

# Copenhagen at CoNLL–SIGMORPHON 2018: Multilingual Inflection in Context with Explicit Morphosyntactic Decoding

**Yova Kementchedjheva**  
University of Copenhagen  
yova@di.ku.dk

**Johannes Bjerva**  
University of Copenhagen  
bjerva@di.ku.dk

**Isabelle Augenstein**  
University of Copenhagen  
augenstein@di.ku.dk

## Abstract

This paper documents the Team Copenhagen system which placed first in the CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection, Task 2 with an overall accuracy of 49.87. Task 2 focuses on morphological inflection in context: generating an inflected word form, given the lemma of the word and the context it occurs in. Previous SIGMORPHON shared tasks have focused on context-agnostic inflection—the “inflection in context” task was introduced this year. We approach this with an encoder-decoder architecture over character sequences with three core innovations, all contributing to an improvement in performance: (1) a wide context window; (2) a multi-task learning approach with the auxiliary task of MSD prediction; (3) training models in a multilingual fashion.

## 1 Introduction

This paper describes our approach and results for Task 2 of the CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection (Cotterell et al., 2018). The task is to generate an inflected word form given its lemma and the context in which it occurs.

Morphological (re)inflection from context is of particular relevance to the field of computational linguistics: it is compelling to estimate how well a machine-learned system can capture the morphosyntactic properties of a word given its context, and map those properties to the correct surface form for a given lemma.

There are two tracks of Task 2 of CoNLL–SIGMORPHON 2018: in Track 1 the context is given in terms of word forms, lemmas and morphosyntactic descriptions (MSD); in Track 2 only word forms are available. See Table 1 for an example. Task 2 is additionally split in three settings based on data size: high, medium and low, with

high-resource datasets consisting of up to 70K instances per language, and low-resource datasets consisting of only about 1K instances.

The baseline provided by the shared task organisers is a *seq2seq* model with attention (similar to the winning system for reinflection in CoNLL–SIGMORPHON 2016, Kann and Schütze (2016)), which receives information about context through an embedding of the two words immediately adjacent to the target form. We use this baseline implementation as a starting point and achieve the best overall accuracy of 49.87 on Task 2 by introducing three augmentations to the provided baseline system: (1) We use an LSTM to encode the entire available context; (2) We employ a multi-task learning approach with the auxiliary objective of MSD prediction; and (3) We train the auxiliary component in a multilingual fashion, over sets of two to three languages.

In analysing the performance of our system, we found that encoding the full context improves performance considerably for all languages: 11.15 percentage points on average, although it also highly increases the variance in results. Multi-task learning, paired with multilingual training and subsequent monolingual finetuning, scored highest for five out of seven languages, improving accuracy by another 9.86% on average.

## 2 System Description

Our system is a modification of the provided CoNLL–SIGMORPHON 2018 baseline system, so we begin this section with a reiteration of the baseline system architecture, followed by a description of the three augmentations we introduce.

WORD FORMS	We	were	to	feel	very	welcome	.
LEMMAS	we	be	make	to	feel	welcome	.
MSD TAGS	PRO;NOM;PL;1	AUX;IND;PST;FIN		PART	V;NFIN	ADV	ADJ PUNCT

Table 1: Example input sentence. Context MSD tags and lemmas, marked in gray, are only available in Track 1. The cyan square marks the main objective of predicting the word form *made*. The magenta square marks the auxiliary objective of predicting the MSD tag *V;PST;V.PTCP;PASS*.

## 2.1 Baseline

The CoNLL-SIGMORPHON 2018 baseline<sup>1</sup> is described as follows:

The system is an encoder-decoder on character sequences. It takes a lemma as input and generates a word form. The process is conditioned on the context of the lemma [...]. The baseline treats the lemma, word form and MSD of the previous and following word as context in track 1. In track 2, the baseline only considers the word forms of the previous and next word. [...] The baseline system concatenates embeddings for context word forms, lemmas and MSDs into a context vector. The baseline then computes character embeddings for each character in the input lemma. Each of these is concatenated with a copy of the context vector. The resulting sequence of vectors is encoded using an LSTM encoder. Subsequently, an LSTM decoder generates the characters in the output word form using encoder states and an attention mechanism.

