Can a Transformer Pass the Wug Test? Tuning Copying Bias in Neural Morphological Inflection Models

Ling Liu and **Mans Hulden**
University of Colorado
`first.last@colorado.edu`

Abstract

Deep learning sequence models have been successful with morphological inflection generation. The SIGMORPHON shared task results in the past several years indicate that such models can perform well, but only if the training data covers a good amount of different lemmata, or if the lemmata to be inflected at test time have also been seen in training, as has indeed been largely the case in these tasks. Surprisingly, we find that standard models such as the Transformer almost completely fail at generalizing inflection patterns when trained on a limited number of lemmata and asked to inflect previously unseen lemmata—i.e. under "wug test"-like circumstances. This is true even though the actual number of training examples is very large. While established data augmentation techniques can be employed to alleviate this shortcoming by introducing a copying bias through hallucinating synthetic new word forms using the alphabet in the language at hand, our experiment results show that, to be more effective, the hallucination process needs to pay attention to substrings of syllable-like length rather than individual characters.[1]

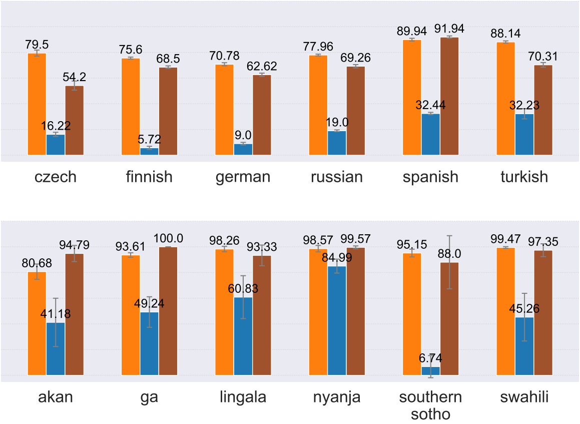


Figure 1: Transformer performance in the **common-practice** setting (left), "**wug test**"-like setting (middle), and '**'wug test"-like setting with our best data hallucination** method (right)

1 Introduction

The Transformer model has delivered convincing results in many different tasks related to word-formation and analysis (Vylomova et al., 2020; Moeller et al., 2020, 2021; Liu, 2021). Especially on inflection tasks, where an input lemma such as `dog`, and input inflectional features such as `{N,PL}`, are expected to produce an output such as `dogs`, the model has shown to be particularly adept at generalizing patterns (Vylomova et al., 2020; Liu and Hulden, 2020a,b; Wu et al., 2021). However, we have discovered that this is only true if the training data covers a diversity of lemmata or *some* variant of the input lemma to be inflected has been witnessed during training. In a "wug test" (Berko, 1958) setting where the witnessed lemmata are usually limited and a previously unseen lemma—like `wug`—is to be inflected in some way, we find that the Transformer almost completely fails to generalize inflection patterns, despite abundant inflected forms for training. It has been noted earlier that neural sequence-to-sequence models are apt to perform poorly for morphological inflection if they have been exposed to little training data and data augmentation can be leveraged to alleviate the problem (Cotterell et al., 2017, 2018; Kann and Schütze, 2017; Liu and Hulden, 2021). Our starting point is our observation that the poor "wug test" performance is maintained even with abundant training inflected forms.

In our study, we show three main results. (1) We demonstrate that, even if trained with relatively large amounts of inflected forms, a Transformer model of the kind that has been very successful at recent shared tasks largely fails to generalize inflection patterns if it has not been exposed during training to a variety of lemmata or any lemmata in

[1] The code and data are available at `https://github.com/LINGuistLIU/transformer-wug-test`.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 739 - 749
May 22-27, 2022 ©2022 Association for Computational Linguistics

the test set. This is true even for datasets where all words inflect in the same way—i.e. there are no inflectional classes or allomorphs of morphemes, as is found in the low-resource Niger-Congo language datasets used in SIGMORPHON 2020 shared task (Vylomova et al., 2020). (2) We show that simply exposing the model to uninflected lemmata in the test set—without providing a single inflected form—allows the model to dramatically improve its performance when inflecting such lemmata. (3) Further, we investigate several strategies that avoid leveraging test set lemmata. We show that when inducing a copy bias in the model by hallucinating new lemmata, or by hallucinating new inflected forms, the method of hallucination is much more effective if it is sensitive to substrings of syllable-like length rather than individual characters or stems. Our best models achieve substantial improvement upon earlier state-of-the-art data hallucination methods (Silfverberg et al., 2017; Anastasopoulos and Neubig, 2019).

