The IMS–CUBoulder System for the SIGMORPHON 2020 Shared Task on Unsupervised Morphological Paradigm Completion

Manuel Mager
Institute for Natural Language Processing
University of Stuttgart
manuel.mager@ims.uni-stuttgart.de

Katharina Kann
University of Colorado Boulder
katharina.kann@colorado.edu

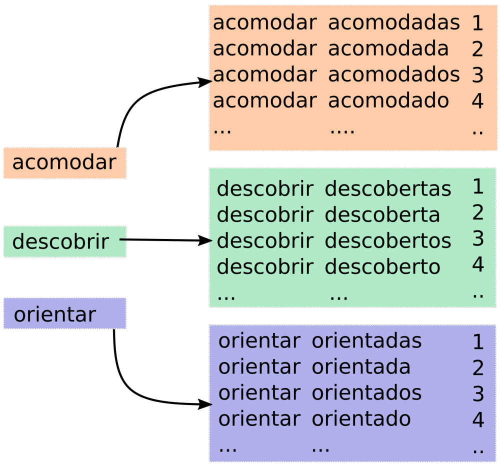
Abstract

In this paper, we present the systems of the University of Stuttgart IMS and the University of Colorado Boulder (IMS–CUBoulder) for SIGMORPHON 2020 Task 2 on unsupervised morphological paradigm completion (Kann et al., 2020). The task consists of generating the morphological paradigms of a set of lemmas, given only the lemmas themselves and unlabeled text. Our proposed system is a modified version of the baseline introduced together with the task. In particular, we experiment with substituting the inflection generation component with an LSTM sequence-to-sequence model and an LSTM pointer-generator network. Our pointer-generator system obtains the best score of all seven submitted systems on average over all languages, and outperforms the official baseline, which was best overall, on Bulgarian and Kannada.

1 Introduction

In recent years, a lot of progress has been made on the task of morphological inflection, which consists of generating an inflected word, given a lemma and a list of morphological features (Kann and Schütze, 2017; Makarov and Clematide, 2018; Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019). The systems developed for this task learn to model inflection in morphologically complex languages in a supervised fashion.

However, not all languages have annotated data available. For the 2018 SIGMORPHON shared task (Cotterell et al., 2018), data for 103 unique languages has been provided. Even this highly multilingual dataset is just covering 1.61% of the 6359 languages[1] that exist in the world (Lewis, 2009). The unsupervised morphological paradigm completion task (Jin et al., 2020) aims at generating inflections – more specifically all inflected forms, i.e., the entire paradigms, of given lemmas – without any explicit morphological information during training. A system that is able to solve this problem can generate morphological resources for most of the world's languages easily. This motivates us to participate in the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion (Kann et al., 2020).


acomodar
descobrir
orientar

acomodar acomodadas 1
acomodar acomodada 2
acomodar acomodados 3
acomodar acomodado 4
... .

descobrir descobertas 1
descobrir descoberta 2
descobrir descobertos 3
descobrir descoberto 4
... .

orientar orientadas 1
orientar orientada 2
orientar orientados 3
orientar orientado 4
... .

Figure 1: Partial Portuguese development examples. The input is a list of lemmas, and the output is a list of all inflected forms of each lemma. In this example, unnamed paradigm slots correspond to the following UniMorph features: 1=V.PTCP;FEM;PL;PST, 2=V.PTCP;FEM;SG;PST, 3=V.PTCP;MASC;PL;PST, 4=V.PTCP;MASC;SG;PST.

The task, however, is challenging: As the number of inflected forms per lemma is unknown a priori, an unsupervised morphological paradigm completion system needs to detect the paradigm size from raw text. Since the names of morphological features expressed in a language are not known if there is no supervision, a system should mark which inflections correspond to the same morphological features across lemmas, but needs to do so without using names, cf. Figure 1. For the shared

[1] The number of languages can vary depending on the classification schema used.

task, no external resources such as pretrained models, annotated data, or even additional monolingual text can be used. The same holds true for multilingual models.

