Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3978–3983
Marseille, 11–16 May 2020

Getting More Data for Low-resource Morphological Inflection: Language Models and Data Augmentation

Alexey Sorokin
Moscow State University, Moscow Institute of Physics and Technology
Moscow, Leninskie Gory, GSP-1, Faculty of Mathematics and Mechanics,
name dot surname at list dot ru

Abstract

We investigate the effect of data augmentation on low-resource morphological segmentation. We compare two settings: the pure low-resource one, when only 100 annotated word forms are available, and the augmented one, where we use the original training set and 1000 unlabeled word forms to generate 1000 artificial inflected forms. Evaluating on Sigmorphon 2018 dataset, we observe that using the best among these two models reduces the error rate of state-of-the-art model by 6%, while for our baseline model the error reduction is 17%.

Keywords: inflection, encoder-decoder, abstract paradigms, language models, data augmentation

1. Introduction

Morphological inflection is the process that generates the word form given its lexeme and morphological properties. For example, the inputs *volver* and V;PRS;1;SG produce the inflected form *vuelvo*. This task can be an intermediate stage of text generation, especially in pattern-based approaches. It is also significant by itself for creation and expansion of lexical and morphological resources. Morphological inflection is especially important and challenging for low-resource languages, where no or little annotated data is present.

As any string-to-string task, morphological inflection can be solved by attention-based encoder-decoder architectures(Kann and Schütze, 2016). The effectiveness of different approaches was thoroughly tested during multiple editions of Sigmorphon Shared Task on Morphological Inflection.[1] However, its 2018 edition demonstrated that in high-resource conditions (10000 inflection examples for training) several systems are almost perfect, achieving average accuracy above 96% across 100 languages, whereas for a significant part of these languages the top accuracy exceeds 99%(Cotterell et al., 2018). For medium (1000 word forms) and low (100 word forms) settings the results are also satisfiable, however, a plenty room for improvement remains. Especially hard is the low setting, when some morphological tags may have no training forms. Indeed, in some languages (e.g., Basque) 100 words cannot cover even a single verb paradigm. However, even in such restricted conditions some systems perform significantly better than others, the state-of-the-art approach is imitation learning via minimization of Levenshtein distance between the network output and the correct word form (Makarov and Clematide, 2018b). This model is built on top of the previous systems (Aharoni and Goldberg, 2017; Makarov et al., 2017) that use monotonic attention to generate a sequence of string edits.

We examine two ways of providing more data to the model. The first are language models. There are multiple ways to adapt them to this task: language models can be used to additionally ensure that the generated string satisfies the rules of the language, e.g. phonetic. This approach is inspired by well-known Hidden Markov Models and was applied in Gulcehre et al. (2017) for neural machine translation and in Sorokin (2018) for morphological inflection. In the latter paper the authors conclude that character language models are useless in low-resource setting, though give some advantage in the medium one. However, they trained the language model on the same dataset as the inflection network itself, which led to overfitting. We try to revise Sorokin (2018) conclusion by training the model on additional list of word forms.

The second approach is data augmentation or synthetic data generation. This technique is widely used in many areas of modern computational linguistics, such as grammar error correction (Bryant et al., 2019) or reading comprehension (Yuan et al., 2017). We construct the artificial training examples by the following procedure[2]: first, the nonse lexemes are generated using a character language model; second, the inflection patterns are extracted from training data using abstract paradigms (Ahlberg et al., 2015; Sorokin, 2016); on the third, the most probable pattern for a given lexeme is selected. If the model is not confident enough to select one pattern, the generated word is not added to the data. We evaluate the usefulness of data augmentation for three models: the two baselines from Sorokin (2018) and the state-of-the-art one of Makarov and Clematide (2018b).

Our main result is the following: training on augmented data improves the mean accuracy of Makarov and Clematide (2018b) model from $53,2\%$ to 53.8%. This improvement is rather modest, however, for 40 of 103 languages the augmented model reduces the error rate by more than 5%. If we select the best of two models, the average accuracy goes up to $55,6\%$. Additionally, for a weaker model of (Sorokin, 2018) the effect is much greater: its average accuracy goes up from 42% to 49%.

[1] sigmorphon.github.io

[2] Actually, this is the prediction algorithm of paradigm-based model of Sorokin (2018).

