Noise Isn't Always Negative: Countering Exposure Bias in Sequence-to-Sequence Inflection Models

Garrett Nicolai
Department of Linguistics
University of British Columbia
garrett.nicolai@ubc.ca

Miikka Silfverberg
Department of Linguistics
University of British Columbia
msilfver@ubc.ca

Abstract

Morphological inflection, like many sequence-to-sequence tasks, sees great performance from recurrent neural architectures when data is plentiful, but performance falls off sharply in lower-data settings. We investigate one aspect of neural seq2seq models that we hypothesize contributes to overfitting - teacher forcing. By creating different training and test conditions, exposure bias increases the likelihood that a system too closely models its training data. Experiments show that teacher-forced models struggle to recover when they enter unknown territory. However, a simple modification to the training algorithm to more closely mimic test conditions creates models that are better able to generalize to unseen environments.

1 Introduction

Morphological inflection has gained substantial interest in recent years due to a number of shared tasks focusing on morphology learning (Cotterell et al., 2016; Cotterell et al., 2017; Cotterell et al., 2018; McCarthy et al., 2019; Vylomova et al., 2020; Kann et al., 2020). Systems for inflection are trained on a sequence of pairs: {`Lemma, MorphoSyntactic Descriptor` (MSD)}, free of context, and must produce as output an inflected word form. For example, given the input pair {`run, V.PTCP;PRS`}, a successful system should produce the present participial form: `running`.

The shared tasks have illuminated several idiosyncrasies of inflection. Although the state-of-the-art methods have been inspired by Neural Machine Translation (NMT), inflection is, in many ways, a more straightforward task. Often, a majority of characters can be copied directly from input to output, and reordering of tokens is not necessary to the extent that it is in translation. Thus, systems with copy-mechanisms (Makarov et al., 2017), and hard, monotonic attention (Aharoni and Goldberg, 2017) tend to perform well, even when data is scarce.

Recurrent neural architectures require that the output from a previous time step be fed back into the model. During training, existing systems all use a variant of *teacher forcing*, which feeds gold-standard tokens into the model at time $t + 1$. This methodology differs significantly from how the model progresses at test-time, where silver tokens must be used. This problem has been dubbed *exposure bias* by Wiseman and Rush (2016). We hypothesize that exposure bias can lead models to overfit the training data, particularly in low-resource settings.

In this paper, we compare teacher forcing to silver label propagation, or *student forcing*, where model predictions are fed into the decoder during training instead of gold standard labels. Our experiments on a geographically and linguistically diverse set of 10 languages show that student forcing typically outperforms teacher forcing in a low-resource setting.

Our analysis of the results suggests that teacher forced models are, indeed, overfitting the training data by ignoring input characters at test time. We note substantial gains when student forcing is applied to soft-attentional systems, which bears promise for tasks other than inflection; however, less prominent gains are also observed in conjunction with hard attention.

Proceedings of the 28th International Conference on Computational Linguistics, pages 2837–2846
Barcelona, Spain (Online), December 8-13, 2020

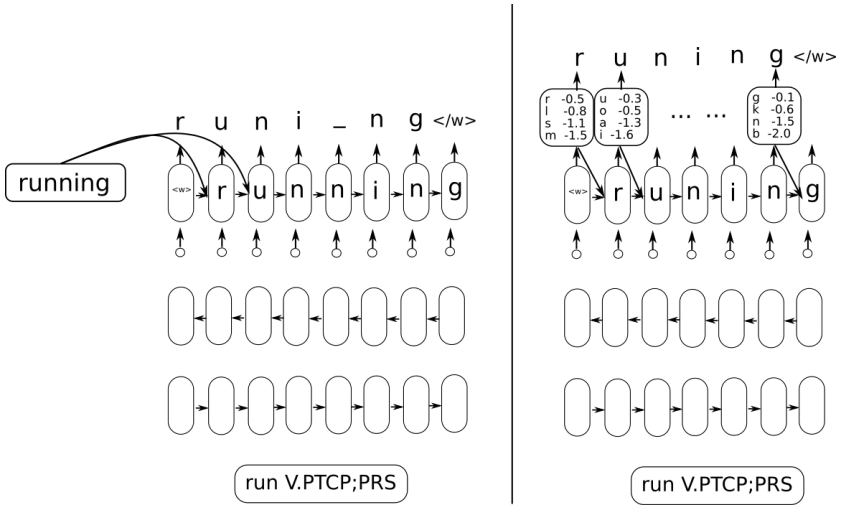

Figure 1: Teacher forcing (left) and Student forcing (right); some connections have been left out to reduce clutter.

