CUNI Systems in WMT21: Revisiting Backtranslation Techniques for English-Czech NMT

Petr Gebauer and **Ondřej Bojar** and **Vojtěch Švandelík** and **Martin Popel**
Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics,
Prague, Czechia
pgebauer27@seznam.cz bojar@ufal.mff.cuni.cz
vsvandelik@matfyz.cz popel@ufal.mff.cuni.cz

Abstract

We describe our two NMT systems submitted to the WMT2021 shared task in English-Czech news translation: CUNI-DocTransformer (document-level CUBBITT) and CUNI-Marian-Baselines. We improve the former with a better sentence-segmentation pre-processing and a post-processing for fixing errors in numbers and units. We use the latter for experiments with various backtranslation techniques.

1 Introduction

In this paper, we describe our two NMT systems submitted to the WMT 2021 English-Czech news translation shared task: "CUNI-DocTransformer" (Charles University document-level Transformer) and "CUNI-Marian-Baselines". In addition, we submitted also "CUNI-Transformer2018", which is exactly the same system (sentence-level) as submitted in 2018 (Popel, 2018).

CUNI-DocTransformer uses the same model as submitted last year (Popel, 2020), but with improved sentence segmentation (Section 3.1) and number-unit postprocessing (Section 3.2). This system was submitted for both English→Czech and Czech→English.

CUNI-Marian-Baselines is an attempt at reimplementation of the original CUNI-Transformer2018 in Marian (Junczys-Dowmunt et al., 2018), where we experiment with various setups of tagged backtranslation (Section 4). This system was trained only for English→Czech.

According to automatic evaluation provided by the WMT organizers (Table 1), CUNI-DocTransformer is the third best English→Czech system.

	cased BLEU		chrF
system	**ref A**	**ref B**	**ref A**
Facebook-AI	**24.80**	**(1) 22.69**	**(1) 0.5358**
Online-W	23.02	(2) 21.57	(2) 0.5285
CUNI-DocTransformer	22.19	(3) 19.85	(3) 0.5170
CUNI-Transformer2018	21.63	(4) 19.67	(4) 0.5091
eTranslation	21.03	(5) 19.38	(5) 0.5063
Online-A	20.16	(7) 18.18	(7) 0.4989
CUNI-Marian-Baselines	20.09	(6) 18.29	(6) 0.4992
Online-B	20.04	(8) 17.90	(8) 0.4956
Online-Y	18.13	(9) 16.13	(9) 0.4807
Online-G	15.30	(10) 13.87	(10) 0.4570

Table 1: Evaluation of English→Czech WMT21 systems. The systems are ordered by BLEU with reference A, ordering by the other metrics is provided in parentheses. Names of systems described in this paper are in bold.

2 Common settings

Both our systems use the Transformer (Vaswani et al., 2017) architecture, checkpoint averaging (using the last 8 checkpoints) and a 32k joint English-Czech subword vocabulary. Both systems are trained on CzEng 2.0 (Kocmi et al., 2020) with 61M authentic parallel and 127M synthetic (back-translated) sentences (see Table 2), but the English→Czech CUNI-DocTransformer does not use directly the EN-mono section,[1] while CUNI-Marian-Baselines uses all three sections including EN-mono (i.e. using forward-translation).

Both systems use Block-backtranslation (Popel et al., 2020), although CUNI-Marian-Baselines uses too small block size, so it does not have the expected positive effect as described in Section 4.

3 DocTransformer improvements

3.1 Sentence segmentation

CUNI-DocTransformer was trained on multi-sentence sequences of up to 3000 characters and

[1]The synthetic data in CzEng 2.0 were prepared using iterated backtranslation, so the EN-mono data were used for training a Czech→English system, which produced the English translation of the CS-mono data in CzEng 2.0. Thus, indirectly also the EN-mono data were used for training the English→Czech CUNI-DocTransformer.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 128–134
Novermber 10–11, 2021.

data set	sentence pairs (M)	words (M) EN	CS
authentic	61	617	702
EN-mono (NewsCrawl 2016–2018)	76	1296	1474
CS-mono (NewsCrawl 2013–2018)	51	700	833
total	188	2613	3009

Table 2: Training data sizes (in millions). All the data are taken from CzEng 2.0.

