# A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages

**Pedro Javier Ortiz Suárez**[1,2] **Laurent Romary**[1] **Benoît Sagot**[1]
[1]Inria, Paris, France
[2]Sorbonne Université, Paris, France
{pedro.ortiz, benoit.sagot, laurent.romary}@inria.fr

## Abstract

We use the multilingual OSCAR corpus, extracted from Common Crawl via language classification, filtering and cleaning, to train monolingual contextualized word embeddings (ELMo) for five mid-resource languages. We then compare the performance of OSCAR-based and Wikipedia-based ELMo embeddings for these languages on the part-of-speech tagging and parsing tasks. We show that, despite the noise in the Common-Crawl-based OSCAR data, embeddings trained on OSCAR perform much better than monolingual embeddings trained on Wikipedia. They actually equal or improve the current state of the art in tagging and parsing for all five languages. In particular, they also improve over multilingual Wikipedia-based contextual embeddings (multilingual BERT), which almost always constitutes the previous state of the art, thereby showing that the benefit of a larger, more diverse corpus surpasses the cross-lingual benefit of multilingual embedding architectures.

## 1 Introduction

One of the key elements that has pushed the state of the art considerably in neural NLP in recent years has been the introduction and spread of transfer learning methods to the field. These methods can normally be classified in two categories according to how they are used:

- *Feature-based* methods, which involve pre-training real-valued vectors ("embeddings") at the word, sentence, or paragraph level; and using them in conjunction with a specific architecture for each individual downstream task.
- *Fine-tuning* methods, which introduce a minimal number of task-specific parameters, and instead copy the weights from a pre-trained network and then tune them to a particular downstream task.

Embeddings or language models can be divided into *fixed*, meaning that they generate a single representation for each word in the vocabulary; and *contextualized*, meaning that a representation is generated based on both the word and its surrounding context, so that a single word can have multiple representations, each one depending on how it is used.

In practice, most fixed embeddings are used as feature-based models. The most notable examples are *word2vec* (Mikolov et al., 2013), *GloVe* (Pennington et al., 2014) and *fastText* (Mikolov et al., 2018). All of them are extensively used in a variety of applications nowadays. On the other hand, contextualized word representations and language models have been developed using both feature-based architectures, the most notable examples being ELMo and Flair (Peters et al., 2018; Akbik et al., 2018), and transformer based architectures, that are commonly used in a fine-tune setting, as is the case of GPT-1, GPT-2 (Radford et al., 2018, 2019), BERT and its derivatives (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019) and more recently T5 (Raffel et al., 2019). All of them have repeatedly improved the state-of-the art in many downstream NLP tasks over the last year.

In general, the main advantage of using language models is that they are mostly built in an *unsupervised* manner and they can be trained with raw, unannotated plain text. Their main drawback is that enormous quantities of data seem to be required to properly train them especially in the case of contextualized models, for which larger corpora are thought to be needed to properly address polysemy and cover the wide range of uses that commonly exist within languages.

For gathering data in a wide range of languages,



*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714

Wikipedia is a commonly used option. It has been used to train fixed embeddings (Al-Rfou et al., 2013; Bojanowski et al., 2017) and more recently the multilingual BERT (Devlin et al., 2018), hereafter mBERT. However, for some languages, Wikipedia might not be large enough to train good quality contextualized word embeddings. Moreover, Wikipedia data all belong to the same specific genre and style. To address this problem, one can resort to crawled text from the internet; the largest and most widespread dataset of crawled text being Common Crawl.[1] Such an approach generally solves the quantity and genre/style coverage problems but might introduce noise in the data, an issue which has earned the corpus some criticism, most notably by Trinh and Le (2018) and Radford et al. (2019). Using Common Crawl also leads to data management challenges as the corpus is distributed in the form of a large set of plain text each containing a large quantity of unclassified multilingual documents from different websites.

In this paper we study the trade-off between quantity and quality of data for training contextualized representations. To this end, we use the OSCAR corpus (Ortiz Suárez et al., 2019), a freely available[2] multilingual dataset obtained by performing language classification, filtering and cleaning of the whole Common Crawl corpus.[3] OSCAR was created following the approach of Grave et al. (2018) but proposing a simple improvement on their filtering method. We then train OSCAR-based and Wikipedia-based ELMo contextualized word embeddings (Peters et al., 2018) for 5 languages: Bulgarian, Catalan, Danish, Finnish and Indonesian. We evaluate the models by attaching them to the to UDPipe 2.0 architecture (Straka, 2018; Straka et al., 2019) for dependency parsing and part-of-speech (POS) tagging. We show that the models using the OSCAR-based ELMo embeddings consistently outperform the Wikipedia-based ones, suggesting that big high-coverage noisy corpora might be better than small high-quality narrow-coverage corpora for training contextualized language representations[4]. We also establish a new state of the art for both POS tagging and dependency parsing in 6 different treebanks covering all 5 languages.

The structure of the paper is as follows. In Section 2 we describe the recent related work. In Section 3 we present, compare and analyze the corpora used to train our contextualized embeddings, and the treebanks used to train our POS tagging and parsing models. In Section 4 we examine and describe in detail the model used for our contextualized word representations, as well as the parser and the tagger we chose to evaluate the impact of corpora in the embeddings' performance in downstream tasks. Finally we provide an analysis of our results in Section 5 and in Section 6 we present our conclusions.