2 Data

2018-languages We use six languages from the CoNLL-SIGMORPHON 2018 shared task 1 medium setting, where each language has 1,000 (LEMMA, TARGET TAGS, TARGET FORM) triples for training (Cotterell et al., 2018). The six languages, Czech, Finnish, German, Russian, Spanish and Turkish, are selected to represent the diversity of language typology and morphological inflection challenges. Though there are only 1,000 training triples, they cover a fair number of lemmata as each lemma appears only once or twice, an amount very hard to obtain for really low-resource languages. In the original shared task, between 2% and 27% of the lemmata in the dev and test sets are also found in the training set.

To prepare training data for the "wug test"-like circumstance, we select the UniMorph (Kirov et al., 2018) paradigms for the first 100 most frequent lexemes found in Wikipedia text,[2] which are not included in the 2018 shared task 1 dev and test sets. The shared task dev and test sets are used for validation and evaluation without any change. The 100 full inflection tables give us over 1,000 (for Czech, German and Russian) or over 7,000 (for Finnish, Spanish and Turkish) training triples.

Niger-Congo languages In addition, we use six Niger-Congo languages from SIGMORPHON 2020 shared task 0 (Vylomova et al., 2020): Akan, Ga, Lingala, Nyanja, Southern Sotho and Swahili. These languages are low-resource, but the dataset only contains very regular inflections. In the original shared task data split, The overlap between the lemmata in the dev and test sets and those in the training set is 100%. The number of paradigms which we can obtain by combining the training, dev and test sets of this dataset is around 100 for Akan, Ga and Swahili, 227 for Nyanja, 57 for Lingala and only 26 for Southern Sotho.

For the "wug test", we divide the inflection tables reconstructed from this dataset into a 7:1:2 train-dev-test split, i.e. we use the same ratio as the shared task, but the division is by inflection tables rather than lemma-tag-form triples, to ensure that the lemmata used for validation and test are disjoint from those for training. We provide details on the data statistics in Appendix A for reference.

3 Experiments

Inflection model The Transformer (Vaswani et al., 2017) is the seq2seq architecture which produces the current state-of-the-art result on the morphological inflection task (Vylomova et al., 2020; Liu and Hulden, 2020a,b; Wu et al., 2021). It takes the lemma and target tag(s) as input and predicts the target form character by character. Our experiments use the Transformer implemented in fairseq (Ott et al., 2019) and adopt the same hyperparameters as Liu and Hulden (2020a). [3]

Evaluation metric The evaluation metric is accuracy. For the original shared task data and experiments on 2018 languages, we train five inflection models each with a different random initialization and report the average accuracy with standard deviation. Due to data scarcity, for Niger-Congo languages at the "wug test"-like setting, we perform a 5-fold cross-validation and report the average accuracy and the standard deviation.

Common-practice test and "wug test" We first compare the performance of the Transformer in the common-practice setting and the "wug test"-like setting. The "common practice" is represented by

[2] We also experimented with using 100 random UniMorph lexemes, and did not find substantial difference between using random ones and the most frequent ones.

[3] We also conducted experiments with the encoder-decoder with hard monotonic attention model (Wu and Cotterell, 2019), but found the same conclusion as for the Transformer model. Experiments on the hard monotonic model is provided in Appendix C for reference.