750 subwords. However, the WMT submission format requires a segment-level alignment and also the CUNI-DocTransformer decoding employs overlapping sequences where sentence alignment is needed (for details see Popel (2020)). Thus, the sentences within a sequence are separated with a special token on both source and target side (both during training and at inference time), which allows a simple extraction of the sentence alignment.[2]

Some segments in the WMT input format contain multiple sentences. When treating such segments as a single sentence, the resulting translations often missed sentence-initial capital letters because there were almost no such examples in the training data, where multiple sentences would not be separated by the special token.

We thus decided to first split the input segments into sentences using UDPipe (Straka et al., 2016). Unfortunately, UDPipe tends to over-segment.[3] Such over-segmentation may lead to serious errors in the translation, even when using the document-level model. We thus restrict the sentences boundaries detected by UDPipe only to boundaries after sentence-final punctuation, using a simple rule-based segmenter from Udapi (Popel et al., 2017). This improved BLEU on our dev set slightly.

3.2 Number-unit post-processing

We noticed three types of translation errors related to numbers and units.

1. Attempt at converting numbers and units. For example, the Czech sentence *Je vysoký pouhých 190 cm* (meaning *He's only 190 cm tall*) was translated as *He's only six feet tall*. Note that six feet is 183 cm, so the translation was not exact.

2. Converting units without numbers. For example, *27 Kč* was translated as *$27*, while the correct translation should be *27 crowns* or *27 CZK*.

3. Not converting separators. English uses commas (or thin spaces) as thousand separators and dots as decimal separators, but Czech uses the opposite convention (with space being a more common thousand separator than dot). So e.g. Czech *179,500 kg* means 179 and a half kg (with precision up to 1 gram) and the correct translation to English should be *179.500 kg*, but CUNI-DocTransformer (and many other systems) keeps the phrase untranslated, resulting in a thousand times higher value.

The first type is quite rare – 0.7% of numerical expressions with units in cs-en and 0.6 in en-cs, according to Table 3, while some of these cases may be correct translation (correctly converted number and unit). The second type is more frequent – 11.1% and 6.5%, respectively. The third type is also frequent – in 100,000 Czech sentences from CzEng 2.0 cs-mono, there were 2594 numbers with a separator and out of these 275 (10.6%) were not correctly converted in the English CUBBITT translations; similarly in 100,000 English sentences in en-mono, there were 4376 numbers with a separator and out of these 263 (6.0%) were not converted in the Czech CUBBITT translations. We have noticed all three types of errors not only in CUBBITT, but we have not inspected these other MT systems in detail yet.

We implemented a rule-based tool which tries to fix such errors in post-processing.[4] It detects imperial/SI units of length, weight, speed, area and volume; units of temperature (Fahrenheit/Celsius) and currencies (USD, CZK, EUR), but it can be easily extended. By default, it keeps the units and numbers the same (except for the thousand/decimal separators), but it can be configured to convert the units and numbers. We had to deal with several edge cases, such as various ways how to write numbers and units or handling multiple numbers in a sentence with a possibly changed word order (using a word aligner).

[2] If the number of special tokens on the source side does not match the number of special tokens on the target side at inference time, we back off to translating each sentence in a given sequence independently.

[3] UDPipe is trained on Universal Dependencies (Zeman et al., 2018), where titles and headlines with no final punctuation are treated as sentences, which need to be detected by the sentence segmentation.

[4] https://github.com/vsvandelik/cubbitt-fixer

	kept		cs-en		en-cs	
	number	**unit**	#	**%**	#	**%**
A	yes	yes	2 689	86.5	3 548	85.7
B	yes	no	346	**11.1**	268	**6.5**
C	no	yes	21	**0.7**	24	**0.6**
D	no	no	21	0.7	13	0.3
E	detection failure		31	1.0	287	6.9
	total		3 108	100.0	4 140	100.0

Table 3: Automatic analysis of numerical expressions with units in a sample of 100 000 sentences from the synthetic parts of CzEng 2.0. Numerical expressions that were detected only in the source sentences, but not in the (MT) translation, are marked as *detection failure*. Cases B and C where only the unit or only the number were converted can be safely considered as errors – so the percentages are marked in bold.

Using our tool, we analyzed a sample of the synthetic training data in CzEng 2.0 and found out that at least 11.8% of Czech and 7.1% of English expressions with numbers and units are translated wrong, see Table 3.