To that we add a few details regarding model size and training schedule:

- the number of LSTM layers is one;
- embedding size, LSTM layer size and attention layer size is 100;
- models are trained for 20 epochs;
- on every epoch, training data is subsampled at a rate of 0.3;
- LSTM dropout is applied at a rate 0.3;
- context word forms are randomly dropped at a rate of 0.1;
- the Adam optimiser is used, with a default learning rate of 0.001; and

<sup>1</sup>Code available at: <https://github.com/sigmorphon/conll2018>

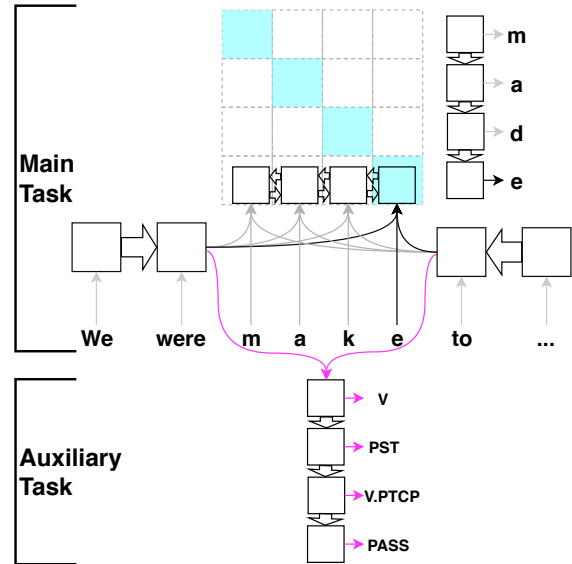


Figure 1: Schematic representation of our approach. The focus here is on the prediction of the final character, *e*, of the word form *made*. The attention matrix indicates that this character should be based on the final state of the encoder, which contains information about the final character of the input form, and the past and future context. The input and output of the auxiliary decoder are marked in magenta.

- trained models are evaluated on the development data (the data for the shared task comes already split in train and dev sets).

## 2.2 Our system

Here we compare and contrast our system<sup>2</sup> to the baseline system. A diagram of our system is shown in Figure 1.

### 2.2.1 Entire Context Encoded with LSTMs

The idea behind this modification is to provide the encoder with access to all morpho-syntactic cues present in the sentence. In contrast to the baseline, which only encodes the immediately adjacent context of a target word, we encode the *entire context*. All context word forms, lemmas, and MSD

<sup>2</sup>Code available at: [https://github.com/YovaKem/inflection\\_in\\_context](https://github.com/YovaKem/inflection_in_context)

		Track 1		Track 2	
		base	our	base	our
high	DE	64.51	<b>72.40</b>	<b>65.72</b>	64.81
	EN	72.91	<b>77.84</b>	70.39	<b>71.90</b>
	ES	53.44	<b>56.24</b>	<b>51.05</b>	48.95
	FI	49.05	<b>55.27</b>	<b>34.82</b>	32.40
	FR	63.54	<b>70.67</b>	58.45	<b>61.51</b>
	RU	71.18	<b>77.91</b>	46.89	<b>49.00</b>
	SV	62.23	<b>69.26</b>	54.04	<b>55.96</b>
med.	DE	54.40	<b>62.18</b>	56.93	<b>57.33</b>
	EN	60.02	<b>66.67</b>	57.60	<b>66.67</b>
	ES	23.14	<b>51.33</b>	41.23	<b>42.50</b>
	FI	28.21	<b>35.71</b>	19.19	<b>22.24</b>
	FR	45.01	<b>60.29</b>	21.38	<b>45.62</b>
	RU	50.30	<b>63.05</b>	30.52	<b>35.94</b>
	SV	47.55	<b>57.66</b>	43.09	<b>45.96</b>
low	DE	0.20	<b>4.85</b>	0.10	<b>18.91</b>
	EN	1.81	<b>33.84</b>	2.22	<b>59.42</b>
	ES	8.98	<b>31.42</b>	8.98	<b>31.84</b>
	FI	0.76	<b>12.83</b>	0.38	<b>12.33</b>
	FR	0.00	<b>34.42</b>	0.00	<b>29.53</b>
	RU	0.00	<b>25.90</b>	2.71	<b>22.69</b>
	SV	1.17	<b>27.55</b>	0.96	<b>30.96</b>

Table 2: Official shared task test set results.

tags (in Track 1) are embedded in their respective high-dimensional spaces as before, and their embeddings are concatenated. However, we now reduce the entire past context to a fixed-size vector by encoding it with a forward LSTM, and we similarly represent the future context by encoding it with a backwards LSTM.