We submit two systems, which are both modifications of the official shared task baseline. The latter is a pipeline system, which performs four steps: edit tree retrieval, additional lemma retrieval, paradigm size discovery, and inflection generation (Jin et al., 2020). We experiment with substituting the original generation component, which is either a simple non-neural system (Cotterell et al., 2017) or a transducer-based hard-attention model (Makarov and Clematide, 2018) with an LSTM encoder-decoder architecture with attention (Bahdanau et al., 2015) – IMS-CUB1– and a pointer-generator network (See et al., 2017) – IMS-CUB2. IMS-CUB2 achieves the best results of all submitted systems, outperforming the second best system by 2.07% macro-averaged best-match accuracy (BMAcc; Jin et al., 2020), when averaged over all languages. However, we underperform the baseline system, which performs 1.03% BMAcc better than IMS-CUB2. Looking at individual languages, IMS-CUB2 obtains the best results overall for Bulgarian and Kannada.

The findings from our work on the shared task are as follows: i) the copy capabilities of a pointer-generator network are useful in this setup; and ii) unsupervised morphological paradigm completion is a challenging task: no submitted system outperforms the baselines.

2 Related Work

Unsupervised methods have shown to be effective for morphological surface segmentation. LINGUISTICA (Goldsmith, 2001) and MORFESSOR (Creutz, 2003; Creutz and Lagus, 2007; Poon et al., 2009) are two unsupervised systems for the task.

In the realm of morphological generation, Yarowsky and Wicentowski (2000) worked on a task which was similar to unsupervised morphological paradigm completion, but required additional knowledge (e.g., a list of morphemes). Dreyer and Eisner (2011) used a set of seed paradigms to train a paradigm completion model. Ahlberg et al. (2015) and Hulden et al. (2014) also relied on information about the paradigms in the language. Erdmann et al. (2020) proposed a system for a task similar to this shared task.

Learning to generate morphological paradigms in a fully supervised way is the more common approach. Methods include Durrett and DeNero (2013), Nicolai et al. (2015), and Kann and Schütze (2018). Supervised morphological inflection has further gained popularity through previous SIGMORPHON and CoNLL–SIGMORPHON shared tasks on the topic (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019). The systems proposed for these shared tasks have a special relevance for our work, as we investigate the performance of morphological inflection components based on Kann and Schütze (2016a,b) and Sharma et al. (2018) within a pipeline for unsupervised morphological paradigm completion.

Language	Training	Development	Test
Basque	85	16	499
Bulgarian	1609	441	2874
English	343	83	302
Finnish	2306	522	1789
German	3940	999	667
Kannada	832	211	2854
Navajo	17	4	279
Spanish	1940	494	2506
Turkish	3095	787	8502

Table 1: Number of instances retrieved by steps 1 to 3 in our pipeline, which are used for training and development of our inflection generation components. The test set contains the lemma and paradigm slot for forms that need to be generated.

3 System Description

In this section, we introduce our pipeline system for unsupervised morphological paradigm completion. First, we describe the baseline system, since we rely on some of its components. Then, we describe our morphological inflection models.

3.1 The Shared Task Baseline

For the initial steps of our pipeline, we employ the first three components of the baseline (Jin et al., 2020), cf. Figure 2, which we describe in this subsection. We use the official implementation.[2]

Retrieval of relevant edit trees. This component (cf. Figure 2.1) identifies words in the monolingual corpus that could belong to a given lemma's paradigm by computing the longest common substring between the lemma and all words. Then, the

