0.8%	**83.1%±0.9%**	78.2%±0.9%
Finnish	V	**88.1%±2.1%**	79.8%±2.8%	84.3%±1.0%
Hungarian	V	**90.9%±1.1%**	89.6%±0.5%	89.7%±0.8%
Georgian	N	90.2%±0.5%	**91.4%±0.8%**	90.3%±0.6%
Georgian	V	42.2%±2.0%	28.4%±1.5%	**44.2%±4.1%**
Latvian	N	88.4%±0.8%	**90.0%±0.6%**	85.6%±0.5%
Latvian	V	**76.5%±0.9%**	70.9%±0.9%	67.9%±1.9%
Albanian	V	84.3%±1.0%	79.6%±1.4%	**86.9%±2.2%**
Swahili	Adj	66.7%±2.9%	**74.4%±4.5%**	64.4%±12.6%
Swahili	V	90.9%±1.0%	87.0%±2.1%	**92.4%±1.2%**
Turkish	Adj	**91.6%±2.1%**	79.0%±4.3%	76.8%±2.3%
Turkish	V	**82.5%±0.5%**	75.8%±2.1%	74.9%±0.9%
Average		**83.6%±0.2%**	80.3%±0.2%	80.8%±0.9%

Table 3: Graphemic Accuracy of the *Transduce*, averaged over 3 seeds. Highest value per row is in **bold**.

6 Discussion and Conclusion

In this work we incorporated phonological information into morphological tasks. We proposed two methods: one that modifies the data, and one that also manipulates the model. We exemplified them on reinflection for two models and found out that, on average, our methods are comparable with the baseline and do not surpass it. We conclude that the embeddings obtained for the graphemic representations in such tasks may already encode the underlying phonological information in the data.

This conclusion is in line with the work of Wiemerslage et al. (2018), who similarly aimed, with no success, to use phonological data in morphological inflection. Unlike our work, they used a weaker inflection model as a baseline for modification and they had a different method in constructing the phonologically-aware embeddings. More crucially, they experimented with a *form-split* setting, which means that there was significant overlap between the sampled lemmas in the train-test split. Our results also corroborate the findings of Silfverberg et al. (2018), who examined phoneme embeddings from various sources, including from a morphological inflection model, and showed that they implicitly encode phonological features, thus supporting our main conclusion.

Limitations

One limitation of our work is the experimentation only with languages with shallow orthographies, i.e. relatively simple G2P and P2G mappings. The results might vary for deeper-orthographies languages.

Although we took extra care to verify our conversions are correct and complete, and designed the rules to be as comprehensive as possible, automatic rule-based processes in languages may not be 100% perfect and some corner cases may introduce errors. These errors may propagate to affect the numerical results. To mitigate this issue, when ambiguities in determining a target phoneme (or grapheme) in a given language occur, we purposefully select the values that occur more frequently in the UniMorph data of that particular language.

Acknowledgements

This research is funded by a grant from the European Research Council, ERC-StG grant number 677352, and a grant by the Israeli Ministry of Science and Technology (MOST), grant number 3-17992, for which we are grateful.

References

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. Association for Computational Linguistics.

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. 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, Géraldine 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, Vancouver. 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.

Omer Goldman, David Guriel, and Reut Tsarfaty. 2022. (un)solving morphological inflection: Lemma overlap artificially inflates models' performance. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 864–870, Dublin, Ireland. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2018. Neural transition-based string transduction for limited-resource setting in morphology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 83–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Robert Malouf. 2017. Abstractive morphological learning with a recurrent neural network. *Morphology*, 27(4):431–458.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

John J Ohala. 1990. There is no interface between phonology and phonetics: a personal view. *Journal of phonetics*, 18(2):153–171.

Janet Pierrehumbert. 1990. Phonological and phonetic representation. *Journal of phonetics*, 18(3):375–394.

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics.

Miikka Silfverberg and Mans Hulden. 2018. An encoder-decoder approach to the paradigm cell filling problem. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2889, Brussels, Belgium. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Adam Wiemerslage, Miikka Silfverberg, and Mans Hulden. 2018. Phonological features for morphological inflection. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 161–166, Brussels, Belgium. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction.

