THOMAS: The Hegemonic OSU Morphological Analyzer using Seq2seq

Byung-Doh Oh[1]* **Pranav Maneriker**[2]* **Nanjiang Jiang**[1]*
[1]Department of Linguistics, The Ohio State University
[2]Department of Computer Science and Engineering, The Ohio State University
{oh.531, maneriker.1, jiang.1879}@osu.edu

Abstract

This paper describes the OSU submission to the SIGMORPHON 2019 shared task, *Crosslinguality and Context in Morphology*. Our system addresses the *contextual morphological analysis* subtask of Task 2, which is to produce the morphosyntactic description (MSD) of each fully inflected word within a given sentence. We frame this as a sequence generation task and employ a neural encoder-decoder (seq2seq) architecture to generate the sequence of MSD tags given the encoded representation of each token. Follow-up analyses reveal that our system most significantly improves performance on morphologically complex languages whose inflected word forms typically have longer MSD tag sequences. In addition, our system seems to capture the structured correlation between MSD tags, such as that between the verb V tag and TAM-related tags.

1 Introduction

For many natural language processing (NLP) applications such as parsing and machine translation, correctly analyzing the part-of-speech and fine-grained morphological information (e.g. tense, mood, and aspect) of a given string of words is crucial for satisfactory performance. This task depends on the system's ability to learn reliable representations of the sequence on two distinct levels – one at the character-level, which is indicative of the morphosyntactic values of the word, and the other at the word-level, which is informative of subsequent words that are likely to appear in the sequence. In addition, the system needs to have representational flexibility in order to be used in a cross-linguistic setting, as languages with typologically distinct morphological systems (e.g. isolating, agglutinative, and fusional) have different methods of realizing morphological information.

Input	They buy and sell books .
MSD tags	N;NOM;PL | V;SG;1;PRS | CONJ | V;PL;3;PRS | N;PL | PUNCT

Table 1: Example English contextual morphological analysis problem from SIGMORPHON 2019 Shared Task 2 (McCarthy et al., 2019).

Task 2 of the SIGMORPHON 2019 Shared Task, *Morphological Analysis and Lemmatization in Context* (McCarthy et al., 2019), provides an appropriate setting to examine the applicability of morphological analyzers on typologically distinct languages. As mentioned on the shared task webpage,[1] the goal of the *contextual morphological analysis* subtask of Task 2 is to produce the morphosyntactic description (MSD) of each word within a given sentence (i.e. "context," see Table 1 for example).[2] The system's performance is evaluated on a total of 107 treebanks from the UniMorph dataset (McCarthy et al., 2018), which covers more than 70 languages. Again, this requires the system to generalize across typologically different languages without being biased towards a particular morphological system.

In this paper, we present our approach of treating contextual morphological analysis as the generation of the correct sequence of MSD tag dimensions. To address the task, we take a similar approach as the shared task baseline system (Malaviya et al., 2019) in encoding each word in the sequence with a representation learned by a

*First authors. Ordering determined by dice roll.

[1] https://sigmorphon.github.io/sharedtasks/2019/task2/

[2] For the other subtask of *contextual lemmatization*, the goal of which is to return the correct lemmata of the fully inflected forms, we generated the predictions using the pretrained shared task baseline lemmatizer (Malaviya et al., 2019). As the baseline system conducts lemmatization by conditioning on predicted MSD tags, we provided the system with the predictions from our seq2seq model as input.

Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 80–86
Florence, Italy. August 2, 2019 ©2019 Association for Computational Linguistics