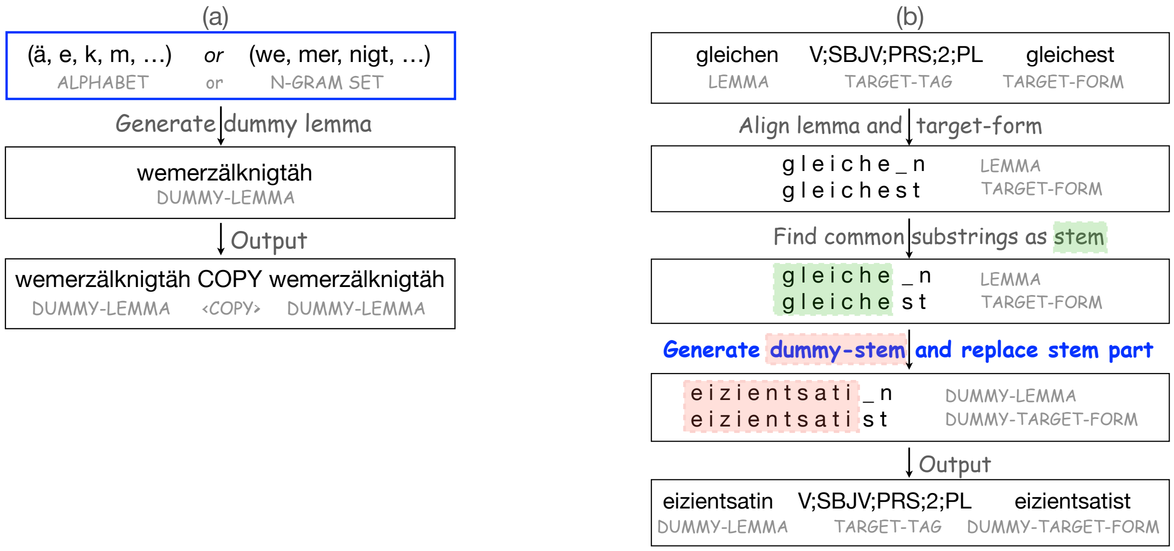

(a)
(ä, e, k, m, ...) or (we, mer, nigt, ...)
ALPHABET or N-GRAM SET
Generate dummy lemma
wemerzälknigtäh
DUMMY-LEMMA
Output
wemerzälknigtäh COPY wemerzälknigtäh
DUMMY-LEMMA <COPY> DUMMY-LEMMA
(b)
gleichen V;SBJV;PRS;2;PL gleichest
LEMMA TARGET-TAG TARGET-FORM
Align lemma and target-form
g l e i c h e _ n LEMMA
g l e i c h e s t TARGET-FORM
Find common substrings as stem
g l e i c h e _ n LEMMA
g l e i c h e s t TARGET-FORM
Generate dummy-stem and replace stem part
e i z i e n t s a t i _ n DUMMY-LEMMA
e i z i e n t s a t i s t DUMMY-TARGET-FORM
Output
eizientsatin V;SBJV;PRS;2;PL eizientsatist
DUMMY-LEMMA TARGET-TAG DUMMY-TARGET-FORM

Figure 2: (a) Dummy lemma generation with a German example. *+copy-2k-char* generates random strings by uniformly sampling from the alphabet, while *+copy-2k-substr* samples from the set of 2-, 3- and 4-grams; (b) Data hallucination with a German example. *+hall-2k-substr* is different from *+hall-2k-char* in how the dummy-stem is generated.

previous years' shared tasks and related work (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020); here the training data usually covers a fair number of lemmata and there is overlap between lemmata in the training and test sets. We use the shared task data to represent the common-practice setting. In the "wug test" setting, we control the number of lemmata for training but not inflected forms (as explained in Section 2) and the lemmata to be inflected are always previously unseen. To our surprise, the performance of the Transformer at the "wug test"-like setting is very poor despite the large amount of training triples for 2018-languages or the very regular and straightforward inflection for Niger-Congo languages. The performance is dramatically inferior to the common-practice setting, even when the number of training triples is seven times larger for Finnish, Spanish and Turkish (see Figure 1).

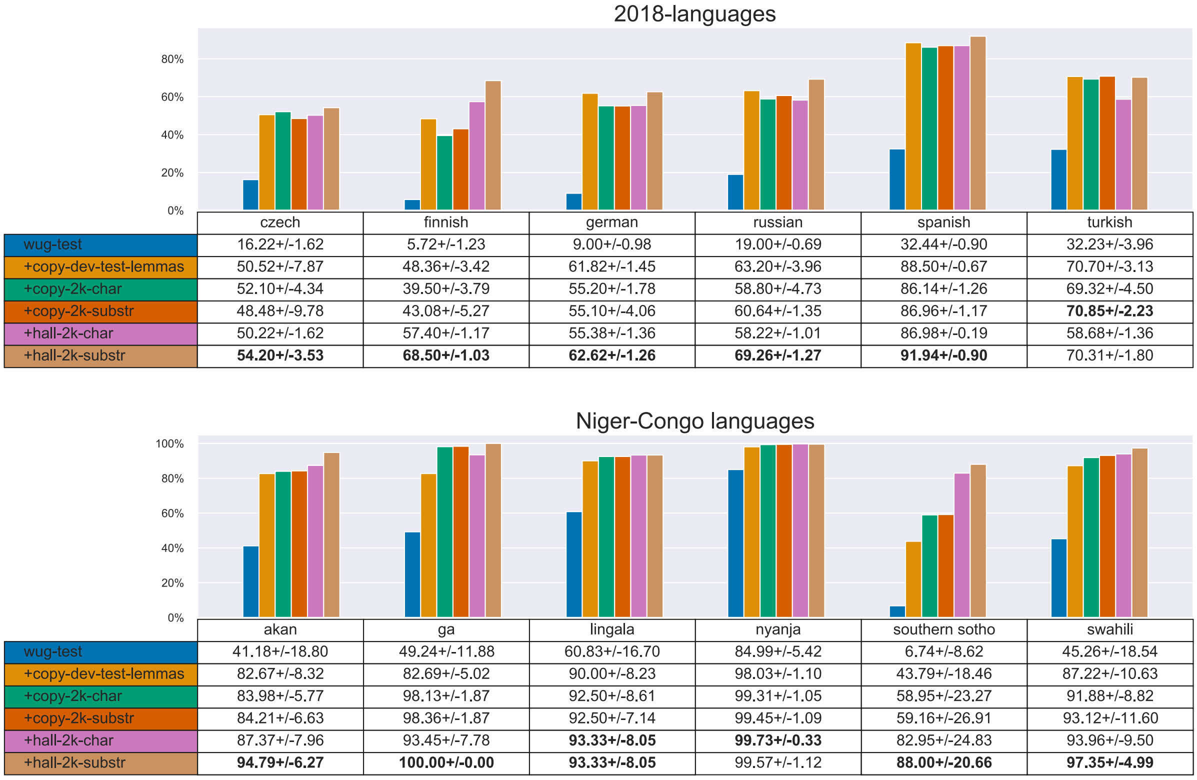
We hypothesize four reasons for the poor performance of the model under the "wug test"-like circumstance: (1) missing copy bias regarding the entire stem, i.e. the model can't copy a stem *abcde* if that exact stem has never been seen during training, (2) missing copy bias on individual letters, i.e. the model can't copy letter *a* if the letter is underrepresented in training, (3) missing copy bias on subsequences of letters, i.e. the model can't copy sequence *ab* if the sequence is underrepresented in training, (4) some combination of all the factors above. To test these hypotheses, we conduct five experiments designed to help the model learn to copy with different biases by adding to the training set for each language 2,000[4] dummy data points generated in five different ways, explained below.