After submitting CUNI-DocTransformer, we analyzed the WMT2021 news test sets and found out that there were only 4 sentences affected by our post-processing. All 4 cases were of the same type – "korun" was translated as "$", which was corrected to "crowns",

4 Experiments in Marian

The goals of the experiments described in this section were:

- Reimplement the Block-backtranslation training (Popel et al., 2020) in Marian (Junczys-Dowmunt et al., 2018). Block-backtranslation was first implemented in the Tensor2Tensor framework in the CUBBITT system, also known as CUNI-Transformer2018 (Popel, 2018).
- Explore the effect of Block-backtranslation (vs. standard shuffled backtranslation (Sennrich et al.)), checkpoint averaging and Tagged backtranslation (Caswell et al., 2019).
- Try a novel type of Tagged backtranslation with tags on the target side.
- Explore interactions of the above-mentioned methods.

4.1 Marian settings

We followed the standard Transformer Big hyperparameters, with 6 encoder and 6 decoder layers (unlike CUNI-DocTransformer, which has 12 encoder layers). Other differences from CUNI-DocTransformer are: Marian was trained on sentences (no document level) of up to 150 subwords (`--max-length 150`). It was trained on a single GPU (instead of 8), but using 8 batches per updated (`--optimizer-delay 8`), thus resulting in a similar effective batch size. Due to time reasons we trained all our Marian models just for a single epoch on the whole CzEng 2.0 training data, containing all three parts: authentic parallel data, synthetic CS-mono and synthetic EN-mono, i.e. using both backtranslation and forward translation (Ueffing et al., 2007; Kim and Rush, 2016). The English→Czech CUNI-DocTransformer was not trained on the EN-mono part, but it was trained "until convergence", for 700k updates (which is not easily converted to epochs because the authentic data was upsampled for the Block backtranslation), i.e. several times more updates that the Marian model. Finally, we accidentally used too small blocks in the Block backtranslation, as described in the following section.

4.2 Replicating CUBBITT

In our first experiment, we tried to replicate the CUNI-Transformer2018, which also uses the Transformer Big hyperparameters (with 6 encoder and 6 decoder layers) and sentence-level training. Our Marian results were about 1.5 BLEU worse on various WMT dev sets on average, which is better than we expected when training for a single epoch only. According to our preliminary experiments, including forward-translation data (EN-mono in our case) makes the initial training faster (i.e. better BLEU after the first epoch), although it does not improve the final BLEU when training until convergence. Forward translation data are great for fast uptraining and model distillation – the newly trained model is being trained to behave similarly as the original model used to produce the synthetic translations. The synthetic translations are consistent (if no noising is used) – the same sentence is translated always the same way.

While the final BLEU results are good enough, the learning BLEU curves on Figure 1 do not show the camel-shape progress typical for Block Backtranslation Popel et al. (2020). We also did not

observe the synergy effect of Block backtranslation and checkpoint averaging. The explanation is simple: when dividing the data for Block backtranslation, we accidentally used 10 blocks of authentic data and 20 blocks of synthetic data. Thus there was less than one checkpoint per each block on average, which goes against the main idea of Block backtranslation, where each block of authentic or synthetic data should be big enough to fit at least 8 checkpoints (considering checkpoint averaging with 8 checkpoints). We think this is the reason why we do not see any significant differences between *block* and *shuffled* in Tables 4 and 5 and also between these two tables (as an effect of checkpoint averaging).

4.3 Tagged backtranslation

For our experimenting, we decided to try labeling the data based on its authenticity — The labels would have two parts, one specifying whether the source side was an authentic sentence, or created using back translation or forward translation (Ueffing et al., 2007; Kim and Rush, 2016), and the other part specifying the same for the target side. We tried having no labels at all, labeling only one side or the other, or labeling both sides. However, in all these scenarios, every label that existed specified the authenticity of both the source and the target side. This is very similar to tagged backtranslation (Caswell et al., 2019) but we tried using our labels on the target side as well and we explored possible synergy between block ordering or checkpoint averaging by trying the different versions.

Since all of czeng20-train, czeng20-enmono, czeng20-csmono were used, the labels were `auth+auth`, `auth+synth` and `synth+auth`. In each dataset, where the label was present, it was situated at the beginning of each sentence, space-separated from the sentence itself.

In addition to the main experimenting with the four variants of source and target side labeling, we created versions with data ordered in blocks (of authentic vs backtranslated data). This resulted in eight versions being trained — all combinations of: source side labeling yes/no, targets side labeling yes/no, block order / completely shuffled data.

When the training data was ordered into blocks, there were about 10 blocks of each of the data kinds (czeng20-train, czeng20-csmono, czeng20-enmono) meaning 30 blocks in total. With our checkpoint frequency this meant that one block was slightly smaller than the data seen between two neighboring checkpoints, which are very small blocks. The completely shuffled datasets were created from the block ones by shuffling them using a random permutation. The order of data points was the same among all block-ordered datasets and same among all completely shuffled datasets.