In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

A Learning Curves

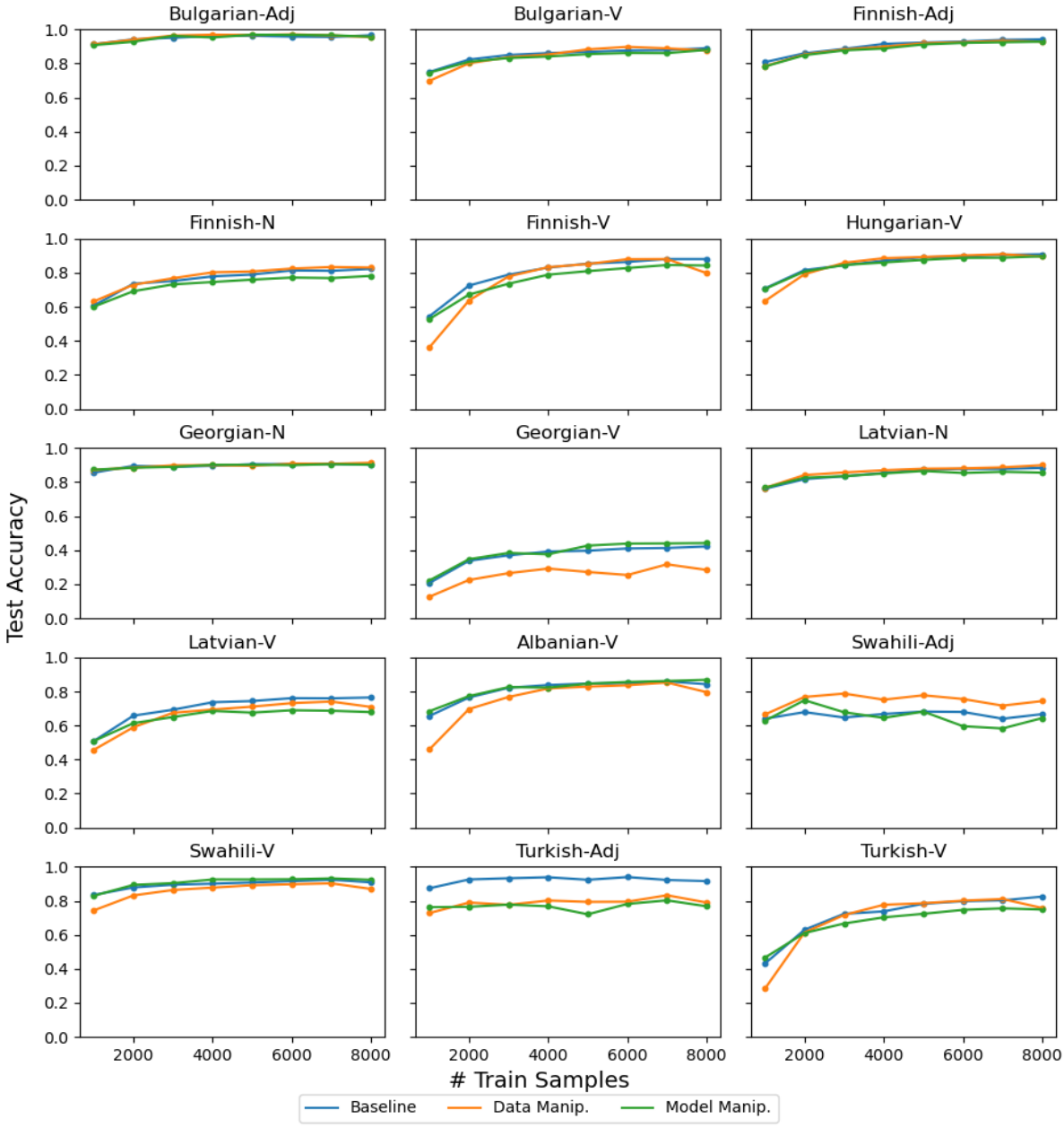

Figure 1: Learning curves for accuracy over test sets for each language-POS dataset, as a function of the train set size.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Described in the Limitations section

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4

☑ B1. Did you cite the creators of artifacts you used?
4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4

C ☑ Did you run computational experiments?

4

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.