+copy-dev-test-lemmas In order to test the first hypothesis that the model does not learn to copy parts of a stem it has not seen at the training stage, we augment the training data for each language by adding to it the lemmata in its development and test sets with a special tag `COPY`. In other words, 2,000 `(LEMMA, COPY, LEMMA)` triples are added to the initial "wug test" training set for each language.

+copy-2k-char* and *+copy-2k-substr Previous work found that adding random strings can help seq2seq models learn a copy bias and thus improve the performance when the training data is limited (Kann and Schütze, 2017). We adopt a similar method to augment the training data with dummy lemmata generated by the process shown in Figure 2 (a). The *+copy-2k-char* method takes as input the alphabet created by collecting characters in the language's training set.

Considering that a natural linguistic sub-unit of a word is a syllable, we propose to use sub-

[4] The choice of 2,000 is in order to match the augmentation size of *+copy-dev-test-lemmas* method for 2018-languages. We did not try to tune for the best data augmentation size. Appendix B provides plots of data augmentation size comparison, where we found no consistent difference in all the languages.



	czech	finnish	german	russian	spanish	turkish
wug-test	16.22+/-1.62	5.72+/-1.23	9.00+/-0.98	19.00+/-0.69	32.44+/-0.90	32.23+/-3.96
+copy-dev-test-lemmas	50.52+/-7.87	48.36+/-3.42	61.82+/-1.45	63.20+/-3.96	88.50+/-0.67	70.70+/-3.13
+copy-2k-char	52.10+/-4.34	39.50+/-3.79	55.20+/-1.78	58.80+/-4.73	86.14+/-1.26	69.32+/-4.50
+copy-2k-substr	48.48+/-9.78	43.08+/-5.27	55.10+/-4.06	60.64+/-1.35	86.96+/-1.17	**70.85+/-2.23**
+hall-2k-char	50.22+/-1.62	57.40+/-1.17	55.38+/-1.36	58.22+/-1.01	86.98+/-0.19	58.68+/-1.36
+hall-2k-substr	**54.20+/-3.53**	**68.50+/-1.03**	**62.62+/-1.26**	**69.26+/-1.27**	**91.94+/-0.90**	70.31+/-1.80

	akan	ga	lingala	nyanja	southern sotho	swahili
wug-test	41.18+/-18.80	49.24+/-11.88	60.83+/-16.70	84.99+/-5.42	6.74+/-8.62	45.26+/-18.54
+copy-dev-test-lemmas	82.67+/-8.32	82.69+/-5.02	90.00+/-8.23	98.03+/-1.10	43.79+/-18.46	87.22+/-10.63
+copy-2k-char	83.98+/-5.77	98.13+/-1.87	92.50+/-8.61	99.31+/-1.05	58.95+/-23.27	91.88+/-8.82
+copy-2k-substr	84.21+/-6.63	98.36+/-1.87	92.50+/-7.14	99.45+/-1.09	59.16+/-26.91	93.12+/-11.60
+hall-2k-char	87.37+/-7.96	93.45+/-7.78	**93.33+/-8.05**	**99.73+/-0.33**	82.95+/-24.83	93.96+/-9.50
+hall-2k-substr	**94.79+/-6.27**	**100.00+/-0.00**	**93.33+/-8.05**	99.57+/-1.12	**88.00+/-20.66**	**97.35+/-4.99**