## 2 Related work

Since the introduction of *word2vec* (Mikolov et al., 2013), many attempts have been made to create multilingual language representations; for fixed word embeddings the most remarkable works are those of (Al-Rfou et al., 2013) and (Bojanowski et al., 2017) who created word embeddings for a large quantity of languages using Wikipedia, and later (Grave et al., 2018) who trained the fastText word embeddings for 157 languages using Common Crawl and who in fact showed that using crawled data significantly increased the performance of the embeddings especially for mid- to low-resource languages.

Regarding contextualized models, the most notable non-English contribution has been that of the mBERT (Devlin et al., 2018), which is distributed as (i) a single multilingual model for 100 different languages trained on Wikipedia data, and as (ii) a single multilingual model for both Simplified and Traditional Chinese. Four monolingual fully trained ELMo models have been distributed for Japanese, Portuguese, German and Basque[5]; 44 monolingual ELMo models[6] where also released by the *HIT-SCIR* team (Che et al., 2018) during the *CoNLL 2018 Shared Task* (Zeman et al., 2018), but their training sets where capped at 20 million words. A German BERT (Chan et al., 2019) as well as a French BERT model (called CamemBERT) (Martin et al., 2019) have also been released. In general no particular effort in creating a set of high-quality monolingual contextualized representations has been shown yet, or at least not on a scale that

[1] https://commoncrawl.org

[2] https://oscar-corpus.com

[3] Snapshot from November 2018

[4] Both the Wikipedia- and the OSCAR-based embeddings for these 5 languages are available at: https://oscar-corpus.com/#models.

[5] https://allennlp.org/elmo

[6] https://github.com/HIT-SCIR/ELMoForManyLangs

is comparable with what was done for fixed word embeddings.

For dependency parsing and POS tagging the most notable non-English specific contribution is that of the *CoNLL 2018 Shared Task* (Zeman et al., 2018), where the 1st place (LAS Ranking) was awarded to the *HIT-SCIR* team (Che et al., 2018) who used Dozat and Manning (2017)'s *Deep Bi-affine parser* and its extension described in (Dozat et al., 2017), coupled with deep contextualized ELMo embeddings (Peters et al., 2018) (capping the training set at 20 million words). The 1st place in universal POS tagging was awarded to Smith et al. (2018) who used two separate instances of Bohnet et al. (2018)'s tagger.

More recent developments in POS tagging and parsing include those of Straka et al. (2019) which couples another CoNLL 2018 shared task participant, UDPipe 2.0 (Straka, 2018), with mBERT greatly improving the scores of the original model, and UDify (Kondratyuk and Straka, 2019), which adds an extra attention layer on top of mBERT plus a Deep Bi-affine attention layer for dependency parsing and a Softmax layer for POS tagging. UDify is actually trained by concatenating the training sets of 124 different UD treebanks, creating a single POS tagging and dependency parsing model that works across 75 different languages.

## 3 Corpora

We train ELMo contextualized word embeddings for 5 languages: Bulgarian, Catalan, Danish, Finnish and Indonesian. We train one set of embeddings using only Wikipedia data, and another set using only Common-Crawl-based OSCAR data. We chose these languages primarily because they are morphologically and typologically different from one another, but also because all of the OSCAR datasets for these languages were of a sufficiently manageable size such that the ELMo pre-training was doable in less than one month. Contrary to *HIT-SCIR* team (Che et al., 2018), we do not impose any cap on the amount of data, and instead use the entirety of Wikipedia or OSCAR for each of our 5 chosen languages.

### 3.1 Wikipedia

Wikipedia is the biggest online multilingual open encyclopedia, comprising more than 40 million articles in 301 different languages. Because articles are curated by language and written in an open collaboration model, its text tends to be of very high-quality in comparison to other free online resources. This is why Wikipedia has been extensively used in various NLP applications (Wu and Weld, 2010; Mihalcea, 2007; Al-Rfou et al., 2013; Bojanowski et al., 2017). We downloaded the XML Wikipedia dumps[7] and extracted the plain-text from them using the `wikiextractor.py` script[8] from Giuseppe Attardi. We present the number of words and tokens available for each of our 5 languages in Table 1. We decided against deduplicating the Wikipedia data as the corpora are already quite small. We tokenize the 5 corpora using *UDPipe* (Straka and Straková, 2017).

| Language | Size | #Ktokens | #Kwords | #Ksentences |
|---|---|---|---|---|
| Bulgarian | 609M | 64,190 | 54,748 | 3,685 |
| Catalan | 1.1G | 211,627 | 179,108 | 8,293 |
| Danish | 338M | 60,644 | 52,538 | 3,226 |
| Finnish | 669M | 89,580 | 76,035 | 6,847 |
| Indonesian | 488M | 80,809 | 68,955 | 4,298 |

Table 1: Size of Wikipedia corpora, measured in bytes, thousands of tokens, words and sentences.