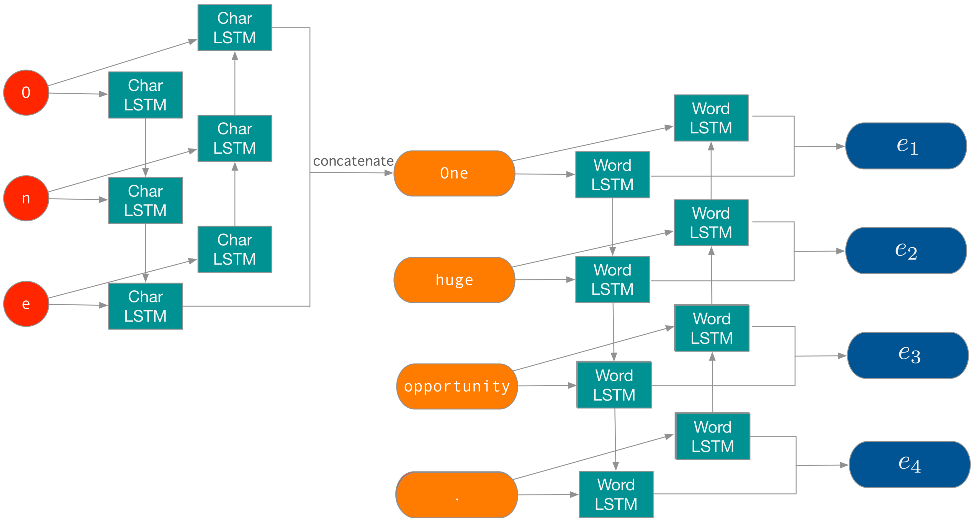
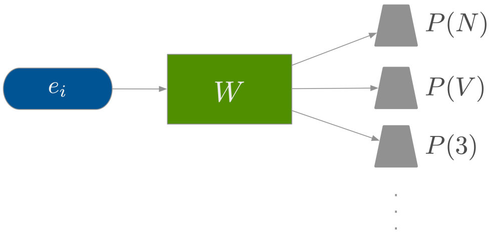
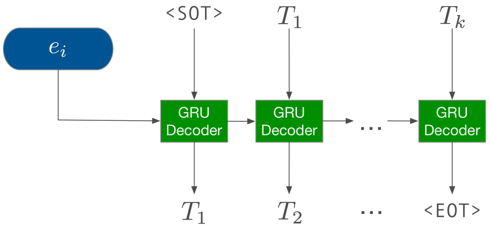


Figure 1: The encoder based on bidirectional LSTM for the baseline, binary relevance, and seq2seq models.



(a) The decoder of the binary relevance model, which makes independent binary decisions for each possible tag dimension.



(b) The GRU decoder of the seq2seq model, which predicts the next tag dimension given the encoder representation and the prediction at the previous timestep.

Figure 2: Overview of the decoder architectures.

character-level recurrent neural network (RNN). With the baseline system that treats each possible combination of MSD tag dimensions separately and chooses the most likely combination, we first demonstrate that modifying the system to make multiple independent binary decisions over each possible tag dimension results in higher performance. Furthermore, we present an encoder-decoder (seq2seq) model that decodes the representation of each input word into a sequence of MSD tag dimensions. The use of the seq2seq model further improves model performance, especially in terms of exact match accuracy for tokens that have long sequences of MSD tag dimensions. Our best-performing model outperforms the official baseline by 14.25 on exact match accuracy and by 4.6 on micro-averaged F1.

2 Model Description

Baseline model The baseline model takes as input each sentence in the training data, and uses a bidirectional LSTM (Long Short-Term Memory, Hochreiter and Schmidhuber, 1997) to learn a representation for each word by attending to its individual characters. The learned representation is then subsequently fed into a fully connected linear layer, which maps the representation of the word to the space of every observed combination of MSD tag dimensions. The network is updated based on the cross-entropy loss between the model's prediction and the correct combination of MSD tag dimensions.

Binary relevance model An obvious limitation to the above baseline approach is that the number of observed combinations of MSD tag dimensions is typically large for most languages, and especially for agglutinative and fusional languages whose words contain relatively more morphological information than those of other languages (see Table 2). In addition, treating each combination separately prevents the model from generalizing to other instances of the same MSD tag dimension that might simply appear in a different combination. We hypothesize that this would most unfavorably impact system performance on

	Sents	Tokens	Tags	Combinations
en	13297	204857	36	178
es	14144	439925	40	419
hi	13317	281948	43	1508
ru	4024	79989	47	1385
tr	4508	46417	55	1896
zh	3997	98734	21	39

Table 2: Descriptive statistics for the six UniMorph treebanks used for training. Number of tags refers to the number of different MSD tag dimensions, and the number of combinations refers to the number of different MSD tag combinations present in each training set.

agglutinative languages, which typically have a clear correspondence between surface string and MSD tag dimension. In order to mitigate this issue, we mapped the learned representation of each word to the space of individual MSD tag dimensions, where independent binary decisions about the presence of each tag dimension are made.