For time reasons, we only managed to train each model on a single epoch (using marian's `--after-epochs 1`). From the training, we obtained eight variants, which we then did checkpoint averaging on, creating additional eight variants. We then evaluated these 16 variants on the wmt17 newstest dataset, and chose two representing models for each — one was the model at the end of the training, the other was the model that achieved the best BLEU score on wmt17. We evaluated these 32 models on concatenation of wmt15, wmt16 and wmt18 and chose those seven models that reached the best BLEU on this testset.

4.4 Results

We observed some differences in performance among the trained versions. The images below show the development of BLEU score (measured on a test set, not the training data) as the training progressed. We can see that there are differences among the versions but it is hard to find a pattern in them. They also do not seem to be consistent among the test sets — wmt15, wmt16, wmt17, wmt18. When wmt15, wmt16 and wmt17 are concatenated, the differences seem to largely disappear (see the tables below) and we still do not see any clear pattern in the results.

We also fail to see clear differences in performance between block ordering vs. completely shuffled corpora, and checkpoint averaging vs. no averaging. There is also no synergy between those two in our results, which is very likely caused by our setup of extremely small blocks. The blocks used in CUBBITT were large enough to contain all eight averaged checkpoints of certain models, while our blocks didn't even fully contain one checkpoint.

5 Conclusions

In this paper, we presented two sets of experiments: automatic correction of numeric expressions with units in rule-based post-processing and various settings of Tagged backtranslation.

The correction of numeric expressions with units focuses on errors which are relatively rare and do

Source labeling	Target labeling	Ordering	best-BLEU	final-BLEU
yes	yes	block	27.3	27.4
yes	yes	shuffled	27.3	27.2
yes	no	block	27.2	27.4
yes	no	shuffled	27.2	27.3
no	yes	block	27.2	27.2
no	yes	shuffled	27.4	27.4
no	no	block	27.3	27.3
no	no	shuffled	27.5	27.5

Table 4: Both BLEU scores shown were measured no the concatenation of wmt15, wmt16 and wmt18. best-BLEU is the score of the model that achieved the best BLEU on wmt17, while final-BLEU is the BLEU of the model at the end of the training. All of the models in this table are without checkpoint averaging.

Source labeling	Target labeling	Ordering	best-BLEU	final-BLEU
yes	yes	block	27.4	27.4
yes	yes	shuffled	27.2	27.2
yes	no	block	27.5	27.5
yes	no	shuffled	27.2	27.2
no	yes	block	27.3	27.3
no	yes	shuffled	27.4	27.4
no	no	block	27.3	27.3
no	no	shuffled	27.5	27.5

Table 5: This table contains the BLEU scores of models with checkpoint averaging. The columns are the same and have the same meaning as in the previous table.

Source labeling	Target labeling	Ordering	checkpoint averaging	point	wmt21 BLEU
yes	**no**	**blocks**	**yes**	**last**	**20.1**
yes	no	blocks	no	last	20.0
yes	no	blocks	yes	best	19.9
no	no	shuffled	no	last	19.9
no	no	shuffled	no	best	19.6
no	no	shuffled	yes	last	19.6
no	yes	shuffled	no	last	19.6

Table 6: These are the BLEU scores of the submited models on the wmt21 test set.

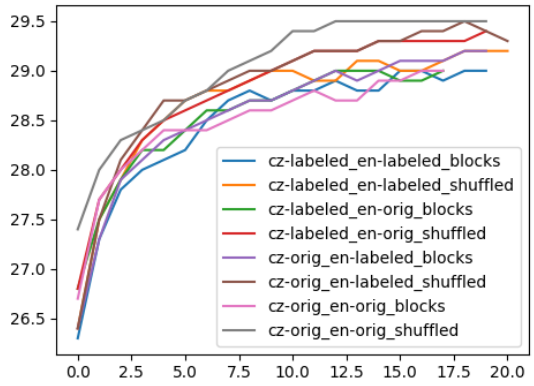

Figure 1: wmt16 BLEU training curves of averaged models

not affect automatic metrics such as BLEU much, but can result in serious misunderstanding of the meaning of the translation. Unfortunately, these errors won't be properly reflected even in the official WMT (context-sensitive, but sentence-level) manual evaluation, where each sentence's score is weighted the same, even if some errors are crucial for the meaning of the whole document.