2. Model description

2.1. Baseline model.

Our baseline model is based on Makarov et al. (2017) and is referred as LM-based in Sorokin (2018). Here we describe its basic component, focusing on the differences in the next subsection. The main feature of the model is that it predicts not the word itself, but the sequence of edits from the basic lemma form to the inflected word. For example, the Spanish verb *volver* "to return" and its +1+Sg present form *vuelvo* "*(I) return*" gives birth to the sequence of edits in Figure 1:

The model consists of the bidirectional LSTM encoder and the gated decoder. The decoder also has a pointer observing current encoder state. The gate outputs the weighted sum of two distributions: the first is obtained via usual softmax, while the second outputs the COPY action. Formally,

$$\begin{aligned}
\widehat{z}_i &= \max\left(W_p z_i + b_p, \overline{0}\right), \\
p_i &= \mathrm{softmax}(W_o \widehat{z}_i + b_o), \\
\sigma_i &= \mathrm{sigmoid}(W_\sigma z_i + b_\sigma), \\
\widehat{p}_i &= \sigma_i I(k = c_j) + (1 - \sigma_i) p_i, \\
y_i &= \mathrm{argmax}_k\, \widehat{p}_{ik}.
\end{aligned}$$

When the STEP operation is predicted, the pointer is moved to the next symbol, so the model implements hard monotonic attention mechanism.[3] We refer the reader to the source paper for more details.

2.2. Language model

We use character-level language model in the same way as in Sorokin (2018): the state of the language model is concatenated with the state of the encoder before passing to the decoder. This state is changed when the model predicts a letter (not the STEP move) and is kept unchanged when a STEP is predicted. The model itself is a gated combination of LSTM network and attention over recent symbols, as proposed in Tran et al. (2016). It is trained on the set of word-feature pairs, where features encode the morphological category of the word. Features are encoded via 0/1-vector whose embedding is concatenated to letter embeddings. When no morphological features are present, this vector simply consists of all zeros. Model is trained to optimize perplexity without any supervision, therefore our approach differs from language model pretraining in the sense of Peters et al. (2018) and other related works.

2.3. State-of-the-art model

As the state-of-the-art model we choose the one of Makarov and Clematide (2018b). It also uses a pointer to attend current input symbol, but, in contrast to most other models, applies imitation learning to train the decoder. On each step, the model learns to mimick the decision that minimizes the edit distance between the golden output and the sequence generated so far. It allows the model not to rely on external alignment, but to recover from its own suboptimal decisions. The training algorithm additionally penalizes generation of incorrect letters, as such actions lead to an erroneous form no matter which decisions the model select afterwards.

On inference step, the decoder is simply the recurrent network that conditions on the attended input symbol h_i, the previous output action, a_{t-1} and the global vector f of morphological features (case, gender, etc.). Formally:

$$\begin{aligned}
s_t &= \mathrm{LSTM}(c_{t-1}, [E(a_{t-1}), h_i, f]), \\
P(a_t | a_{<t}, x) &= \mathrm{softmax}(W s_t + b)
\end{aligned}$$

We refer the reader to Makarov and Clematide (2018a) and Makarov and Clematide (2018b) for more details.

2.4. Abstract paradigms

Another model we use as a baseline is based on abstract paradigms (APs). *Abstract paradigms* are patterns used to encode word inflection tables by replacing the components of longest common subsequence (LCS) of the word forms by variables. For examples, the pair *volver-vuelvo*("to return"-"(I) return") is encoded as 1+o+2+er#1+ue+2+o, the same pattern also represents the pair *mover-muevo*("to move"-"(I) move"). Note that instead of generating the inflected form one may predict the abstract paradigm class of the given lemma, thus reducing the inflection task to classification. That is the approach pursued in Ahlberg et al. (2015) and Sorokin (2016). Though abstract paradigms lose information, e.g., about the length of LCS segments and are too rigid to represent all possible inflection patterns, e.g., in languages with complex phonological phenomena, they can "memorize" inflection schemata from small amounts of data and therefore often require less data than neural networks to achieve decent performance.