Encoder-decoder (seq2seq) model[3] Nonetheless, given the fact that particular MSD tag dimensions tend to co-occur within a same word (e.g. the "verb" tag dimension frequently co-occurs with tense- or aspect-related tag dimensions), the independence assumption between individual tag dimensions made in the binary relevance model may be too strong to capture this inherent structure. To account for the potential dependence between predicted tag dimensions, we feed the encoded representation of each word as the initial hidden states of a GRU (Gated Recurrent Unit, Cho et al., 2014) decoder, which is then trained to predict one tag dimension at each decoding timestep. The use of such a seq2seq model is also partly motivated by its state-of-the-art performance in various NLP tasks such as machine translation (Bahdanau et al., 2015; Luong et al., 2015), document classification (Nam et al., 2017; Yang et al., 2018), morphological reinflection (Kann and Schütze, 2016; Kann et al., 2017), and morphological analysis like the current shared task (Tkachenko and Sirts, 2018). Our seq2seq model resembles Tkachenko and Sirts's (2018) SEQ model, with the primary difference being the use of a GRU decoder (instead of their unidirectional LSTM) and the sorting of tag dimensions in decreasing order of frequency during training. An overview of our model architecture is presented in Figures 1 and 2.

	Baseline		Bin. Rel.		Seq2seq	
	Acc.	F1	Acc.	F1	Acc.	F1
en	80.17	90.91	92.53	**95.75**	**93.72**	95.41
es	84.35	95.35	96.39	**98.42**	**96.77**	98.31
hi	80.60	93.92	87.59	**96.37**	**88.13**	95.99
ru	63.37	87.49	81.42	**92.92**	**84.92**	**92.92**
tr	62.94	86.10	84.15	**93.87**	**87.08**	93.84
zh	75.97	83.79	89.61	91.18	**91.57**	**91.35**

Table 3: Exact match accuracy and micro-averaged F1 scores of the models evaluated on the test portion of each respective UniMorph treebank. For each dataset, the best results under each metric are in bold.

Our seq2seq model strongly outperforms the official baseline, scoring 14.25 and 4.6 points higher on average across 107 datasets on exact match accuracy and micro-averaged F1 scores respectively. For an in-depth analysis of each model, we focus on 6 languages and compare the performance of our two models (binary relevance and seq2seq) to that of the baseline model.

3 Experimental Design

Training data Following the shared task guidelines, six different treebanks from the UniMorph dataset (McCarthy et al., 2018) provided the data for training and evaluating the model. The six treebanks – English-EWT, Spanish-Ancora, Hindi-HDTB, Russian-GSD, Turkish-IMST, and Chinese-GSD – cover a wide spectrum of morphological typology, thus making it suitable to assess the generalizability of each morphological analysis system. The descriptive statistics of each training set are outlined in Table 2.

Training and evaluation procedure For the binary relevance model, most of the hyperparameters followed the default settings of the baseline system code[4]; characters were embedded into 128-dimension representations, and the character-level biLSTM was trained to output a 256-dimension representation. Adam (Kingma and Ba, 2015) was used as the optimizer, using the default settings of the PyTorch deep learning library (Paszke et al., 2017). The model was trained for five epochs using batches of size 16, with early stopping.[5] The same hyperparameters were used

[3] The predictions from this model were submitted to the shared task. The code repository can be found at https://github.com/njjiang/THOMAS

[4] https://github.com/sigmorphon/contextual-analysis-baseline

[5] As the task organizers do not explicitly mention the hyperparameters used to train the baseline models, it is assumed