2 Methods

We investigate the role that teacher forcing plays on morphological inflection by comparing it against a simple alternative learning strategy, which we call *student forcing*. In our experiments, we apply student forcing to three established neural inflection models: the Pointer-Generator network (**PG**), a soft attention model originally introduced for document summarization (See et al., 2017) and subsequently applied to inflection by Sharma et al. (2018), a neural transducer utilizing hard attention trained on aligned lemmata and inflected word forms (Aharoni and Goldberg, 2017; Makarov et al., 2017) (**HA**) and modified with an explicit copy mechanism, and a later development of the HA system which applies minimum risk training in order to avoid relying on an explicit aligner for the lemma and inflected word form during training (**MRT**) (Makarov and Clematide, 2018a).

2.1 Student forcing

Teacher and student forcing are illustrated in Figure 1. On the left, the teacher-forced system feeds the gold character from time t into the decoder to predict the character at time $t + 1$. The student forcing system, on the right, instead feeds the predicted token from the model. Both systems in this toy example make a mistake in their predictions, but must learn different correction strategies – the teacher-forced system must learn to "get back on track" after an error, while the student-forced system must only be conditioned on its previous prediction, regardless of whether it has made an error or not. While teacher forcing provides a stronger signal, student forcing is more similar to what the system will observe at test time. It can be seen as a special case of a more general approach which employs beam search during training (Wiseman and Rush, 2016).

While the two extremes depicted in Figure 1 force a system to learn using one method or the other, systems typically benefit from a mixture of the two - leveraging the gold information of teacher forcing, while resisting exposure bias through a small amount of student forcing. Therefore, we alternate between teacher and student forcing during training. For each training example, we randomly select either student forcing with probability P or teacher forcing with probability $1 - P$.[1] We treat the probability P as a hyperparameter and tune it on a held-out development set. In our preliminary experiments, we considered alternative strategies, including assigning the decoder strategy at the character level, and schedules that increased or decreased P at set epoch intervals. Our most consistent results were obtained with a word-level constant P, and this is the only method described in the experimental results.

During development, we also experimented with transformer architectures (Vaswani et al., 2017), but found that they performed extremely poorly in the low-resource setting. As low-resource is the focus of

[1] We apply mini-batch training and one mini-batch can contain both teacher forced and student forced training examples.

our attention, we did not continue to explore transformers.

2.2 Student forcing for PG

We modify the pointer-generator network to incorporate student forcing by enhancing the decoder to accept predicted tokens instead of gold-standard ones. If student forcing is assigned to an example, we perform inference alongside training (after disabling gradient accumulation), producing the predicted token for each timestep t. At time $t+1$, the predicted token is added to the decoder input in place of the gold target token.

2.3 Student forcing for HA and MRT

Our implementation of student forcing for the hard-attentional model is similar to that for the pointer-generator: at time t, we generate the predicted edit-action given the current state of the decoder, and subsequently, the predicted token. However, hard attention makes it difficult to completely discount the gold target. The attention relies on alignments generated prior to training, and we do not generate new alignments for the student-forcing. The regeneration of alignments at every time step would result in an unacceptable slowdown of training. While we make use of the gold target for attentional purposes, we replace the input to the decoder with the character predicted at time t, rather than the gold target character.

We make no modifications to the MRT training algorithm from the original system, beyond the modifications made to the decoder.