Figure 3: "Wug test" results. +*copy-2k-char* adds random strings generated with the alphabet. +*copy-2k-substr* adds random strings generated with the n-gram set. +*hall-2k-char* adds data hallucinated with the method by Anastasopoulos and Neubig (2019). +*hall-2k-substr* adds data hallucinated with our method.

strings of syllable-like length for the +*copy-2k-substr* method. The input of this method is the set of bigrams, trigrams and four-grams from the language's training data. For both methods, we generate the dummy lemma by uniformly sampling from the input and concatenating the sampled items to a random length between the minimum and maximum word length we see in the training data. The output of the dummy lemma generation process is a triple of a dummy lemma, a special symbol COPY and the dummy lemma, which is added to the initial "wug test" training set for data augmentation.

+hall-2k-char* and *+hall-2k-substr The dummy lemma generation methods do not leverage knowledge about word structure which can be inferred from the training data. Silfverberg et al. (2017) found that it is very effective to augment training data in low-resource situations with a data hallucination approach by replacing a hypothesized stem of the training triples with a random string. Anastasopoulos and Neubig (2019) improves this data hallucination method by taking into discontinuous stems into consideration as well; this is the best data hallucination method so far. We conduct the +*hall-2k-char* experiment by augmenting the initial "wug test" training set with dummy data generated with Anastasopoulos and Neubig (2019)'s method. The implementation from SIGMORPHON 2020 shared task 0 baseline is used.

In addition, we propose to generate the dummy stem by uniformly sampling from substrings of syllable-like length, i.e. the bigram, trigram and four-gram set. This experiment is referred to as +*hall-2k-substr*. Specifically, both data hallucination methods (illustrated in Figure 2 (b)) take as input a triple from the training set, aligns the lemma and the target form with the alignment method from SIGMORPHON 2016 shared task baseline (Cotterell et al., 2016), finds the common substrings between the lemma and the target form as the stem, replaces the stem with a dummy stem, and outputs a dummy triple which is adopted for data augmentation. Our proposed method is different from Anastasopoulos and Neubig (2019)'s method at the dummy stem generation step in two main aspects: (1) Instead of sampling from the alphabet, we sample from the set of bigrams, trigrams and four-grams. (2) Instead of forcing the dummy stem to be of the same length as the stem to be replaced, we only constrain the minimum and maximum length of the stem based on the training data. In addition, for discontinuous stems, we only replace the first part of the stem.[5]

4 Results and discussion

"Wug test" with data augmentation Figure 3 shows results for the "wug test"-like setting and results after augmenting the initial training set with different methods. Every language sees a substantial improvement with data augmentation, indicating that the Transformer model in the vanilla "wug test" circumstance will not learn a copy bias well.

The substring-based data hallucination we propose, +*hall-2k-substr*, achieves accuracies which are substantially higher than other methods for most languages. For Turkish and Nyanja, +*hall-2k-substr* is lower than the best performance, but the difference is not obvious. For Lingala, +*hall-2k-substr* has the same best performance as +*hall-2k-char*. The consistent advantage of +*hall-2k-substr* implies that substrings of syllable-like length is more helpful than individual characters for data hallucination. It also provides support to the fourth hypothesis we made in section 3 that the poor performance of the Transformer in the vanilla "wug test"-like setting is due to a combination of factors including missing copying bias for letters, subsequences of letters and even entire stems.

Common practice vs "wug test" Figure 1 plots the Transformer accuracies with standard deviations in the common-practice setting, vanilla "wug test"-like setting, and "wug test"-like setting with data augmentation by the substring-based data hallucination methods (+*hall-2k-substr*). Though data augmentation can improve the model's performance for a "wug test", results are still inferior to the common practice setting without any data augmentation for most languages, especially the morphologically challenging 2018 CoNLL-SIGMORPHON languages.

5 Conclusion

In this work, we examine limiting the number of training lemmata and keeping training lemmata disjoint from the evaluation sets in morphological inflection. By comparing the performance of the Transformer under the "wug test"-like circumstance with the common practice, we find that the common-practice setting where the training data covers a fair amount of lemmata and there is overlap of lemmata in training and evaluation, has obscured the difficulty of the task. We propose to augment the training data with substring-based data hallucination, and achieve substantial improvement over previous data hallucination methods.