However, when little data is available, abstract paradigms lack generalization capacity to handle variations, e. g., phonological. With more data, they face the opposite problem – ambiguity. For example, for the +Pres+1+Sg form of Spanish verbs there exist at least the following patterns:

comer	1+er#1+o	*como*
volver	1+o+2+er#1+ue+2+o	*vuelvo*
tener	1+er#1+go	*tengo*
conocer	1+2+er#1+z+2+o	*conozco*
querer	1+2+er#1+i+2+o	*quiero*

Given a previously unseen lemma, for example, *temer*, these patterns produce the following candidate forms[4]:

1+er#1+o	*temo*
1+o+2+er#1+ue+2+o	—
1+er#1+go	*temgo*
1+2+er#1+z+2+o	*tzemo, tezmo*
1+2+er#1+i+2+o	*tiemo, teimo*

As in Sorokin (2016), we rank these candidate forms using language models. That is, we select the form w with the highest value $\log p_{\text{Left}}(w|t) + p_{\text{Right}}(w|t)$. Here $p_{\text{Left}}(w|t)$

[3] This mechanish allows the movement only from left to right, which is adequate for most inflection systems. However, some cases as *hacerse* "to become" $\mapsto$ *se hizo* "I became" are problematic.

[4] One may impose additional restrictions by noting that, e. g., in the pattern 1+2+er#1+z+2+o the second variable is always -*c*-, or that in 1+2+er#1+i+2+o the second variable always starts with -*i*. However, this only reduces the ambiguity, but does not avoid it.

BEGIN	COPY	STEP	u	e	STEP	COPY	STEP	COPY	STEP	o	STEP	STEP	END
BEGIN	v	v	o	o	o	l	l	v	v	e	e	r	END

Figure 1: Transformation of alignment to source-target pair.

Tag	Paradigm
N PSS1S INS PL	1#1+ykuna**wan**
N PSS1S ESS PL	1#1+ykuna**pi**
N PSS1S INS SG	1#1+niy**wan**
N PSS1S INS PL	*1#1+niy**pi***

Table 1: Filling the missing paradigm cells.

and $p_{\text{Right}}(w|t)$ are the probabilities with respect to left-to-right and right-to-left character language models trained on the set of word forms. We discuss this in more details in Section 3.. Sorokin (2018) has shown that this AP-based approach is able to outperform neural baselines at least on a substantial subset of Sigmorphon 2018 languages. Clearly, the performance of AP model is bounded form above by the fraction of abstract paradigms observed in the training data and, consequently, by the fraction of tags in the test data that were also present in the training sample. This may be a serious obstacle for the languages with large inflection tables, no matter how simple these tables are. To deal with issue we take into account the intraparadigmatic interactions: consider the Quechua data in Table 1. Observing the top three forms in it, one may deduce that the essive form is obtained from the instrumentalis form by substituting *-pi-* for *-wan-* in the affix. It yields the pattern 1#1+niy**pi** for the N PSS1S INS PL form. We discuss this procedure more in the Section 3..

3. Data augmentation

The algorithm introduced in 2.4. is too weak to compete with state-of-the-art neural approaches, as the one of Makarov and Clematide (2018b). Therefore we mainly apply it to extend the training datasets by artificial inflection forms. To sample a lemma-tag-word triple, we start with selecting a tag for which at least one pattern is available in the set of abstract paradigms, extracted from the training data. For this tag we generate a lemma, using an ngram model[5], conditioned on its part-of-speech. More precisely, the probability of the symbol c given history h and part-of-speech t is calculated as

$$\begin{aligned} p(c|h,T) &= (\alpha+\beta)p_c(c|h,T)+ \\ &\quad (1-\alpha)p(c|h)+(1-\beta)p(c|h',t), \end{aligned}$$

where h' refers to the history without the first word in h, p_c denotes the probability calculated using raw counts and coefficients α and β are calculated analogously to Witten-Bell smoothing. Given the constructed lemma and tag, we select the most probable word form according to AP inflection model.

[5]With only 100 examples for training, ngram models generally achieve better perplexity, than the neural ones.

Since AP model is imperfect by itself and therefore often produces incorrect output, we cannot rely on all the generated pseudoforms. To filter out the improper inflections we apply two heuristics:

1. Use only the forms whose probability is at least two times higher than the second most probable suggestion.

2. Keep only the words where all letters except at most one have probability higher than a fixed threshold (we set it to 0.001).