The experiments with Tagged backtranslation using a Marian reimplementation of CUBBITT did not show any substantial differences in the results nor any consistent pattern. However, we hope that future work continuing the research on various types of training data (authentic vs. synthetic; forward vs backward; different domains) and their synergies may bring new results and better understanding of the backtranslation training etc.

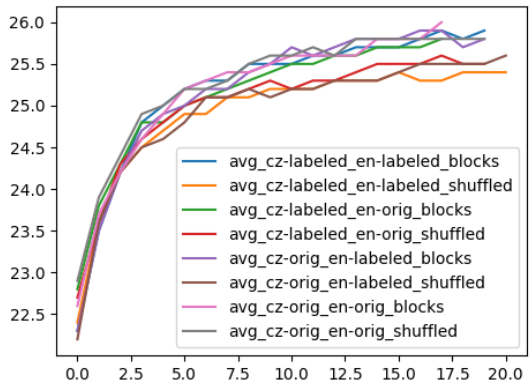

Figure 2: wmt17 BLEU training curves of averaged models

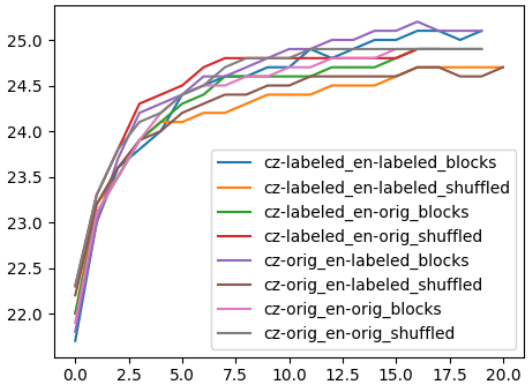

Figure 3: wmt18 BLEU training curves of averaged models

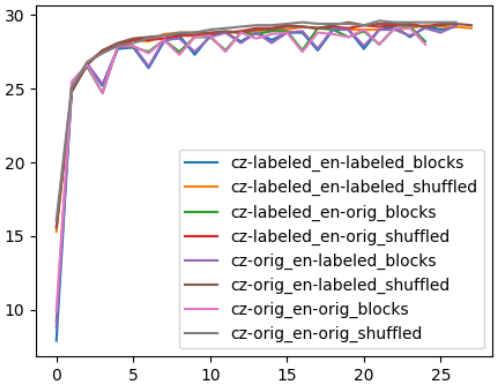

Figure 4: wmt16 BLEU training curves of non-averaged models

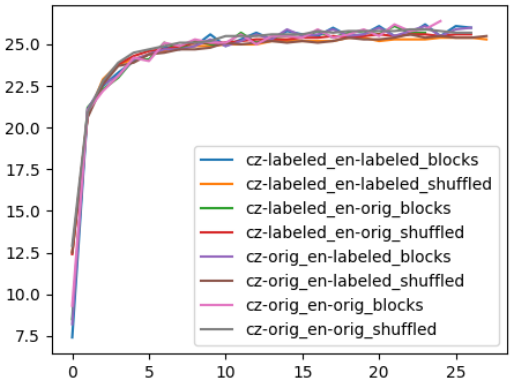

Figure 5: wmt17 BLEU training curves of non-averaged models

Acknowledgements

The work was supported by the grants 19-26934X (NEUREM3) and 20-16819X (LUSyD) by the

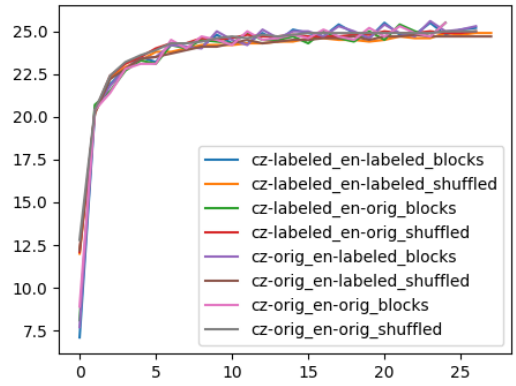

Figure 6: wmt18 BLEU training curves of non-averaged models

Czech Science Foundation. The work has been using language resources developed and distributed by the LINDAT/CLARIAHCZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Martin Popel. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel. 2020. CUNI English-Czech and English-Polish systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 269–273, Online. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch.2016b. Improving neural machine translation-models with monolingual data.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Semi-supervised model adaptation for statistical machine translation.

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 30*, pages 6000–6010. Curran Associates, Inc.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.