Considering the findings in this paper, we suggest that future experiments include evaluations on model performance using lemmata not found in the training set and use unique lemma counts rather than triple counts to document data set sizes.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Roee Aharoni, Yoav Goldberg, and Yonatan Belinkov. 2016. Improving sequence to sequence learning for morphological inflection generation: The BIU-MIT systems for the SIGMORPHON 2016 shared task for morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 41–48. Association for Computational Linguistics.

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.

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14(2-3):150–177.

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.

[5] Using the first part only is for implementation simplicity in the current work. It should be adjusted for languages with a large number of discontinuous stems.

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.

Katharina Kann and Hinrich Schütze. 2017. Unlabeled data for morphological generation with character-based sequence-to-sequence models. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 76–81, Copenhagen, Denmark. Association for Computational Linguistics.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, 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).

Ling Liu. 2021. Computational morphology with neural network approaches. *arXiv preprint arXiv:2105.09404*.

Ling Liu and Mans Hulden. 2020a. Analogy models for neural word inflection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2861–2878, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ling Liu and Mans Hulden. 2020b. Leveraging principal parts for morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 153–161, Online. Association for Computational Linguistics.

Ling Liu and Mans Hulden. 2021. Backtranslation in neural morphological inflection. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 81–88, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2018a. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882. Association for Computational Linguistics.

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

Peter Makarov and Simon Clematide. 2018c. UZH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75, 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.

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.

Sarah Moeller, Ling Liu, and Mans Hulden. 2021. To POS tag or not to POS tag: The impact of POS tags on morphological learning in low-resource settings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 966–978, Online. Association for Computational Linguistics.

Sarah Moeller, Ling Liu, Changbing Yang, Katharina Kann, and Mans Hulden. 2020. IGT2P: From interlinear glossed texts to paradigms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5251–5262, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Vancouver. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy. 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.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4425–4438, Brussels, Belgium. Association for Computational Linguistics.

A Data information

	triple-counts			lemma-counts			lemma-overlap (%)	
Language	train	dev	test	train	dev	test	dev-train	test-train
czech	1000	1000	1000	848	848	849	24.53	20.38
finnish	1000	1000	1000	985	983	987	2.34	3.04
german	1000	1000	1000	961	945	962	9.42	9.46
russian	1000	1000	1000	973	985	977	3.65	3.79
spanish	1000	1000	1000	906	902	922	15.74	16.49
turkish	906	928	912	764	802	779	26.06	26.57

Table 1: CoNLL-SIGMORPHON 2018 shared task 1 medium-size data information.

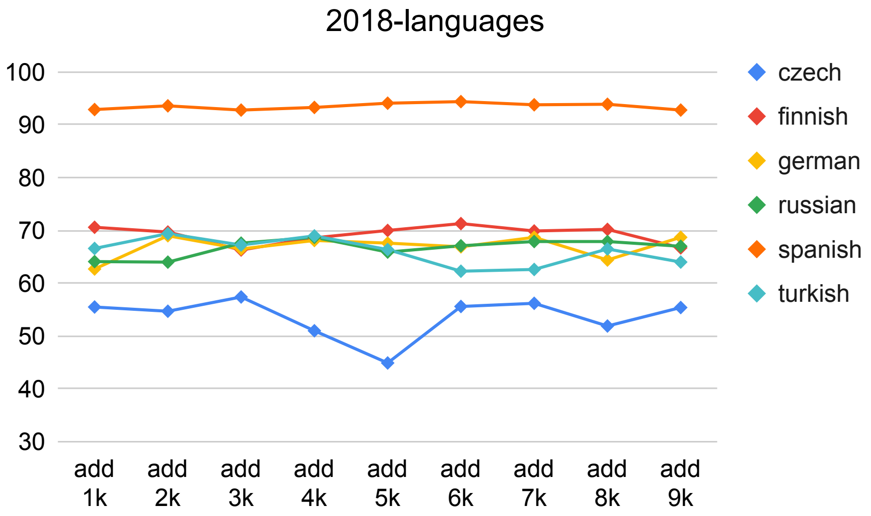
	triple-counts	lemma-counts	lemma-overlap (%)	
Language	train	train	dev-train	test-train
czech	1582	100	0	0
finnish	7136	100	0	0
german	1290	100	0	0
russian	1311	100	0	0
spanish	7132	100	0	0
turkish	7632	100	0	0

Table 2: Data information of the training set we create for 2018-languages. We use the same dev and test sets as CoNLL-SIGMORPHON 2018 shared task 1.

	triple-counts			lemma-counts			lemma-overlap (%)	
Language	train	dev	test	train	dev	test	dev-train	test-train
akan	2793	380	763	96	94	95	100.0	100.0
ga	607	79	169	95	59	80	100.0	100.0
lingala	159	23	46	57	23	34	100.0	100.0
nyanja	3031	429	853	227	199	226	100.0	100.0
southern sotho	345	50	99	26	24	25	100.0	100.0
swahili	3374	469	910	97	97	96	100.0	100.0

Table 3: Data information of Niger-Congo languages from SIGMORPHON 2020 shared task 0.