### 2.2.2 Auxiliary Task: MSD of the Target Form

We introduce an auxiliary objective that is meant to *increase the morpho-syntactic awareness of the encoder* and to regularise the learning process—the task is to predict the MSD tag of the target form. MSD tag predictions are conditioned on the context encoding, as described in 2.2.1. Tags are generated with an LSTM one component at a time, e.g. the tag *PRO;NOM;SG;I* is predicted as a sequence of four components,  $\langle \text{PRO}, \text{NOM}, \text{SG}, \text{I} \rangle$ .

For every training instance, we backpropagate the sum of the main loss and the auxiliary loss without any weighting.

As MSD tags are only available in Track 1, this augmentation only applies to this track.

### 2.2.3 Multilinguality

The parameters of the entire MSD (auxiliary-task) decoder are shared across languages.

Since a grouping of the languages based on language family would have left several languages in single-member groups (e.g. Russian is the sole representative of the Slavic family), we experiment with random *groupings of two to three languages*. Multilingual training is performed by randomly alternating between languages for every new minibatch. We do not pass any information to the auxiliary decoder as to the source language of the signal it is receiving, as we assume abstract morpho-syntactic features are shared across languages.

**Finetuning** After 20 epochs of multilingual training, we perform 5 epochs of monolingual finetuning for each language. For this phase, we reduce the learning rate to a tenth of the original learning rate, i.e. 0.0001, to ensure that the models are indeed being finetuned rather than retrained.

### 2.2.4 Model Size and Training Schedule

We keep all hyperparameters the same as in the baseline. Training data is split 90:10 for training and validation. We train our models for 50 epochs, adding early stopping with a tolerance of five epochs of no improvement in the validation loss. We do not subsample from the training data.

### 2.2.5 Ensemble Prediction

We train models for 50 different random combinations of two to three languages in Track 1, and 50 monolingual models for each language in Track 2. Instead of picking the single model that performs best on the development set and thus risking to select a model that highly overfits that data, we use an ensemble of the five best models, and make the final prediction for a given target form with a majority vote over the five predictions.

## 3 Results and Discussion

Test results are listed in Table 2. Our system outperforms the baseline for all settings and languages in Track 1 and for almost all in Track 2—only in the high resource setting is our system not definitively superior to the baseline.

Interestingly, our results in the low resource setting are often higher for Track 2 than for Track 1, even though contextual information is less explicit

in the Track 2 data and the multilingual multi-tasking approach does not apply to this track. We interpret this finding as an indicator that a simpler model with fewer parameters works better in a setting of limited training data. Nevertheless, we focus on the low resource setting in the analysis below due to time limitations. As our Track 1 results are still substantially higher than the baseline results, we consider this analysis valid and insightful.

### 3.1 Ablation Study

We analyse the incremental effect of the different features in our system, focusing on the low-resource setting in Track 1 and using development data.

**Entire Context Encoded with LSTMs** Encoding the entire context with an LSTM highly increases the variance of the observed results. So we trained fifty models for each language and each architecture. Figure 2 visualises the means and standard deviations over the trained models. In addition, we visualise the average accuracy for the five best models for each language and architecture, as these are the models we use in the final ensemble prediction. Below we refer to these numbers only.

The results indicate that encoding the full context with an LSTM highly enhances the performance of the model, by 11.15% on average. This observation explains the high results we obtain also for Track 2.

#### Auxiliary Task: MSD of the Target Form

Adding the auxiliary objective of MSD prediction has a variable effect: for four languages (DE, EN, ES, and SV) the effect is positive, while for the rest it is negative. We consider this to be an issue of insufficient data for the training of the auxiliary component in the low resource setting we are working with.

**Multilinguality** We indeed see results improving drastically with the introduction of multilingual training, with multilingual results being 7.96% higher than monolingual ones on average.