### 3.2 OSCAR

Common Crawl is a non-profit organization that produces and maintains an open, freely available repository of crawled data from the web. Common Crawl's complete archive consists of petabytes of monthly snapshots collected since 2011. Common Crawl snapshots are not classified by language, and contain a certain level of noise (e.g. one-word "sentences" such as "OK" and "Cancel" are unsurprisingly very frequent).

This is what motivated the creation of the freely available multilingual OSCAR corpus (Ortiz Suárez et al., 2019), extracted from the November 2018 snapshot, which amounts to more than 20 terabytes of plain-text. In order to create OSCAR from this Common Crawl snapshot, Ortiz Suárez et al. (2019) reproduced the pipeline proposed by (Grave et al., 2018) to process, filter and classify Common Crawl. More precisely, language classification was performed using the *fastText* linear classifier (Joulin et al., 2016, 2017), which was trained by Grave et al. (2018) to recognize 176 languages and was shown to have an extremely good accuracy to processing time trade-off. The filtering step as performed by Grave et al. (2018) consisted in only keeping the lines exceeding 100

[7] XML dumps from April 4, 2019.

[8] Available here.

| Language | Size | #Ktokens | #Kwords | #Ksentences |
|---|---|---|---|---|
| Bulgarian | 14G | 1,466,051 | 1,268,115 | 82,532 |
| Catalan | 4.3G | 831,039 | 729,333 | 31,732 |
| Danish | 9.7G | 1,828,881 | 1,620,091 | 99,766 |
| Finnish | 14G | 1,854,440 | 1,597,856 | 142,215 |
| Indonesian | 16G | 2,701,627 | 2,394,958 | 140,138 |

Table 2: Size of OSCAR subcorpora, measured in bytes, thousands of tokens, words and sentences.

| Language | OOV Wikipedia | OOV OSCAR |
|---|---|---|
| Bulgarian | 60,879 | 66,558 |
| Catalan | 34,919 | 79,678 |
| Danish | 134,677 | 123,299 |
| Finnish | 266,450 | 267,525 |
| Indonesian | 116,714 | 124,607 |

Table 3: Number of out-of-vocabulary words in random samples of 1M words for OSCAR and Wikipedia.

bytes in length.[9] However, considering that Common Crawl is a mutilingual UTF-8 encoded corpus, this 100-byte threshold creates a huge disparity between ASCII and non-ASCII encoded languages. The filtering step used to create OSCAR therefore consisted in only keeping the lines containing at least 100 UTF-8-encoded characters. Finally, as in (Grave et al., 2018), the OSCAR corpus is deduplicated, i.e. for each language, only one occurrence of a given line is included.

As we did for Wikipedia, we tokenize OSCAR corpora for the 5 languages we chose for our study using UDPipe. Table 2 provides quantitative information about the 5 resulting tokenized corpora.

We note that the original Common-Crawl-based corpus created by Grave et al. (2018) to train fastText is not freely available. Since running the experiments described in this paper, a new architecture for creating a Common-Crawl-based corpus named CCNet (Wenzek et al., 2019) has been published, although it includes specialized filtering which might result in a cleaner corpus compared to OSCAR, the resulting CCNet corpus itself was not published. Thus we chose to keep OSCAR as it remains the only very large scale, Common-Crawl-based corpus currently available and easily downloadable.

### 3.3 Noisiness

We wanted to address (Trinh and Le, 2018) and (Radford et al., 2019)'s criticisms of Common Crawl, so we devised a simple method to measure how noisy the OSCAR corpora were for our 5 languages. We randomly extract a number of lines from each corpus, such that the resulting random sample contains one million words.[10] We test if the words are in the corresponding *GNU Aspell*[11] dictionary. We repeat this task for each of the 5 languages, for both the OSCAR and the Wikipedia corpora. We compile in Table 3 the number of out-of-vocabulary tokens for each corpora.

As expected, this simple metric shows that in general the OSCAR samples contain more out-of-vocabulary words than the Wikipedia ones. However the difference in magnitude between the two is strikingly lower that one would have expected in view of the criticisms by Trinh and Le (2018) and Radford et al. (2019), thereby validating the usability of Common Crawl data when it is properly filtered, as was achieved by the OSCAR creators. We even observe that, for Danish, the number of out-of-vocabulary words in OSCAR is lower than that in Wikipedia.

## 4 Experimental Setting

The main goal of this paper is to show the impact of training data on contextualized word representations when applied in particular downstream tasks. To this end, we train different versions of the *Embeddings from Language Models* (ELMo) (Peters et al., 2018) for both the Wikipedia and OSCAR corpora, for each of our selected 5 languages. We save the models' weights at different number of epochs for each language, in order to test how corpus size affect the embeddings and to see whether and when overfitting happens when training elmo on smaller corpora.

We take each of the trained ELMo models and use them in conjunction with the UDPipe 2.0 (Straka, 2018; Straka et al., 2019) architecture for dependency parsing and POS-tagging to test our models. We train UDPipe 2.0 using gold tokenization and segmentation for each of our ELMo models, the only thing that changes from training to training is the ELMo model as hyperparameters always remain at the default values (except for number of training tokens) (Peters et al., 2018).

### 4.1 Contextualized word embeddings

*Embeddings from Language Models* (ELMo) (Peters et al., 2018) is an LSTM-based language model.