3 Related Work

To the best of our knowledge, there has been little investigation into the role of teacher forcing for inflection systems. However, Bengio et al. (2015) apply scheduled sampling for RNN's in the context of image captioning, constituency parsing and speech recognition. This approach has been extended to transformers by Mihaylova and Martins (2019). During training, Bengio et al. (2015) randomly sample either the gold standard token y_{t-1} or the model suggestion $\hat{y}_{t-1}$ when predicting y_t. The probability for selecting a model suggestion rather than the gold standard example is continuously increased during training which they call scheduling. Our work can be seen as an application of this approach except that we make the choice between gold standard context and the model suggestion at the example-level instead of the level of individual characters. According to our preliminary experiments, this delivered superior results in the task of inflection. Another difference is that we fix the probability for choosing the model suggestion throughout training and treat this as a hyperparameter. We do this because we did not observe consistent improvements from scheduling.

Goyal et al. (2017) present a modification of the approach by Bengio et al. (2015) for named-entity recognition and machine translation. They replace the argmax model prediction with an average of all output embeddings, weighted by the prediction scores. Also related to scheduled sampling is the approach by Wiseman and Rush (2016) who experiment with beam search. The downside of this approach is that it can substantially increase training times.

4 Data and Experiments

In this section, we lay out our experiment schedule, and describe the system architectures, hyperparameters, and data specifications. We also motivate our choice of evaluation metrics.

4.1 Data

For evaluation, we choose 10 languages from the evaluation data set of the 2018 ConLL-SIGMORPHON Shared Task on Morphological Reinflection (Cotterell et al., 2018). Languages were chosen to represent a wide variety of morphological phenomena. We perform experiments on Arabic, Basque, English, Finnish, German, Navajo, Persian, Sanskrit, Turkish, and Zulu. These ten languages represent a variety of writing systems, including alphabets, abjads (Arabic and Persian), and abugidas (Sanskrit). They demonstrate fusional, agglutinative (Basque, Finnish, Turkish, Zulu) and templatic (Arabic) inflection.

They both suffix and prefix (Navajo, Zulu) extensively. We feel that our language set will allow at least some investigation into the role that linguistic typology plays on exposure bias and the choice of neural inflection models.

We use the standard data splits, consisting of 1000 development and testing instances, and run experiments in three training settings – low, medium, and high, consisting of 100, 1000, and 10,000 training instances, respectively. Systems are evaluated using two metrics: whole word accuracy, and Levenshtein distance, normalized by the length of the inflected form.

Word accuracy has come to be the dominant evaluation metric for inflectional systems, following the standards of the Shared Tasks[2]. However, accuracy does not tell the entire story - we hypothesize that although teacher-forced systems perform very well on instances that resemble training examples, they do very poorly on novel examples. When determining the plural of "dog", accuracy will assign the same score to a system that produces "doggs", and one that produces "elephants". While we do not claim that inflectional models make mistakes of this magnitude, a high edit-distance paired with a high accuracy may suggest significant mistakes are being made in incorrect predictions. Furthermore, student forcing is applicable to other learning tasks where all-or-nothing evaluations are less frequent - BLEU score, for example, has much more in common with edit distance than accuracy.

Our pointer-generator network uses the PyTorch implementation from OpenNMT (Klein et al., 2017). They were trained for 10,000 steps, except for the "high" setting, which was trained for 20,000, upon observation that 10,000 steps was not enough for convergence. Model checkpoints were saved every 500 steps, and the model with the highest development accuracy was used for inference. Under the low setting, the RNN hidden dimension was set to 50, while in the medium and high settings, the RNN size was set to 200. All models were run on 10 different random seeds, using general copy attention. All other system parameters were set to the OpenNMT defaults.

Our HA model was trained using the best training parameters of Makarov and Clematide (2018a), employing their hard-attentional model with a copy mechanism (HACM). We use a single-layer encoder and decoder of 200 hidden units, with a character embedding of size 100 and an action embedding of size 100, no dropout, and a ReLU non-linearity function. The models in the low settings train for a maximum of 60 epochs, with patience of 20 epochs. The medium setting trains for 30 epochs, and the high for 20 - each with patience of 10 epochs. Alignments are obtained using the longest common substring. MRT models sample 20 instances from the distribution.