4. Models and data

All our experiments are conducted on Sigmorphon 2018 dataset (Cotterell et al., 2018). For every language we use the low subset as the basic training data. We also utilize 1000 word forms from the medium dataset to train the language models used in data augmentation. Namely, the language model for the word forms is trained on 100 word forms from low training tags together with their tags and on 1000 medium word forms *without* tags. Here we mimick[6] the real-world situation where we probably have an external unlabeled data (e. g., Web or Wikipedia). This language model is used to produce 1000 additional artificial lemma-tag-word triples, as described in Section 3.. Consequently, the augmented training dataset consists of 1100 items, 100 original and 1000 generated.

Our main goal is to measure the effect of data augmentation on models of different quality, therefore we evaluate 4 models:

1. The LM-enhanced model of Sorokin (2018) with language model trained on augmented data.

2. The model from (Makarov and Clematide, 2018b), the winner of Sigmorphon 2018 contest.

3. Their versions trained on augmented dataset.

For our sanity checks we also compare the original versions of LM-enhanced and AP models used in (Sorokin, 2018), that use the language models trained on only 100 words from low dataset, with their variants that use language models trained on the extended dataset of 1100 words as described above.

All the models use the parameters from the original publications, so refer to (Makarov and Clematide, 2018b) and (Sorokin, 2018) for training and hyperparameter details.

[6]We acknowledge that the distribution of word forms in Sigmorphon dataset or Unimorph word list significantly differs from the one in real texts.

5. Related work

Morphological inflection and related tasks (lemmatization and transliteration) were addressed studied for quite a long time. Early works include CRF and related architectures (Nicolai et al., 2015) and paradigm-based approaches (Ahlberg et al., 2014; Ahlberg et al., 2015; Sorokin, 2016). The first successful neural model was the one of Kann and Schütze (2016). This model adapted the soft attention mechanism of Bahdanau et al. (2014). Soft attention was replaced by hard monotonic one in (Aharoni and Goldberg, 2017), whose model was further improved by (Makarov et al., 2017).

In Sigmorphon 2018 Shared Task the first place was taken by Makarov and Clematide (2018b) model, that used imitation learning to directly optimize edit distance. This idea was first applied in Makarov and Clematide (2018a). Sharma et al. (2018) applied soft attention over the source word and the sequence of morphological tags. Hard monotonic attention approach, pioneered by Aharoni and Goldberg (2017) and Makarov et al. (2017) was further developed by Wu et al. (2018).

Data augmentation was extensively used, in particular, in 2017 Shared Task (Cotterell et al., 2017). The method of Bergmanis et al. (2017) uses automatically induced patterns and mined corresponding lemma-word pairs from Wikipedia dump. Kann and Schütze (2018) extracted patterns from full inflection tables, filling the missed paradigm cells. Automatic sampling of word stems and paradigm completion were also applied by Silfverberg et al. (2018). However, these approaches work with higher amount of data (1000-2000 inflection pairs) or/and with complete paradigm tables, which is not the case for our study.

Among others, language models were applied to morphological inflection by Sorokin (2016). They were also extensively used in Najafi et al. (2018) system, that took the second place in Sigmorphon 2018 Shared Task. However, they utilized the complete Unimorph data, which is sufficiently more than 1000 word forms used in our work.

Abstract paradigms based on LCS method were introduced by Ahlberg et al. (2014) and further studied in Ahlberg et al. (2015). In Sorokin (2016) their algorithm was extended by additional constraints on prefix and gap length. A recent study (Silfverberg et al., 2018) also uses this notion and investigate the interconnection between different cells inside the paradigm, as well as between different paradigms. However, most of these studies deal with much larger datasets than our work.

6. Results and discussion

6.1. Experiments

Our first experiment is a sanity check: we verify that using a larger dataset to train a language model actually improves performance, while our paradigm completion described in Table 1 does not hamper it. So we compare the models using the LMs trained on 100 words (low setting) with the ones using LMs trained on 1100 words (medium). We also use paradigm completion (PC) in AP model. In Table 2 we present average accuracy across languages, the results for mean edit distance show the same pattern. Consequently, we use the +medium in the following experiments.