[9] Script available here.

[10] We remove tokens that are capitalized or contain less than 4 UTF-8 encoded characters, allowing us to remove bias against Wikipedia, which traditionally contains a large quantity of proper nouns and acronyms.

[11] http://aspell.net/

More precisely, it uses a bidirectional language model, which combines a forward and a backward LSTM-based language model. ELMo also computes a context-independent token representation via a CNN over characters.

We train ELMo models for Bulgarian, Catalan, Danish, Finnish and Indonesian using the OSCAR corpora on the one hand and the Wikipedia corpora on the other. We train each model for 10 epochs, as was done for the original English ELMo (Peters et al., 2018). We save checkpoints at 1st, 3rd and 5th epoch in order to investigate some concerns about possible overfitting for smaller corpora (Wikipedia in this case) raised by the original ELMo authors.[12]

### 4.2 UDPipe 2.0

For our POS tagging and dependency parsing evaluation, we use UDPipe 2.0, which has a freely available and ready to use implementation.[13] This architecture was submitted as a participant to the *2018 CoNLL Shared Task* (Zeman et al., 2018), obtaining the 3rd place in LAS ranking. UDPipe 2.0 is a multi-task model that predicts POS tags, lemmas and dependency trees jointly.

The original UDPipe 2.0 implementation calculates 3 different embeddings, namely:

- *Pre-trained word embeddings*: In the original implementation, the Wikipedia version of fastText embeddings is used (Bojanowski et al., 2017); we replace them in favor of the newer Common-Crawl-based fastText embeddings trained by Grave et al. (2018).
- *Trained word embeddings*: Randomly initialized word representations that are trained with the rest of the network.
- *Character-level word embeddings*: Computed using bi-directional GRUs of dimension 256. They represent every UTF-8 encoded character with two 256 dimensional vectors, one for the forward and one for the backward layer. This two vector representations are concatenated and are trained along the whole network.

After the CoNLL 2018 Shared Task, the UDPipe 2.0 authors added the option to concatenate contextualized representations to the embedding section of the network (Straka et al., 2019), we use this new implementation and we concatenate our pretrained deep contextualized ELMo embeddings to the three embeddings mentioned above.

Once the embedding step is completed, the concatenation of all vector representations for a word are fed to two shared bidirectional LSTM (Hochreiter and Schmidhuber, 1997) layers. The output of these two BiLSTMS is then fed to two separate specific LSTMs:

- The tagger- and lemmatizer-specific bidirectional LSTMs, with Softmax classifiers on top, which process its output and generate UPOS, XPOS, UFeats and Lemmas. The lemma classifier also takes the character-level word embeddings as input.
- The parser-specific bidirectional LSTM layer, whose output is then passed to a bi-affine attention layer (Dozat and Manning, 2017) producing labeled dependency trees.

| Treebank | #Ktokens | #Ksentences |
|---|---|---|
| Bulgarian-BTB | 156 | 11 |
| Catalan-AnCora | 530 | 17 |
| Danish-DDT | 100 | 6 |
| Finnish-FTB | 159 | 19 |
| Finnish-TDT | 202 | 15 |
| Indonesian-GSD | 121 | 6 |

Table 4: Size of treebanks, measured in thousands of tokens and sentences.

### 4.3 Treebanks

To train the selected parser and tagger (cf. Section 4.2) and evaluate the pre-trained language models in our 5 languages, we run our experiments using the Universal Dependencies (UD)[14] paradigm and its corresponding UD POS tag set (Petrov et al., 2012). We use all the treebanks available for our five languages in the UD treebank collection version 2.2 (Nivre et al., 2018), which was used for the CoNLL 2018 shared task, thus we perform our evaluation tasks in 6 different treebanks (see Table 4 for treebank size information).

- *Bulgarian BTB*: Created at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, it consists of legal documents, news articles and fiction pieces.

[12] https://github.com/allenai/bilm-tf/issues/135

[13] https://github.com/CoNLL-UD-2018/UDPipe-Future

[14] https://universaldependencies.org

- *Catalan-AnCora*: Built on top of the Spanish-Catalan *AnCora corpus* (Taulé et al., 2008), it contains mainly news articles.

- *Danish-DDT*: Converted from the *Danish Dependency Treebank* (Buch-Kromann, 2003). It includes news articles, fiction and non fiction texts and oral transcriptions.

- *Finnish-FTB*: Consists of manually annotated grammatical examples from VISK[15] (The Web Version of the Large Grammar of Finnish).

- *Finnish-TDT*: Based on the Turku Dependency Treebank (TDT). Contains texts from Wikipedia, Wikinews, news articles, blog entries, magazine articles, grammar examples, Europarl speeches, legal texts and fiction.

- *Indonesian-GSD*: Includes mainly blog entries and news articles.

## 5 Results & Discussion

### 5.1 Parsing and POS tagging results

We use UDPipe 2.0 without contextualized embeddings as our baseline for POS tagging and dependency parsing. However, we did not train the model without contextualized word embedding ourselves. We instead take the scores as they are reported in (Kondratyuk and Straka, 2019). We also compare our UDPipe 2.0 + ELMo models against the state-of-the-art results (assuming gold tokenization) for these languages, which are either UDify (Kondratyuk and Straka, 2019) or UDPipe 2.0 + mBERT (Straka et al., 2019).