[2] https://github.com/cai-lw/morpho-baseline

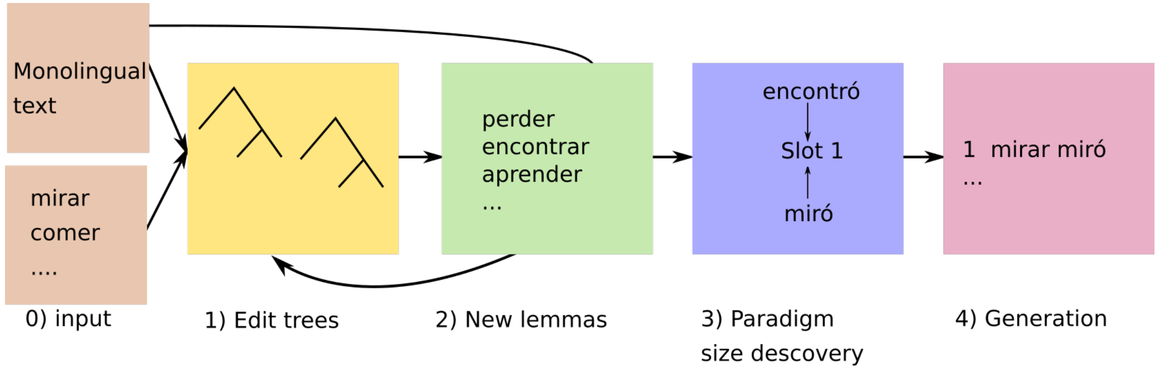


Figure 2: The baseline system. This paper experiments with modifying the generation module. All components are described in §3.1.

transformation from a lemma to each word potentially from its paradigm is represented by edit trees (Chrupała, 2008). Edit trees with frequencies are below a threshold are discarded.

Retrieval of additional lemmas. To increase the confidence that retrieved edit trees represent valid inflections, more lemmas are needed (cf. Figure 2.2). To find those, the second component of the system applies edit trees to potential lemmas in the corpus. If enough potential inflected forms are found in the corpus, a lemma is considered valid.

Paradigm size discovery. Now the system needs to find a mapping between edit trees and paradigms (cf. Figure 2.3). This is done based on two assumptions: that for each lemma a maximum of one edit tree per paradigm slot can be found, and that each edit tree only realizes one paradigm slot for all lemmas. In addition, the similarity of potential slots is measured. With these elements, similar potential slots are merged until the final paradigm size for a language is being determined.

Generation. Now, that the system has a set of lemmas and corresponding potential inflected forms, the baseline employs a morphological inflection component, which learns to generate inflections from lemmas and a slot indicator, and generates missing forms (cf. Figure 2.4). We experiment with substituting this final component.

In the remainder of this paper, we will refer to the original baselines with the non-neural system from Cotterell et al. (2017) and the inflection model from Makarov and Clematide (2018) as BL-1 and BL-2, respectively.

3.2 LSTM Encoder-Decoder

We use an LSTM encoder-decoder model with attention (Bahdanau et al., 2015) for our first system, IMS-CUB1, since it has been shown to obtain high performance on morphological inflection (Kann and Schütze, 2016a). This model takes two inputs: a sequence of characters and a sequence of morphological features. It then generates the sequence of characters of the inflected form. For the input, we simply concatenate the paradigm slot number and all characters.

3.3 Pointer-Generator Network

For IMS-CUB2, we use a pointer-generator network (See et al., 2017).[3] We expect this system to perform better than IMS-CUB1, given the pointer-generator's better performance on morphological inflection in the low-resource setting (Sharma et al., 2018). A pointer-generator network is a hybrid between an attention-based sequence-to-sequence model (Bahdanau et al., 2015) and a pointer network (Vinyals et al., 2015).

The standard pointer-generator network consists of a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) encoder and a unidirectional LSTM decoder with a copy mechanism. Here, we follow (Sharma et al., 2018) and use two separate encoders: one for the lemma and one for the morphological tags. The decoder then computes the probability distribution of the output at each time step as a weighted sum of the probability distribution over the output vocabulary and the attention distribution over the input characters. The weights can be seen as the probability to generate or copy, respectively,

[3] We use the following implementation: https://github.com/abhishek0318/conll-sigmorphon-2018

	IMS-CUB		
Language	1	2-S	2-V
Basque	**25.00**	18.75	12.50
Bulgarian	97.73	**98.19**	97.28
English	96.39	**98.80**	**98.80**
Finnish	**99.04**	98.47	98.85
German	91.49	**93.39**	91.99
Kannada	91.47	**92.89**	91.00
Maltese	79.17	79.17	**85.42**
Navajo	0.00	75.00	**100.00**
Persian	95.56	94.81	**95.56**
Portuguese	93.81	**93.87**	93.74
Russian	92.15	93.02	**93.19**
Spanish	92.91	92.71	**93.52**
Swedish	93.48	**93.69**	93.27
Turkish	93.90	95.30	**95.68**

Table 2: Accuracy of our morphological inflection components on the development sets produced by the first three steps in our pipeline. We list both development and test languages.

and are computed by a feedforward network, given the last decoder hidden state. For details, we refer the reader to Sharma et al. (2018).