We studied the five best models for each language as emerging from the multilingual training (listed in Table 3) and found no strong linguistic patterns. The EN–SV pairing seems to yield good models for these languages, which could be explained in terms of their common language family

DE	FI	SV	FI, SV	RU, FR	FR, FI
EN	RU, SV	RU, FI	RU, FR	SV, ES	SV, FR
ES	DE	FI	SV, DE	SV, EN	SV, FR
FI	DE	ES	FR, ES	EN, RU	RU, SV
FR	SV, EN	EN, ES	DE, FI	SV, EN	EN, SV
RU	SV	DE, FR	EN, SV	SV, FR	EN, FI
SV	EN, DE	FI, EN	FR, RU	ES, EN	RU, EN

Table 3: Five best multilingual models for each language.

and similar morphology. The other natural pairings, however, FR–ES, and DE–SV, are not so frequent among the best models for these pairs of languages.

Finally, monolingual finetuning improves accuracy across the board, as one would expect, by 2.72% on average.

**Overall** The final observation to be made based on this breakdown of results is that the multi-tasking approach paired with multilingual training and subsequent monolingual finetuning outperforms the other architectures for five out of seven languages: DE, EN, FR, RU and SV. For the other two languages in the dataset, ES and FI, the difference between this approach and the approach that emerged as best for them is less than 1%. The overall improvement of the multilingual multi-tasking approach over the baseline is 18.30%.

### 3.2 Error analysis

Here we study the errors produced by our system on the English test set to better understand the remaining shortcomings of the approach. A small portion of the wrong predictions point to an incorrect interpretation of the morpho-syntactic conditioning of the context, e.g. the system predicted *plan* instead of *plans* in the context *Our \_ include raising private capital*. The majority of wrong predictions, however, are nonsensical, like *bomb* for *job*, *fify* for *fixing*, and *gnderrate* for *understand*. This observation suggests that generally the system did not learn to copy the characters of lemma into inflected form, which is all it needs to do in a large number of cases. This issue could be alleviated with simple data augmentation techniques that encourage autoencoding (see, e.g., Bergmanis et al., 2017).

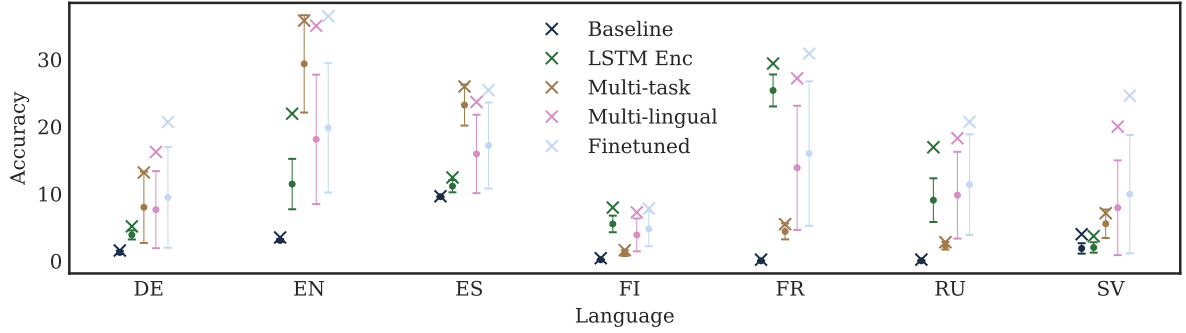


Figure 2: Mean (●) and standard deviation (error bars) over 100 models trained for each language and architecture, and average (×) over the 5 best models. *LSTM Enc* refers to a model that encodes the full context with an LSTM; *Multi-task* builds on *LSTM Enc* with an auxiliary objective of MSD prediction; *Multilingual* refers to a model with an auxiliary component trained in a multilingual fashion; *Finetuned* refers to a multilingual model topped with monolingual finetuning.