B Data augmentation size comparison



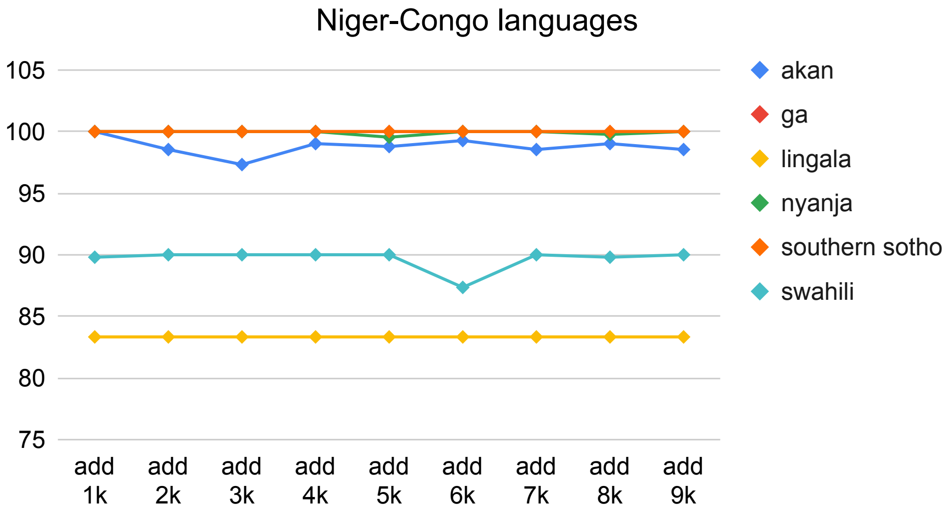


Figure 4: Performance on the dev set in “wug test” after **adding different amounts of dummy data** generated with our substring-based hallucination method.

C Performance of the encoder-decoder with hard monotonic attention model

Considering that the encoder-decoder with hard monotonic attention model (Aharoni et al., 2016; Aharoni and Goldberg, 2017; Makarov et al., 2017; Makarov and Clematide, 2018c,a,b; Wu et al., 2018; Wu and Cotterell, 2019) is designed for the morphological generation task and bias towards copying symbols in the input by leveraging edit actions, we evaluate the performance of the encoder-decoder with exact hard monotonic attention in the "wug test"-like circumstance as well in order to evaluate whether this deep learning model architecture catered to morphological generation is able to learn the generalization ability. We use the encoder-decoder with exact hard monotonic attention model proposed and implemented by Wu and Cotterell (2019).[6]

The performance of the encoder-decoder with exact hard monotonic attention model for the original shared task setup, the "wug test"-like setup with or without our best data hallucination augmentation is presented in Figure 5. Figure 6 provides detailed comparison between different data augmentation methods in the "wug test"-like experimental setup by the encoder-decoder with exact hard monotonic attention model. We observe that the encoder-decoder with exact hard monotonic attention model has the same limitation as the Transformer model pointed out in the previous section.