Results for UPOS, UAS and LAS are shown in Table 5. We obtain the state of the art for the three metrics in each of the languages with the UDPipe 2.0 + $\text{ELMo}_{\text{OSCAR}}$ models. We also see that in every single case the UDPipe 2.0 + $\text{ELMo}_{\text{OSCAR}}$ result surpasses the UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ one, suggesting that the size of the pre-training data plays an important role in downstream task results. This is also supports our hypothesis that the OSCAR corpora, being multi-domain, exhibits a better coverage of the different styles, genres and uses present at least in these 5 languages.

Taking a closer look at the results for Danish, we see that $\text{ELMo}_{\text{Wikipedia}}$, which was trained with a mere 300MB corpus, does not show any sign of overfitting, as the UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ results considerably improve the UDPipe 2.0 baseline. This is the case for all of our $\text{ELMo}_{\text{Wikipedia}}$ models as we never see any evidence of a negative impact when we add them to the baseline model. In fact, the results of UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$ give better than previous state-of-the-art results in all metrics for the Finnish-FTB and in UPOS for the Finnish-TDT. The results for Finnish are actually quite interesting, as mBERT was pre-trained on Wikipedia and here we see that the multilingual setting in which UDify was fine-tuned exhibits sub-baseline results for all metrics, and that the UDPipe + mBERT scores are often lower than those of our UDPipe 2.0 + $\text{ELMo}_{\text{Wikipedia}}$. This actually suggests that even though the multilingual approach of mBERT (in pre-training) or UDify (in pre-training and fine-tuning) leads to better performance for high-resource languages or languages

| Treebank | Model | UPOS | UAS | LAS |
|---|---|---|---|---|
| Bulgarian BTB | UDify | 98.89 | 95.54 | 92.40 |
| | UDPipe 2.0 | 98.98 | 93.38 | 90.35 |
| | +mBERT | 99.20 | 95.34 | 92.62 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 99.17 | 94.93 | 92.05 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **99.40** | **96.01** | **93.56** |
| Catalan-AnCora | UDify | 98.89 | 94.25 | 92.33 |
| | UDPipe 2.0 | 98.88 | 93.22 | 91.06 |
| | +mBERT | **99.06** | **94.49** | 92.74 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 99.05 | 93.99 | 92.24 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **99.06** | **94.49** | **92.88** |
| Danish-DDT | UDify | 97.50 | 87.76 | 84.50 |
| | UDPipe 2.0 | 97.78 | 86.88 | 84.31 |
| | +mBERT | 98.21 | 89.32 | 87.24 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 98.45 | 89.05 | 86.92 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **98.62** | **89.84** | **87.95** |
| Finnish-FTB | UDify | 93.80 | 86.37 | 81.40 |
| | UDPipe 2.0 | 96.65 | 90.68 | 87.89 |
| | +mBERT | 96.97 | 91.68 | 89.02 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 97.27 | 92.05 | 89.62 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **98.13** | **93.81** | **92.02** |
| Finnish-TDT | UDify | 94.43 | 86.42 | 82.03 |
| | UDPipe 2.0 | 97.45 | 89.88 | 87.46 |
| | +mBERT | 97.57 | 91.66 | 89.49 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 97.65 | 91.60 | 89.34 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **98.36** | **93.54** | **91.77** |
| Indonesian-GSD | UDify | 93.36 | 86.45 | 80.10 |
| | UDPipe 2.0 | 93.69 | 85.31 | 78.99 |
| | +mBERT | 94.09 | 86.47 | 80.40 |
| | +$\text{ELMo}_{\text{Wikipedia}}$ | 93.94 | 86.16 | 80.10 |
| | +$\text{ELMo}_{\text{OSCAR}}$ | **94.12** | **86.49** | **80.59** |

Table 5: Scores from UDPipe 2.0 (from Kondratyuk and Straka, 2019), the previous state-of-the-art models UDPipe 2.0+mBERT (Straka et al., 2019) and UDify (Kondratyuk and Straka, 2019), and our ELMo-enhanced UDPipe 2.0 models. Test scores are given for UPOS, UAS and LAS in all five languages. Best scores are shown in bold, second best scores are underlined.