MRT systems are initialized with the best model obtained via *teacher-forced* HA training, making our teacher-forced results equivalent with those described by Makarov and Clematide (2018b). Student forcing is applied to subsequent MRT training epochs. While it would be possible to initialize the MRT models with student-forced models, we feel that any comparison between these systems would be difficult to make - gains in any metric could be attributed to either the initial model's distribution, or to the MRT with student forcing. We leave further exploration of combining student forcing with MRT to future work.

4.2 Experiments

Development experiments demonstrated that although pure student forcing (ie, P=1.0) occasionally outperformed teacher forcing, the results were inconsistent. Teacher forcing regularly dominated the accuracy metric, and gains in edit distance were small. It appears that the strong signal provided by teacher forcing is necessary to accommodate learning, particularly early in training, when inference is very poor.

We thus experiment tuning the percentage of student forcing used by the model. In our first experiment, we train the pointer-generator network while tuning student forcing from 0% to 50%. The results are plotted in Figure 2.

When we allow the percentage of examples using student-forcing to become a tunable hyperparameter, the advantages of including some noise in the training of the model become clear. In the low setting, we see large gains in accuracy for half of our evaluation languages, and notable decreases in edit distance for nine of them. Likewise, in the medium setting, we see modest improvements in

[2]The tasks also evaluate with edit distance, but it is given much less focus in the discussion

Figure 2: Results of training pointer-generator inflection networks with tuned percentages of student forcing. The languages: Arabic (AR), Basque (EU), English (EN), Finnish (FI), German (DE), Navajo (NV), Farsi (FA), Sanskrit (SA), Turkish (TR) and Zulu (ZU). The first row corresponds to the low setting, the second to the medium setting, and the third to the high setting.

both metrics for all of our languages. In the high setting, there is less improvement – 10,000 training examples proves sufficient to expose the model to enough environments that it is not regularly surprised during inference.

In our next experiment, we consider the same settings as our first experiment, but instead train using the HA model. Figure 3 plots the results. We first note that in the low and medium settings, the HA model is significantly more accurate than the PG model. This result is not overly surprising - when training data is small, hard attention has been shown to help inflection (Aharoni and Goldberg, 2017). However, even with the more accurate models, student forcing improves over the teacher baseline - particularly with respect to edit distance. Furthermore, unlike the pointer-generator, we continue to see small gains even in the high setting.

Finally, we extend the HA models with MRT. Perhaps not surprisingly, we see little gain from combining student forcing and MRT. As described by Makarov and Clematide (2018a), their implementation of minimum risk training is specifically designed to counter exposure bias. Sampling from the model's distribution and injecting noise from the model's predictions are similar strategies for countering exposure bias. If we compare MRT and teacher forcing directly, we find that in the low and medium settings, MRT outperforms student forcing by approximately 1%, on average - but requires twice the training time to get there.

In Figure 4, we plot the average error reduction obtained with student forcing over the teacher-forced models. We see that although the gains for hard-attention are small, they surprisingly increase as training data increases. The gains obtained are complementary to the improvements that can be attributed solely to hard attention. In the next section, we provide more in-depth discussion of the types of errors that are being corrected by student forcing.

5 Discussion

We now turn our attention to the output of our systems, and devote some discussion to certain trends that we observed.

HA vs. PG It is quickly apparent that the Hard Attentional model is superior to the pointer-generator, at least when training data is low - one hundred training examples is simply insufficient to learn an adequate soft attention mechanism. However, once training data is available in sufficient quantities, the pointer-generator approaches, and even, in some circumstances, surpasses the hard attentional model. In particular, the Navajo model struggles with the constraints present with hard attention, but Basque and Persian also benefit from soft attention.

We also note that while the pointer-generator sees the most benefit from student-forcing in the low and medium settings, the hard attentional model steadily improves as data increases. We find this to be a surprising result - by the time a model has observed 10,000 training examples, exposure bias is much less of a concern. It is possible that the soft attention mechanism is performing much of the same type of regularization that we would expect from student forcing - but only becomes strong enough to completely supplant it in the higher data setting.