Model	low	+medium
LM-based	39,18	40,79
AP-based	42,06	44,20

Table 2: The effect of language model training data size and paradigm completion.

In the second and main experiment we compare the results on the original and augmented dataset. The results are given in Table 3. MERR denotes mean error reduction rate between the basic model and the best of the two and we count only significant improvements with more than 5% of error reduction, while worsenings are taken into account independently from their value. We observe that the weaker baseline model clearly benefits from data augmentation with 15% of average error reduction and significant error reduction on more than 76% of the languages (79 of 103). However, that mostly demonstrates the relative weaknesses of the original model.

What concerns the state-of-the-art model, the results are less convincing. The average improvement in accuracy is rather mediocre and the average error rate reduction is even negative. That is due to 37 languages where adding more data deteriorates performance. However, if we take the best between two models, the average error rate reduction exceeds 6%, giving the gain in average accuracy of $2,4\%$. It shows that the cases of performance worsening are mostly outliers, we discuss the possible causes in Section 7..

In Table 4 we list the languages for which data augmentation produces the largest gains and worsenings in terms of error reduction. All the results are given for (Makarov and Clematide, 2018b) model.

7. Discussion

We observe the controversial effect of data augmentation: for most languages it does help, often rather significantly, while for several languages it has strong negative effect. We suppose that the key reason lies in properties of our augmentation algorithm and its core methods – abstract paradigms and language models. For most of the negative examples their medium dataset includes less than 1000 words (Telugu even lacks such dataset), therefore the language model has little data to base on. It explains the low quality of the produced auxiliary training forms. Another key is the quality of the abstract paradigm model: for Adyghe, Crimean Tatar and Swahili it is comparable with the basic model, so the generated data is of high quality; however, that explanation fails for Turkmen (abstract paradigm outperforms the basic model, but augmentation has strong negative impact) and Quechua (the AP model is much weaker).

Actually, under clear inspection the errors for Turkmen are explained by imbalanced data generation. As most Turkic languages, it possesses vowel harmony, in particular, the suffix agrees in vowel type with the ultimate syllable of the stem. However, the vowels ä and ü that occur in the last stem syllable in the lemmas that lead to an error were never encountered in such a position in the augmented training

Model	Basic	Augmented	Best	# Sign. impr.	# Worsenings	MERR	MERR(best)
(Sorokin, 2018)	42,06	49,58	50,21	79	12	15,11	17,14
(Makarov and Clematide, 2018b)	53,18	53,75	55,59	40	37	-0,02	6,23

Table 3: The effect of data augmentation on baseline and SOTA models.

Language	Basic	Augm.	ERR	AP model
Crimean Tatar	87,0	93,0	46,2	89,0
Uzbek	90,0	94,0	40,0	75,0
Swahili	56,0	72,0	36,4	53,0
Quechua	58,6	72,3	33,1	41,9
Adyghe	90,0	92,9	29,0	90,0
Turkmen	90,0	84,0	-60,0	96,0
Georgian	85,3	73,2	-82,3	69,9
Middle-high-german	84,0	68,0	-100,0	56,0
Telugu	92,0	82,0	-125,0	74,0
Karelian	84,0	58,0	-162,5	44,0

Table 4: Extreme cases of data augmentation effect.

data, which means, that the algorithm has no chance to correctly learn the corresponding phonological pattern. In the case of Karelian the errors are mostly caused by spirious patterns produced by paradigm extraction algorithm.

Let us discuss the spurious matches in more detail. For example, consider the Russian adjective *krasivyj*"beautiful" and its form *krasivoj*"beautiful"+Fem+Gen+Sg. Our algorithm produces the paradigm 1+y+2#1+o+2, however, *j* is the part of the ending and a correct pattern should be 1+yj#1+oj. In case of Russian it is easy to write a rule that word-final components of length 1 are not included, however, it is not so easy for language with more complex morphonology, e.g. vowel or consonant harmony. Another negative consequence of vowel harmony and other phonological alternations is that they divide one general inflection pattern into several surface realizations and our algorithm cannot see the common abstract inflection model.