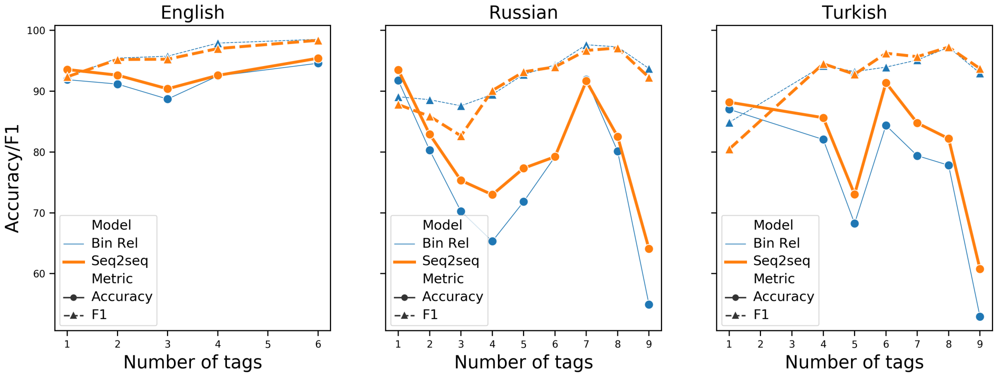


Figure 3: Exact match accuracy and micro-averaged F1 scores of the models on tokens with different numbers of MSD tag dimensions.

	Bin. Rel.	Seq2seq
en	459	89
es	559	70
hi	439	199
ru	423	166
tr	124	63
zh	117	0

Table 4: Number of instances where two tag dimensions that do not co-occur in the test portion of the dataset were predicted together by each model.

to train the encoder portion of the seq2seq model.

As for the GRU decoder, the maximum sequence length was fixed as the maximum sequence length seen during training. Following prior work (Yang et al., 2018), the order of the output tags was fixed to be in decreasing order of frequency of occurrence in the training set. Decoding took place in a greedy manner, and only the highest scoring hypothesis at the previous timestep was further pursued. The model was trained without any teacher forcing, as preliminary results showed that a teacher forcing ratio of 0.5 resulted in a decrease in model performance.

After training was complete, the models' accuracy was evaluated on the held-out test portion of the six treebanks that were used to train the models. As per the shared task guidelines, the exact match accuracy and micro-averaged F1 scores were calculated for each of the trained models.

4 Results and Discussion

As can be seen in Table 3, having the model make independent binary decisions for each possible MSD tag dimension (i.e. the binary relevance model) significantly increases model performance. This is most likely the result of having narrowed down the output space and thereby allowing the model to generalize over instances of the same tag dimension that appear in different combinations. In addition, using a neural decoder to generate a sequence of tag dimensions further improves model performance in terms of exact match accuracy, which is sensitive to predicting the correct number of tag dimensions. This corroborates the results of Tkachenko and Sirts (2018), who found that their sequence generation model outperformed other neural classifiers in terms of accuracy on most languages. The increase in performance is especially salient in Russian and Turkish, which typically have more tag dimensions per word than other languages. An analysis of the distribution of predicted tag dimensions (Table 4) shows that the seq2seq model predicts significantly less "invalid" combinations that are not attested in the gold test set,[6] indicating that the seq2seq model is more capable of capturing the structured dependence compared to the binary relevance model.

Lengths of tag sequences To further examine where the seq2seq model makes significant improvement, the exact match accuracy and micro-averaged F1 scores were calculated according to

that the default settings of the code were used to train them. The only changes to the default settings when training the binary relevance model were in the training epochs (default 10 epochs) and batch size (not implemented, therefore default size 1).

[6] These include combinations of tag dimensions that are either in complementary distribution (e.g. the singular SG and plural PL tags) or linguistically irrelevant (e.g. the noun N tag and tense-related tags).

	Bin. Rel.			**Seq2seq**		
en	**P < G**	**P = G**	**P > G**	**P < G**	**P = G**	**P > G**
0	-	3199	14	-	3201	12
1	0	7819	252	0	7872	199
2	386	9296	164	165	9605	76
3	61	1017	44	58	1037	27
4	125	1995	20	150	1986	4
5	3	384	2	1	386	2
6	27	704	6	20	716	1
ru	**P < G**	**P = G**	**P > G**	**P < G**	**P = G**	**P > G**
0	-	1712	1	-	1712	1
1	0	1751	91	0	1770	72
2	11	165	17	3	176	14
3	21	126	11	4	138	16
4	48	341	29	3	381	34
5	448	3906	235	48	4512	29
6	12	89	0	7	90	4
7	19	386	2	14	389	4
8	11	191	4	4	202	0
9	40	97	5	17	124	1
tr	**P < G**	**P = G**	**P > G**	**P < G**	**P = G**	**P > G**
0	-	1034	0	-	1034	0
1	0	1198	87	0	1183	102
4	175	1382	35	63	1484	45
5	143	477	10	81	538	11
6	28	209	6	9	225	9
7	81	470	26	36	506	35
8	53	257	10	29	279	12
9	24	27	0	18	33	0