[15] http://scripta.kotus.fi/visk

| Treebank | Model | UPOS | UAS | LAS |
|---|---|---|---|---|
| Bulgarian BTB | UDPipe 2.0 | 98.98 | 93.38 | 90.35 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 98.81 | 93.60 | 90.21 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 99.01 | 94.32 | 91.36 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 99.03 | 94.32 | 91.38 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 99.17 | 94.93 | 92.05 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 99.28 | 95.45 | 92.98 |
| | +ELMo$_{\text{OSCAR(3)}}$ | 99.34 | 95.58 | 93.12 |
| | +ELMo$_{\text{OSCAR(5)}}$ | 99.34 | 95.63 | 93.25 |
| | +ELMo$_{\text{OSCAR(10)}}$ | **99.40** | **96.01** | **93.56** |
| Catalan-AnCora | UDPipe 2.0 | 98.88 | 93.22 | 91.06 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 98.93 | 93.24 | 91.21 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 99.02 | 93.75 | 91.93 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 99.04 | 93.86 | 92.05 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 99.05 | 93.99 | 92.24 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 99.07 | 93.92 | 92.29 |
| | +ELMo$_{\text{OSCAR(3)}}$ | **99.10** | 94.29 | 92.69 |
| | +ELMo$_{\text{OSCAR(5)}}$ | 99.07 | 94.38 | 92.75 |
| | +ELMo$_{\text{OSCAR(10)}}$ | 99.06 | **94.49** | **92.88** |
| Danish-DDT | UDPipe 2.0 | 97.78 | 86.88 | 84.31 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 97.47 | 86.98 | 84.15 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 98.03 | 88.16 | 85.81 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 98.15 | 88.24 | 85.96 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 98.45 | 89.05 | 86.92 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 98.50 | 89.47 | 87.43 |
| | +ELMo$_{\text{OSCAR(3)}}$ | 98.59 | 89.68 | 87.77 |
| | +ELMo$_{\text{OSCAR(5)}}$ | 98.59 | 89.46 | 87.64 |
| | +ELMo$_{\text{OSCAR(10)}}$ | **98.62** | **89.84** | **87.95** |
| Finnish-FTB | UDPipe 2.0 | 96.65 | 90.68 | 87.89 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 95.86 | 89.63 | 86.39 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 96.76 | 91.02 | 88.27 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 96.97 | 91.66 | 89.04 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 97.27 | 92.05 | 89.62 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 97.91 | 93.41 | 91.43 |
| | +ELMo$_{\text{OSCAR(3)}}$ | 98.00 | **93.99** | 91.98 |
| | +ELMo$_{\text{OSCAR(5)}}$ | **98.15** | 93.98 | **92.24** |
| | +ELMo$_{\text{OSCAR(10)}}$ | 98.13 | 93.81 | 92.02 |
| Finnish-TDT | UDPipe 2.0 | 97.45 | 89.88 | 87.46 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 96.73 | 89.11 | 86.33 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 97.55 | 90.84 | 88.50 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 97.55 | 91.11 | 88.88 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 97.65 | 91.60 | 89.34 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 98.27 | 93.03 | 91.29 |
| | +ELMo$_{\text{OSCAR(3)}}$ | 98.38 | **93.60** | **91.83** |
| | +ELMo$_{\text{OSCAR(5)}}$ | **98.39** | 93.57 | 91.80 |
| | +ELMo$_{\text{OSCAR(10)}}$ | 98.36 | 93.54 | 91.77 |
| Indonesian-GSD | UDPipe 2.0 | 93.69 | 85.31 | 78.99 |
| | +ELMo$_{\text{Wikipedia(1)}}$ | 93.70 | 85.81 | 79.46 |
| | +ELMo$_{\text{Wikipedia(3)}}$ | 93.90 | 86.04 | 79.72 |
| | +ELMo$_{\text{Wikipedia(5)}}$ | 94.04 | 85.93 | 79.97 |
| | +ELMo$_{\text{Wikipedia(10)}}$ | 93.94 | 86.16 | 80.10 |
| | +ELMo$_{\text{OSCAR(1)}}$ | 93.95 | 86.25 | 80.23 |
| | +ELMo$_{\text{OSCAR(3)}}$ | 94.00 | 86.21 | 80.14 |
| | +ELMo$_{\text{OSCAR(5)}}$ | **94.23** | 86.37 | 80.40 |
| | +ELMo$_{\text{OSCAR(10)}}$ | 94.12 | **86.49** | **80.59** |

Table 6: UPOS, UAS and LAS scores for the UDPipe 2.0 baseline reported by (Kondratyuk and Straka, 2019), plus the scores for checkpoints at 1, 3, 5 and 10 epochs for all the ELMo$_{\text{OSCAR}}$ and ELMo$_{\text{Wikipedia}}$. All scores are test scores. Best ELMo$_{\text{OSCAR}}$ scores are shown in bold while best ELMo$_{\text{Wikipedia}}$ scores are underlined.

that are closely related to high-resource languages, it might also significantly degrade the representations for more isolated or even simply more morphologically rich languages like Finnish. In contrast, our monolingual approach with UDPipe 2.0 + ELMo$_{\text{OSCAR}}$ improves the previous SOTA considerably, by more than 2 points for some metrics. Note however that Indonesian, which might also be seen as a relatively isolated language, does not behave in the same way as Finnish.

### 5.2 Impact of the number of training epochs

An important topic we wanted to address with our experiments was that of *overfitting* and the number of epochs one should train the contextualized embeddings for. The ELMo authors have expressed that increasing the number of training epochs is generally better, as they argue that training the ELMo model for longer reduces held-out perplexity and further improves downstream task performance.[16] This is why we intentionally fully pre-trained the ELMo$_{\text{Wikipedia}}$ to the 10 epochs of the original ELMo paper, as its authors also expressed concern over the possibility of overfitting for smaller corpora. We thus save checkpoints for each of our ELMo model at the 1, 3, 5 and 10 epoch marks so that we can properly probe for overfitting. The scores of all checkpoints are reported in Table 6. Here again we do not train the UDPipe 2.0 baselines without embedding, we just report the scores published in Kondratyuk and Straka (2019).