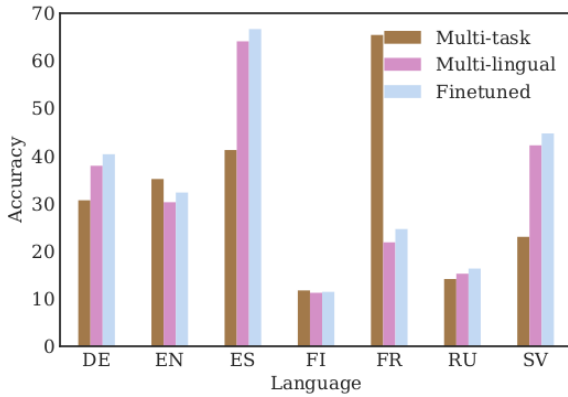


Figure 3: Accuracy on the auxiliary task of MSD prediction with different models. See the caption of Figure 2 for more details.

### 3.3 MSD prediction

Figure 3 summarises the average MSD-prediction accuracy for the multi-tasking experiments discussed above.<sup>3</sup> Accuracy here is generally higher than on the main task, with the multilingual finetuned setup for Spanish and the monolingual setup for French scoring best: 66.59% and 65.35%, respectively. This observation illustrates the added difficulty of generating the correct surface form even when the morphosyntactic description has been identified correctly.

We observe some correlation between these numbers and accuracy on the main task: for DE, EN, RU and SV, the brown, pink and blue bars here pattern in the same way as the corresponding ×’s in Figure 2. One notable exception to this

<sup>3</sup>As MSD tags are not available for target forms in the development data, the accuracy of MSD prediction is measured over all other nouns, adjectives and verbs in the dataset.

pattern is FR where inflection gains a lot from multilingual training, while MSD prediction suffers greatly. Notice that the magnitude of change is not always the same, however, even when the general direction matches: for RU, for example, multilingual training benefits inflection much more than in benefits MSD prediction, even though the MSD decoder is the only component that is actually shared between languages. This observation illustrates the two-fold effect of multi-task training: an auxiliary task can either inform the main task through the parameters the two tasks share, or it can help the main task learning through its regularising effect.

## 4 Related Work

Our system is inspired by previous work on multi-task learning and multi-lingual learning, mainly building on two intuitions: (1) jointly learning related tasks tends to be beneficial (Caruana, 1997; Søgaard and Goldberg, 2016; Plank et al., 2016; Bjerva et al., 2016; Bjerva, 2017b); and (2) jointly learning related languages in an MTL-inspired framework tends to be beneficial (Bjerva, 2017a; Johnson et al., 2017; de Lhoneux et al., 2018). In the context of computational morphology, multilingual approaches have previously been employed for morphological reinflection (Bergmanis et al., 2017) and for paradigm completion (Kann et al., 2017). In both of these cases, however, the available datasets covered more languages, 40 and 21, respectively, which allowed for linguistically-motivated language groupings and for parameter sharing directly on the level of characters. De Lhoneux et al. (2018) explore param-



eter sharing between related languages for dependency parsing, and find that sharing is more beneficial in the case of closely related languages.

## 5 Conclusions

In this paper we described our system for the CoNLL–SIGMORPHON 2018 shared task on Universal Morphological Reinflection, Task 2, which achieved the best performance out of all systems submitted, an overall accuracy of 49.87. We showed in an ablation study that this is due to three core innovations, which extend a character-based encoder-decoder model: (1) a wide context window, encoding the entire available context; (2) multi-task learning with the auxiliary task of MSD prediction, which acts as a regulariser; (3) a multilingual approach, exploiting information across languages. In future work we aim to gain better understanding of the increase in variance of the results introduced by each of our modifications and the reasons for the varying effect of multi-task learning for different languages.

## Acknowledgements

We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

- Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39.
- Johannes Bjerva. 2017a. *One Model to Rule them all – Multitask and Multilingual Modelling for Lexical Analysis*. Ph.D. thesis, University of Groningen.
- Johannes Bjerva. 2017b. Will my auxiliary tagging task help? Estimating Auxiliary Tasks Effectivity in Multi-Task Learning. In *NoDaLiDa*, pages 216–220.
- Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *COLING*, pages 3531–3541.
- Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28 (1):41–75.
- 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*, Brussels, Belgium. Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. One-shot neural cross-lingual transfer for paradigm completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1993–2003. 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.
- Miryam de Lhoneux, Johannes Bjerva, Isabelle Augenstein, and Anders Søgaard. 2018. Parameter sharing between dependency parsers for related languages. In *Proceedings of EMNLP*.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In *Proceedings of ACL (Short Papers)*.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of ACL (Short Papers)*.