Table 5: Comparison of the number of MSD tag dimensions predicted by each model and that in the gold annotation, sorted according to the number of tags in the gold annotation. P refers to the number of tags predicted by the model, and G refers to the number of tags that are in the gold annotation.

		Bin. Rel.		**Seq2seq**	
Tag	Freq.	Acc.	F1	Acc.	F1
COND	18	27.78	86.09	**66.67**	**94.82**
FUT	62	**72.58**	**95.85**	69.35	93.72
HAB	106	75.47	**94.74**	**77.36**	92.67
IMP	32	59.38	77.34	**75.0**	**84.79**
IND	1022	81.12	96.08	**85.71**	**96.51**
OPT	14	71.43	88.4	**85.71**	**95.24**
PFV	944	78.18	94.93	**84.43**	**95.98**
POT	56	**67.86**	**96.91**	62.5	94.19
PROG	134	88.81	98.98	**90.3**	**99.13**
PROSP	3	66.67	95.24	**100.0**	**100.0**
PRS	646	75.54	93.29	**82.82**	**94.47**
PST	439	84.28	98.11	**87.7**	**98.39**
PST+PRF	38	89.47	97.95	**97.37**	**99.26**
FUT/PST	3	66.67	95.24	**100.0**	**100.0**

Table 6: Performance of the two models on TAM-related tokens in the Turkish test set. For each TAM-related tag dimension, the best results under each metric are in bold.

the number of MSD tag dimensions in the test portion of the dataset. In Figure 3, the scores are presented for English, Russian, and Turkish.[7] Additionally, we compared the number of tag dimensions predicted by each model to that of the gold annotation in order to investigate whether there was a tendency for the models to over- or under-predict the correct number of tag dimensions (Table 5). Although there is no clear pattern as to sequences of what length (i.e. short or long) the seq2seq model helps the most, it is clear from the scores that the seq2seq model has the capability to reproduce longer sequences of tag dimensions in comparison to the binary relevance model. Furthermore, while both models predict the correct number of tag dimensions for the vast majority of test examples, the seq2seq model makes more accurate predictions across sequences of nearly all lengths. There is also a general tendency for the two models to under-predict rather than over-predict distinct tag dimensions, with the exception of the seq2seq model on Russian examples with four tag dimensions or less.

Dependence between tag dimensions We hypothesize that the neural decoder of the seq2seq model helped it correctly predict tag dimensions that are low in frequency but often co-occur with a more frequent tag dimension. Such highly dependent examples can be found in the verbal paradigm of a language, where tag dimensions that indicate a particular tense, aspect, and mood (TAM; e.g. present, progressive, indicative) always co-occur with the verb (`V`) tag dimension. We expect that the prediction of the higher-frequency `V` tag dimension during decoding would have helped the model accurately predict these specific TAM-related tag dimensions. As a case study testing this hypothesis, we compared the performance of the two models on TAM-related tokens present in the Turkish test set. The results in Table 6 reveal that the seq2seq model generally outperforms the binary relevance model, indicating that the seq2seq model captures the dependence between the `V` tag dimension and TAM-related tag dimensions.

While the above analyses clearly demonstrate that the seq2seq model learns the structure behind MSD tag dimensions and thus predicts more linguistically plausible sequences in comparison to

[7] There was only one token each with two or three tag dimensions in the test portion of the Turkish dataset (and none in the development portion). As such, the scores for tokens with two or three tag dimensions were omitted in the figure.