The first striking finding is that even though all our Wikipedia data sets are smaller than 1GB in size (except for Catalan), none of the ELMo$_{\text{Wikipedia}}$ models show any sign of overfitting, as the results continue to improve for all metrics the more we train the ELMo models, with the best results consistently being those of the fully trained 10 epoch ELMos. For all of our Wikipedia models, but those of Catalan and Indonesian, we see sub-baseline results at 1 epoch; training the model for longer is better, even if the corpora are small in size.

ELMo$_{\text{OSCAR}}$ models exhibit exactly the same behavior as ELMo$_{\text{Wikipedia}}$ models where the scores continue to improve the longer they are pre-trained, except for the case of Finnish. Here we actually see an unexpected behavior where the model performance caps around the 3$^{\text{rd}}$ to 5$^{\text{th}}$ epoch. This is surprising because the Finnish OSCAR corpus is more than 20 times bigger than our smallest Wikipedia corpus, the Danish Wikipedia, that did not exhibit

[16] Their comments on the matter can be found here.

this behavior. As previously mentioned Finnish is morphologically richer than the other languages in which we trained ELMo, we hypothesize that the representation space given by the ELMo embeddings might not be sufficiently big to extract more features from the Finnish OSCAR corpus beyond the 5$^{th}$ epoch mark, however in order to test this we would need to train a larger language model like BERT which is sadly beyond our computing infrastructure limits (cf. Subsection 5.3). However we do note that pre-training our current language model architectures in a morphologically rich language like Finnish might actually better expose the limits of our existing approaches to language modeling.

One last thing that it is important to note with respect to the number of training epochs is that even though we fully pre-trained our ELMo$_{\text{Wikipedia}}$'s and ELMo$_{\text{OSCAR}}$'s to the recommended 10 epoch mark, and then compared them against one another, the number of training steps between both pre-trained models differs drastically due to the big difference in corpus size (for Indonesian, for instance, 10 epochs correspond to 78K steps for ELMo$_{\text{Wikipedia}}$ and to 2.6M steps for OSCAR; the complete picture is provided in the Appendix, in Table 8). In fact, we can see in Table 6 that all the UDPipe 2.0 + ELMo$_{\text{OSCAR(1)}}$ perform better than the UDPipe 2.0 + ELMo$_{\text{Wikipedia(1)}}$ models across all metrics. Thus we believe that talking in terms of training steps as opposed to training epochs might be a more transparent way of comparing two pre-trained models.

### 5.3 Computational cost and carbon footprint

Considering the discussion above, we believe an interesting follow-up to our experiments would be training the ELMo models for more of the languages included in the OSCAR corpus. However training ELMo is computationally costly, and one way to estimate this cost, as pointed out by Strubell et al. (2019), is by using the training times of each model to compute both power consumption and $CO_2$ emissions.

In our set-up we used two different machines, each one having 4 NVIDIA GeForce GTX 1080 Ti graphic cards and 128GB of RAM, the difference between the machines being that one uses a single Intel Xeon Gold 5118 processor, while the other uses two Intel Xeon E5-2630 v4 processors. One GeForce GTX 1080 Ti card is rated at around 250 W,[17] the Xeon Gold 5118 processor is rated at 105 W,[18] while one Xeon E5-2630 v4 is rated at 85 W.[19] For the DRAM we can use the work of Desrochers et al. (2016) to estimate the total power draw of 128GB of RAM at around 13W. Having this information, we can now use the formula proposed by Strubell et al. (2019) in order to compute the total power required to train one ELMo model:

$$p_t = \frac{1.58t(cp_c + p_r + gp_g)}{1000}$$

Where $c$ and $g$ are the number of CPUs and GPUs respectively, $p_c$ is the average power draw (in Watts) from all CPU sockets, $p_r$ the average power draw from all DRAM sockets, and $p_g$ the average power draw of a single GPU. We estimate the total power consumption by adding GPU, CPU and DRAM consumptions, and then multiplying by the *Power Usage Effectiveness* (PUE), which accounts for the additional energy required to support the compute infrastructure. We use a PUE coefficient of 1.58, the 2018 global average for data centers (Strubell et al., 2019). In table 7 we report the training times in both hours and days, as well as the total power draw (in Watts) of the system used to train each individual ELMo model. We use this in-

| Language | Power | Hours | Days | KWh·PUE | $CO_2$e |
|---|---|---|---|---|---|
| *OSCAR-Based ELMos* | | | | | |
| Bulgarian | 1183 | 515.00 | 21.45 | 962.61 | 49.09 |
| Catalan | 1118 | 199.98 | 8.33 | 353.25 | 18.02 |
| Danish | 1183 | 200.89 | 8.58 | 375.49 | 19.15 |
| Finnish | 1118 | 591.25 | 24.63 | 1044.40 | 53.26 |
| Indonesian | 1183 | 694.26 | 28.93 | 1297.67 | 66.18 |
| *Wikipedia-Based ELMos* | | | | | |
| Bulgarian | 1118 | 15.45 | 0.64 | 27.29 | 1.39 |
| Catalan | 1118 | 51.08 | 2.13 | 90.22 | 4.60 |
| Danish | 1118 | 14.56 | 0.61 | 25,72 | 1.31 |
| Finnish | 1118 | 21.79 | 0.91 | 38.49 | 1.96 |
| Indonesian | 1118 | 20.28 | 0.84 | 35.82 | 1.82 |
| TOTAL EMISSIONS | | | | | 216.78 |

Table 7: Average power draw (Watts), training times (in both hours and days), mean power consumption (KWh) and $CO_2$ emissions (kg) for each ELMo model trained.