4 Experimental Setup

4.1 Data and Languages

The shared task organizers provide data for five development languages, for which development sets with gold solutions are given. Those languages – Maltese, Persian, Portuguese, Russian, Swedish – are not taken into account for the final evaluation.

The test languages, in contrast, are supposed to be only for system evaluation and do not come with developments sets. For those languages – Basque, Bulgarian, English, Finnish, German, Kannada, Navajo, Spanish, and Turkish – only a list of lemmas and a monolingual Bible (McCarthy et al., 2020) are given.

4.2 Evaluation Metric

The official evaluation metric of the shared task is BMAcc (Jin et al., 2020). Gold solutions are obtained from UniMorph (Kirov et al., 2018). Two versions of BMAcc exist: micro-averaged BMAcc and macro-averaged BMAcc. In this paper, we only report macro-averaged BMAcc, the official shared task metric.

During the development of our morphological generation systems, we use regular accuracy, the standard evaluation metric for morphological inflection (Cotterell et al., 2016).

4.3 Morphological Inflection Component

Morphological inflection data. We use the first three components of the baseline model, i.e., the ones performing edit tree retrieval, additional lemma retrieval, and paradigm size discovery, to create training and development data for our inflection models. Those datasets consist of lemma–inflection pairs found in the raw text, together with a number indicating the (predicted) paradigm slot, and are described in Table 1.

The test set for our morphological inflection systems consist of the lemma–paradigm slot pairs not found in the corpus.

Hyperparameters. For IMS-CUB1, we use an embedding size of 300, a hidden layer of size 100, a batch size of 20, Adadelta (Zeiler, 2012) for optimization, and a learning rate of 1. For each language, we train a system for 100 epochs, using early stopping with a patience of 10 epochs.

For IMS-CUB2, we follow two different approaches. The first is to use a single hyperparameter configuration for all languages (IMS-CUB2-S). The second consists of using a variable setup depending on the training set size (IMS-CUB2-V). For IMS-CUB2-S, we use an embedding size of 300, a hidden layer size of 100, a dropout rate of 0.3, and train for 60 epochs with an early-stopping patience of 10 epochs. We further use an Adam (Kingma and Ba, 2014) optimizer with an initial learning rate of 0.001.

For IMS-CUB2-V, we use the following hyperparameters for training set size T:

- $T < 101$: an embedding size of 100, a dropout coefficient of 0.5, 300 epochs of training, and an early-stopping patience of 100;
- $100 < T < 501$: an embedding size of 100, a dropout coefficient of 0.5, 80 training epochs, and an early-stopping patience of 20;
- $500 < T$: the same hyperparameters as for IMS-CUB2-S.

For IMS-CUB2, we select the best performing system (between IMS-CUB2-S and IMS-CUB2-V) as our final model. The models are evaluated on the morphological inflection task development set using accuracy. All scores are shown in Table 2.

	BL		KU-CST		IMS-CUB		NYU-CUB		
Language	1	2	1	2	1	2	1	2	3
Basque	0.06	0.06	0.02	0.01	0.04	00.06	0.05	0.05	**0.07**
Bulgarian	28.30	31.69	2.99	4.15	27.22	**32.11**	27.69	28.94	27.89
English	65.60	**66.20**	3.53	17.29	47.80	61.00	50.20	52.80	51.20
Finnish	05.33	**5.50**	0.39	2.08	04.90	05.38	5.36	5.47	05.35
German	28.35	**29.00**	0.70	4.98	24.60	28.35	27.30	27.35	27.35
Kannada	15.49	15.12	4.27	1.69	10.50	**15.65**	11.10	11.16	11.10
Navajo	3.23	**3.27**	0.13	0.20	0.33	01.17	0.40	0.43	0.43
Spanish	22.96	**23.67**	3.52	10.84	19.50	22.34	20.39	20.56	20.30
Turkish	14.21	**15.53**	0.11	0.71	13.54	14.73	14.88	15.39	15.13
Average	20.39	**21.12**	1.74	04.66	16.49	20.09	17.49	18.02	17.65

Table 3: Final performance (macro-average BMAcc in percentages) of all systems on all test languages. Best scores overall are in bold, and best scores of submitted systems are underlined.