Gold	Prediction
INAN;GEN;PL;V;IPFV;PRS;V.PTCP;PASS	INAN;GEN;PL;ADJ
PL;V;FIN;IND;IPFV;PRS;2	SG;INAN;N;FEM;DAT
PL;V;FIN;IND;IPFV;PRS;2	SG;V;FIN;PFV;2;IMP
PL;V;FIN;IND;IPFV;PRS;MID;2	SG;V;FIN;IND;IPFV;3;PRS
PL;V;FIN;IND;PFV;1;FUT	SG;INAN;MASC;N;NOM
PL;V;FIN;IPFV;MID;2;IMP	SG;N;NOM;FEM;V
SG;INAN;FEM;V;ESS;IPFV;PRS;V.PTCP;PASS	SG;INAN;N;NEUT;ESS
SG;INAN;GEN;FEM;V;PST;PFV;V.PTCP;PASS	SG;INAN;GEN;FEM;ADJ
SG;INAN;NOM;V;NEUT;PST;PFV;V.PTCP;PASS	SG;INAN;N;NOM;NEUT
SG;MASC;NOM;ANIM;V;PST;PFV;V.PTCP;PASS	SG;MASC;N;NOM;ANIM
SG;MASC;V;FIN;IND;PST;PFV	SG;MASC;N;NOM;ANIM;PST;PFV;V.PTCP;PASS
SG;V;FIN;IND;IPFV;PRS;1	SG;N;NOM;V;FIN

Table 7: Representative errors from the seq2seq model on Russian test examples with seven or more tag dimensions in the gold annotation.

the binary relevance model, the binary relevance model slightly outperforms the seq2seq model in terms of micro-averaged F1 score. We conjecture that this is due to the nature of the decoder employed in the seq2seq model. Because the decoder conditions on its prediction at the previous timestep, once the decoder predicts an erroneous tag dimension, it is likely to continue to deviate from the correct sequence. This will result in predictions that do not have many tag dimensions in common with the gold annotation. On the other hand, as the binary relevance model is optimized to predict each individual tag dimension independently, it is more likely to generate "partially correct" sequences that are penalized less severely by the F1 score. Representative errors from the seq2seq model on the Russian test set presented in Table 7 demonstrate this tendency; in general, the prediction of an incorrect tag dimension results in predictions that have little overlap with the gold annotation.

In order to alleviate such decoding errors of the seq2seq model, a beam search could be conducted to pursue multiple hypotheses simultaneously. This could help the model recover from an initial erroneous prediction, albeit at the cost of computational efficiency. Furthermore, to explicitly incorporate the underlying structure between MSD tag dimensions, the binary relevance model could be extended to a multiclass multilabel classifier, which selects one tag among those that are in complementary distribution for each morphological category (e.g. part-of-speech, case, number) as in Tkachenko and Sirts (2018). Finally, a more rigorous search for the optimal hyperparameters (e.g. hidden state sizes, training epochs, learning rate) of each model could further enhance their performance. We leave these directions to future work.

5 Conclusion

In this paper, we present our approach to the SIGMORPHON 2019 *contextual morphological analysis* shared task. Expanding from the baseline model that chooses the most likely combination from all those present in the training data, we demonstrate that having the model make independent binary decisions over each tag dimension alleviates data sparsity and improves model performance. Furthermore, based on the linguistic insight that certain tag dimensions often co-occur together, we employed a neural decoder to turn contextual morphological analysis into a sequence generation task and aimed to capture this dependence. This again improved model performance in terms of exact match accuracy, especially for morphologically rich languages that generally have more MSD tag dimensions for every token. A follow-up case study of Turkish verbal inflections demonstrates that the seq2seq model captures the correlation between the more frequent V tag dimension and the less frequent TAM-related tag dimensions.

Acknowledgement

We thank Micha Elsner for all his valuable advice throughout the course of this project.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

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

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, Valencia, Spain. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70, Berlin, Germany. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. *arXiv preprint arXiv:1904.02306v2*.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Crosslinguality and context in morphology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy. Association for Computational Linguistics.

Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2017. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5413–5423. Curran Associates, Inc.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Alexander Tkachenko and Kairit Sirts. 2018. Modeling composite labels for neural morphological tagging. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 368–379, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.