The HA model, conversely, requires pre-computed alignments, and the quality of these alignments may impact the results significantly. Using these alignments as an approximation of when the model must learn insertion, deletion, or substitution operations, we note that Navajo requires, on average, more than 50% more substitutions than the next closest languages – Arabic, Basque, and Persian. Basque and Persian, in particular perform worse with hard attention in the medium and high settings, while Arabic does so in the high setting. On the other extreme, English, Sanskrit, and German perform few substitutions, and clearly prefer hard attention. Substitutions are much more complex operations than either insertions or deletions, and much harder to generalize. It is possible that hard attention is too strict to fully capture the intricacies of a complex substitutional morphology.

Insertions / Deletions When we look closer at insertions and deletions, particularly in the low setting, we observe an interesting pattern of errors. Analyzing English - a language that benefits immensely from student forcing, we see that teacher-forced models regularly adopt a "clipping" strategy, eliminating

Figure 3: Results of training hard-attentional inflection models with tuned percentages of student forcing. The languages: Arabic (AR), Basque (EU), English (EN), Finnish (FI), German (DE), Navajo (NV), Farsi (FA), Sanskrit (SA), Turkish (TR) and Zulu (ZU). The first row corresponds to the low setting, the second to the medium setting, and the third to the high setting.

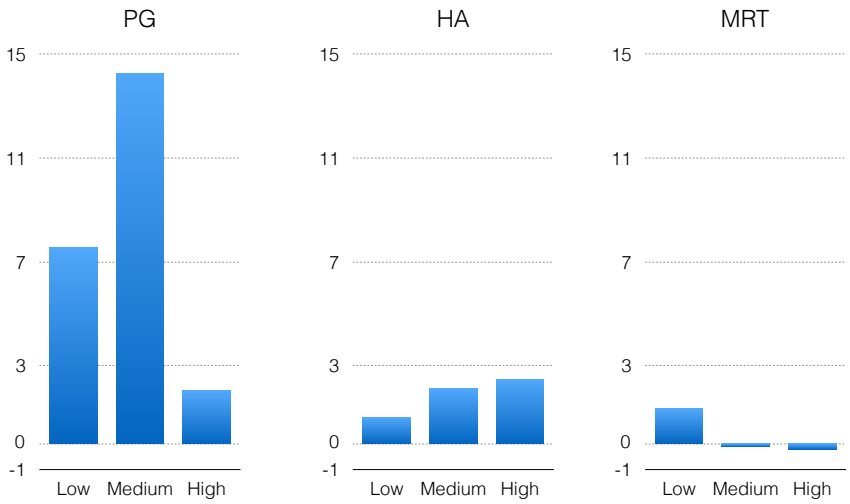

Figure 4: Error reduction rates in percentage under different data settings. PG refers to the Pointer-Generator, HA to hard attention and MRT to minimum risk training

long sequences of characters at the ends of words. For example, the past tense of the English verbs "architect" and "retrospect" are respectively predicted as "*archited", and "*retrosped". The trigram "-ect" is relatively rare in English, and is not observed in training. The teacher-forced model learns that "p" and "t" are viable word-final characters, and that "-ed" is a good suffix for the past tense. "-ed" is much more likely than "-ec", and so the "ect" is clipped from the output. The student-forced model corrects both of these mistakes. Although it still identifies "-ed" as a good suffix, it also learns that copying the input characters is much more likely than deleting, regardless of the previous character.

In an attempt to further classify errors, we calculate the number of prediction errors that are insertions or deletions for each language, as well as the average number of characters that are inserted or deleted with each operation. Furthermore, we calculate whether one or the other is more likely to be corrected by student forcing. We find that, on average, teacher forcing makes more insertion mistakes than deletions (although English is an exception, with 25% more deletions). The student model corrects nearly 37% of insertions, and 27% of deletions. What's more, the student model also significantly reduces the length of errors - reducing the average insertion mistake from 2.5 characters to 1.8, and the average deletion mistake from 2.0 characters to 1.6. By correcting deletion errors, student forcing manages the clipping problem. Correcting spurious insertions, on the other hand, eliminates errors that occur when the model finds itself in an unfamiliar environment, and it generates false characters in an attempt to escape. For example, the pluralization of the German "Nadelbaum" requires umlaut to produce "Nadelbäume", but ä is a relatively rare character, so the model incorrectly produces "m", which drives the model into another unusual environment. Although the model eventually recovers, the predicted output: "*Nadelbmumum-mue" is quite distant from the gold target.