4.4 Results

Table 3 shows the official test set results for IMS-CUB1 and IMS-CUB2, compared to the official baselines and all other submitted systems.

Our best system, IMS-CUB2, achieves the highest scores of all submitted systems (i.e., excluding the baselines), outperforming the second best submission by 2.07% BMAcc. However, BL-1 and BL-2 outperform IMS-CUB2 by 1.03% and 0.3%, respectively. Looking at the results for individual languages, IMS-CUB2 obtains the highest performance overall for Bulgarian (difference to the second best system 0.42%) and Kannada (difference to the second best system 0.53%). Comparing our two submissions, IMS-CUB1 underperforms IMS-CUB2 by 3.6%, showing that vanilla sequence-to-sequence models are not optimally suited for the task. We hypothesize that this could be due to the amount or the diversity of the generated morphological inflection training files.

As our systems rely on the output of the previous 3 steps of the baseline, only few training examples were available for Basque and Navajo: 85 and 17, respectively. Probably at least partially due to this fact, i.e., due to finding patterns in the raw text corpus being difficult, all systems obtain their lowest scores on these two languages. However, even though Finnish has 2306 training instances for morphological inflection, our best system surprisingly only reaches 5.38% BMAcc. The same happens in Kannada and Turkish: the inflection training set is relatively large, but the overall performance on unsupervised morphological paradigm completion is low. On the contrary, even though English has a relatively small training set (343 examples), the performance of IMS-CUB2 is highest for this language, with 66.20% BMAcc. We think that the quality of the generated inflection training set and the correctness of the predicted paradigm size of the languages are the main reasons behind these performance differences. Improving steps 1 to 3 in the overall pipeline thus seems important in order to achieve better results on the task of unsupervised morphological paradigm completion in the future.

5 Conclusion

In this paper, we described the IMS–CUBoulder submission to the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion. We explored two modifications of the official baseline system by substituting its inflection generation component with two alternative models. Thus, our final system performed 4 steps: edit tree retrieval, additional lemma retrieval, paradigm size discovery, and inflection generation. The last component was either an LSTM sequence-to-sequence model with attention (IMS-CUB1) or a pointer-generator network (IMS-CUB2). Although our systems could not outperform the official baselines on average, IMS-CUB2 was the best submitted system. It further obtained the overall highest performance for Bulgarian and Kannada.

Acknowledgments

Thanks to Arya McCarthy, Garrett Nicolai, and Mans Hulden for (co-)organizing this shared task, and to Huiming Jin, Liwei Cai, Chen Xia, and Yihui Peng for providing the baseline system! This project has benefited from financial support to MM by DAAD via a Doctoral Research Grant.

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *NAACL-HLT*, pages 1024–1029, Denver, Colorado. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Grzegorz Chrupała. 2008. *Towards a machine-learning architecture for lexical functional grammar parsing*. Ph.D. thesis, Dublin City University.

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. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Mathias Creutz. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *ACL*, pages 280–287, Sapporo, Japan. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM TSLP*, 4(1):3.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *EMNLP*, pages 616–627, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *NAACL-HLT*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.

Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020. The paradigm discovery problem. *arXiv preprint arXiv:2005.01630*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mans Hulden, Markus Forsberg, and Malin Ahlberg. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *EACL*, pages 569–578, Gothenburg, Sweden. Association for Computational Linguistics.

Huiming Jin, Liwei Cai, Yihui Peng, Chen Xia, Arya D. McCarthy, and Katharina Kann. 2020. Unsupervised morphological paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 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*. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016a. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *14th SIGMORPHON Workshop*, pages 62–70.

Katharina Kann and Hinrich Schütze. 2016b. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

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

Katharina Kann and Hinrich Schütze. 2018. Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3264, Brussels, Belgium. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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).

M Paul Lewis. 2009. *Ethnologue: Languages of the world*. SIL international.

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

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The johns hopkins university bible corpus: 1600+ tongues for typological exploration. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *NAACL-HLT*, pages 922–931, Denver, Colorado. Association for Computational Linguistics.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL-HLT*, pages 209–217. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, pages 1073–1083, Vancouver, Canada. 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, Brussels. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *ACL*, pages 207–216, Hong Kong. Association for Computational Linguistics.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.