Summarizing, the discussion, abstract paradigms are a powerful tool for pattern-based morphology, however, they should be refined to work properly in case of phonological alternations and complex intraparadigmatic structure. In our case the auxiliary data that they generate sometimes occur to be incorrect, thus forcing the model in the wrong direction.

8. Conclusion and future work.

We have developed a method of data augmentation for low-resource morphological inflection by the means of abstract paradigms. Our algorithm significantly improves the quality of the SOTA model on 40 languages of 103. Our method comnines neural and paradigmatic approach and can be applied to any neural architecture. We observed that the coverage and quality of augmentation method is crucial for the model performance on augmented data. Clearly, our version of abstract paradigm approach is not readily applicable to all the languages, especially to the ones with complex intraparadigmatic interactions. We expect the methods of Kann and Schütze (2018) and Silfverberg et al. (2018) to be helpful in our case, however, we have not modified them yet to extremely low-resource setting.

Another problem is the exact amount of augmented data to achieve the best performance. This parameter is clearly language- and data-dependent, however, more investigations are required to uncover the factors that affect its value. We leave this question for future research.

9. Acknowledgements

The author thanks the Deep Learning Lab of MIPT that allowed him to conduct this study. He is grateful to Peter Makarov for providing the instuctions how to use his model. He also thanks the anonymous referees whose suggestions significantly helped to improve the paper.

10. Bibliographical References

Aharoni, R. and Goldberg, Y. (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)*, volume 1, pages 2004–2015.

Ahlberg, M., Forsberg, M., and Hulden, M. (2014). Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, April.

Ahlberg, M., Forsberg, M., and Hulden, M. (2015). Paradigm classification in supervised learning of morphology. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT 2015), Denver, CO*, pages 1024–1029, June.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bergmanis, T., Kann, K., Schütze, H., and Goldwater, S. (2017). Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL*

SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, pages 31–39.

Bryant, C., Felice, M., Andersen, Ø. E., and Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Kübler, S., Yarowsky, D., Eisner, J., et al. (2017). CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30.

Cotterell, R., Kirov, C., Sylak-Glassman, Walther, G., McCarthy, A. D., Kann, K., Mielke, S., Nicolai, G., Silfverberg, M., Yarowsky, D., Eisner, J., and Hulden, M. (2018). The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, Brussels, Belgium, October. Association for Computational Linguistics.

Gulcehre, C., Firat, O., Xu, K., Cho, K., and Bengio, Y. (2017). On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148.

Kann, K. and Schütze, H. (2016). Single-model encoder-decoder with explicit morphological representation for reinflection. *arXiv preprint arXiv:1606.00589*.

Kann, K. and Schütze, H. (2018). Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. *arXiv preprint arXiv:1809.08733*.

Makarov, P. and Clematide, S. (2018a). Imitation learning for neural morphological string transduction. *arXiv preprint arXiv:1808.10701*.

Makarov, P. and Clematide, S. (2018b). 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.

Makarov, P., Ruzsics, T., and Clematide, S. (2017). Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57.

Najafi, S., Hauer, B., Riyadh, R. R., Yu, L., and Kondrak, G. (2018). Combining neural and non-neural methods for low-resource morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 116–120.

Nicolai, G., Cherry, C., and Kondrak, G. (2015). Inflection generation as discriminative string transduction. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 922–931.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sharma, A., Katrapati, G., and Sharma, D. M. (2018). IIT (BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 105–111.

Silfverberg, M., Liu, L., and Hulden, M. (2018). A computational model for the linguistic notion of morphological paradigm. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1615–1626.

Sorokin, A. (2016). Using longest common subsequence and character models to predict word forms. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 54–61.

Sorokin, A. (2018). What can we gain from language models for morphological inflection? In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 99–104.

Tran, K., Bisazza, A., and Monz, C. (2016). Recurrent memory networks for language modeling. *arXiv preprint arXiv:1601.01272*.

Wu, S., Shapiro, P., and Cotterell, R. (2018). Hard non-monotonic attention for character-level transduction. *arXiv preprint arXiv:1808.10024*.

Yuan, X., Wang, T., Gulcehre, C., Sordoni, A., Bachman, P., Subramanian, S., Zhang, S., and Trischler, A. (2017). Machine comprehension by text-to-text neural question generation. *arXiv preprint arXiv:1705.02012*.