Navajo Many of the languages we have investigated follow a regular pattern of results, particularly in the low setting – the pointer-generator performs poorly with teacher forcing, improves with student forcing, is outpaced by the HA model, which in turn then benefits from student forcing and MRT training. However, across all settings and experiments, there is one outlier - Navajo. Navajo sees little improvement from student forcing, regardless of the training algorithm. Furthermore, its performance lags behind other languages when the data increases. When MRT is applied, ED spikes under teacher forcing – quadrupling in the low setting as exploration reveals a weak distribution. We hypothesize that Navajo has several typological features that make it ill-suited both to student-forcing and hard-attentional low-resource models.

As previously indicated, Navajo has a significant number of stem changes, which may make learning with hard-attention difficult – nearly all of the improvement of the PG over the HA in the high setting can be attributed to Navajo. Secondly, Navajo is a primarily prefixing language. Our implementation of

student forcing progresses from the beginning of the word to the end, and makes some strong assumptions in doing so. Suffixing languages will learn to copy characters at the beginning of the word, an easy task that can be learned quickly. This establishes a strong language model before any true inflection occurs – by the time the first suffix character is produced, the model will have a strong prediction of the previous characters. This is not true of prefixing languages, which have no prior language model to inform prefixes; the information from the teacher serves as a strong regularizer. Zulu is also a strongly prefixing language, but the majority of the inflection in our data set consists of insertion. If insertions are easier to learn than substitutions, then it is reasonable that Zulu will eventually be able to learn a strong inflectional model, as long as it has some access to gold recurrence.

Future work The method described in this paper should be easily portable to other machine learning tasks that use RNNs. Furthermore, our results with the pointer-generator suggest that tasks that are less suited to hard attention may benefit the most from student forcing.

Secondly, we believe that a natural extension of student forcing would leverage the entire distribution, rather than a greedy prediction. Implementing a beam search over predictions, such as Wiseman and Rush (2016) might allow the model to further probe the inflectional language model.

Finally, we believe that more investigation of a wider selection of languages is necessary. While we attempted to investigate a diverse set of languages, the exceptions to the observed trends are too small to make strong conclusions. We would need to expand to other languages that are typologically similar to our exceptional languages like Navajo. In particular, there is very little inflectional literature on tonal languages, which can greatly expand the state space of potential operations.

6 Conclusion

We have presented student forcing, a simple method for countering *exposure bias*. The dissimilarity between how information is presented to the decoder during training and inference can lead models to overfit to the particular environments observed in the training data. By injecting some noise into training, through the modification of the recurrent step of the RNN, we create a training scenario that more closely models what systems will see at test time. We demonstrate that this simple change can lead to moderate improvements in morphological inflection, a simpler version of the general sequence-to-sequence task. Whether measured in accuracy or edit distance, the evidence shows that student forcing injects enough variety into training that it becomes less prone to overfitting. While noise is typically seen as an obstacle to overcome in machine learning, a little bit injected in the right place can encourage learning over memorization.

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, July. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA. MIT Press.

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. 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, August. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels, October. Association for Computational Linguistics.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2017. Differentiable scheduled sampling for credit assignment. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 366–371. Association for Computational Linguistics.

Katharina Kann, Arya D. McCarthy, Garrett Nicolai, and Mans Hulden. 2020. The SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 51–62, Online, July. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. 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.

Peter Makarov and Simon Clematide. 2018b. Neural transition-based string transduction for limited-resource setting in morphology. 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, August. 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, August. Association for Computational Linguistics.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. *CoRR*, abs/1906.07651.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July. Association for Computational Linguistics.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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, July. Association for Computational Linguistics.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas, November. Association for Computational Linguistics.