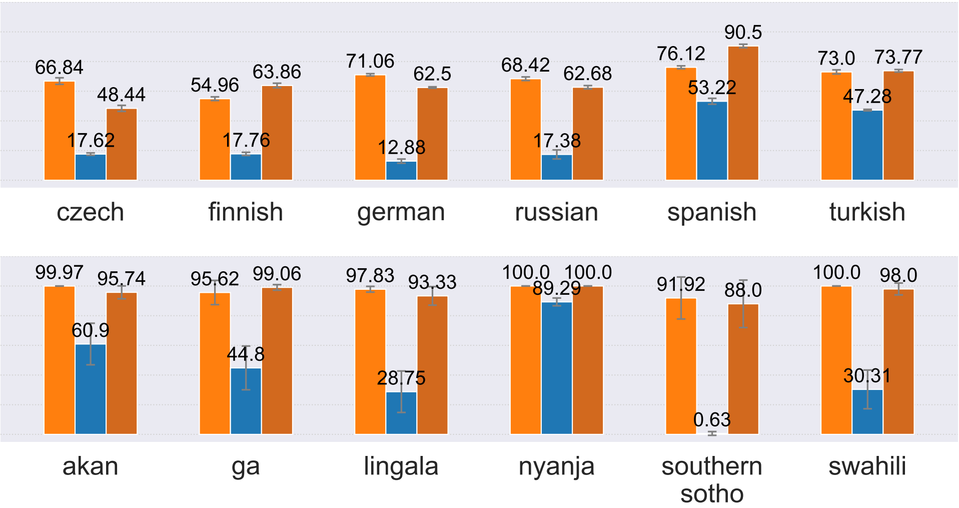
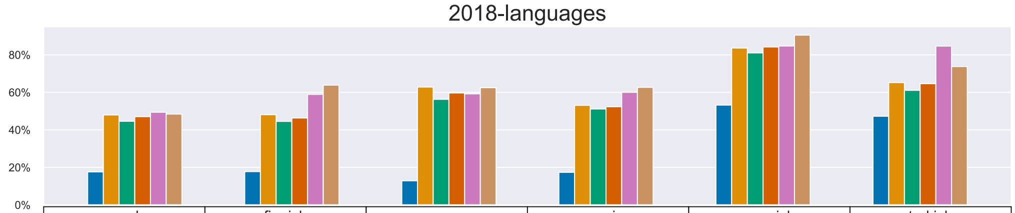
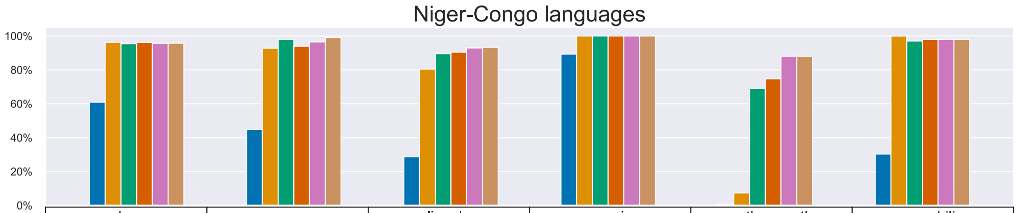


Figure 5: Performance of the encoder-decoder with exact hard monotonic attention model (Wu and Cotterell, 2019) in the **common-practice** setting (left), "**wug test**"-like setting (middle), and **"wug test"-like setting with our best data hallucination** method (right)

[6] https://github.com/shijie-wu/neural-transducer



	czech	finnish	german	russian	spanish	turkish
wug-test	17.62+/-0.79	17.76+/-1.07	12.88+/-1.40	17.38+/-3.05	53.22+/-1.99	47.28+/-0.51
+copy-dev-test-lemmas	47.96+/-2.24	48.10+/-1.37	**62.84+/-1.11**	53.08+/-1.79	83.62+/-1.21	65.22+/-1.40
+copy-2k-char	44.58+/-3.51	44.52+/-0.73	56.28+/-1.53	51.14+/-1.54	81.04+/-1.56	61.08+/-1.64
+copy-2k-substr	47.06+/-2.65	46.34+/-0.92	59.68+/-1.12	52.34+/-1.38	84.20+/-0.24	64.63+/-1.52
+hall-2k-char	**49.38+/-0.79**	58.92+/-0.98	59.20+/-1.31	60.00+/-0.51	84.72+/-0.67	**84.65+/-0.57**
+hall-2k-substr	48.44+/-2.03	**63.86+/-1.54**	62.50+/-0.49	**62.68+/-1.12**	**90.50+/-1.18**	73.77+/-0.85



	akan	ga	lingala	nyanja	southern sotho	swahili
wug-test	60.90+/-14.01	44.80+/-14.74	28.75+/-13.97	89.29+/-2.63	0.63+/-1.26	30.31+/-13.05
+copy-dev-test-lemmas	**96.25+/-2.35**	92.75+/-7.86	80.42+/-19.70	**100.00+/-0.00**	7.37+/-4.36	**99.94+/-0.08**
+copy-2k-char	95.46+/-3.94	98.01+/-2.66	89.58+/-7.45	**100.00+/-0.00**	69.05+/-29.05	97.02+/-4.95
+copy-2k-substr	96.20+/-4.05	93.92+/-3.54	90.42+/-5.83	**100.00+/-0.00**	74.74+/-17.93	97.92+/-3.96
+hall-2k-char	95.66+/-3.95	96.49+/-4.44	92.92+/-6.40	**100.00+/-0.00**	**88.00+/-16.00**	98.00+/-4.00
+hall-2k-substr	95.74+/-4.34	**99.06+/-1.87**	**93.33+/-6.24**	**100.00+/-0.00**	**88.00+/-16.00**	98.00+/-4.00

Figure 6: “Wug test” results by the encoder-decoder with exact hard monotonic attention model (Wu and Cotterell, 2019), with or without different data augmentation methods.