[17] https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-1080-ti/specifications

[18] https://ark.intel.com/content/www/us/en/ark/products/120473/intel-xeon-gold-5118-processor-16-5m-cache-2-30-ghz.html

[19] https://ark.intel.com/content/www/us/en/ark/products/92981/intel-xeon-processor-e5-2630-v4-25m-cache-2-20-ghz.html

formation to compute the total power consumption of each ELMo, also reported in table 7.

We can further estimate the $CO_2$ emissions in kilograms of each single model by multiplying the total power consumption by the average $CO_2$ emissions per kWh in France (where the models were trained). According to the RTE (Réseau de transport d'électricité / Electricity Transmission Network) the average emission per kWh were around 51g/kWh in November 2019,[20] when the models were trained. Thus the total $CO_2$ emissions in kg for one single model can be computed as:

$$\mathrm{CO_2e} = 0.051 p_t$$

All emissions for the ELMo models are also reported in table 7.

We do not report the power consumption or the carbon footprint of training the UDPipe 2.0 architecture, as each model took less than 4 hours to train on a machine using a single NVIDIA Tesla V100 card. Also, this machine was shared during training time, so it would be extremely difficult to accurately estimate the power consumption of these models.

Even though it would have been interesting to replicate all our experiments and computational cost estimations with state-of-the-art fine-tuning models such as BERT, XLNet, RoBERTa or ALBERT, we recall that these transformer-based architectures are extremely costly to train, as noted by the BERT authors on the official BERT GitHub repository,[21] and are currently beyond the scope of our computational infrastructure. However we believe that ELMo contextualized word embeddings remain a useful model that still provide an extremely good trade-off between performance to training cost, even setting new state-of-the-art scores in parsing and POS tagging for our five chosen languages, performing even better than the multilingual mBERT model.

## 6 Conclusions

In this paper, we have explored the use of the Common-Crawl-based OSCAR corpora to train ELMo contextualized embeddings for five typologically diverse mid-resource languages. We have compared them with Wikipedia-based ELMo embeddings on two classical NLP tasks, POS tagging and parsing, using state-of-the-art neural architectures. Our goal was to explore whether the noisiness level of Common Crawl data, often invoked to criticize the use of such data, could be compensated by its larger size; for some languages, the OSCAR corpus is several orders of magnitude larger than the corresponding Wikipedia. Firstly, we found that when properly filtered, Common Crawl data is not massively noisier than Wikipedia. Secondly, we show that embeddings trained using OSCAR data consistently outperform Wikipedia-based embeddings, to the extent that they allow us to improve the state of the art in POS tagging and dependency parsing for all the 6 chosen treebanks. Thirdly, we observe that more training epochs generally results in better embeddings even when the training data is relatively small, as is the case for Wikipedia.

Our experiments show that Common-Crawl-based data such as the OSCAR corpus can be used to train high-quality contextualized embeddings, even for languages for which more standard textual resources lack volume or genre variety. This could result in better performances in a number of NLP tasks for many non highly resourced languages.


### Acknowledgments

We want to thank Ganesh Jawahar for his insightful comments and suggestions during the early stages of this project. This work was partly funded by the French national ANR grant BASNUM (ANR-18-CE38-0003), as well as by the last author's chair in the PRAIRIE institute,[22] funded by the French national ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001. The authors are grateful to Inria Sophia Antipolis - Méditerranée "Nef"[23] computation cluster for providing resources and support.


## References


Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations


[20] https://www.rte-france.com/fr/eco2mix/eco2mix-co2

[21] https://github.com/google-research/bert

[22] http://prairie-institute.fr/

[23] https://wiki.inria.fr/wikis/ClustersSophia

for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Matthias Buch-Kromann. 2003. The danish dependency treebank and the dtag treebank tool. In *2nd Workshop on Treebanks and Linguistic Theories (TLT), Sweden*, pages 217–220.

Branden Chan, Timo Möller, Malte Pietsch, Tanay Soni, and Chin Man Yeung. 2019. German BERT. https://deepset.ai/german-bert.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Spencer Desrochers, Chad Paradis, and Vincent M. Weaver. 2016. A validation of dram rapl power measurements. In *Proceedings of the Second International Symposium on Memory Systems*, MEMSYS '16, page 455–470, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Multilingual BERT. https://github.com/google-research/bert/blob/master/multilingual.md.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. *arXiv e-prints*, page arXiv:1904.02099.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv e-prints*, page arXiv:1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model. *arXiv e-prints*, page arXiv:1911.03894.

Rada Mihalcea. 2007. Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 196–203, Rochester, New York. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. *Challenges in the Management of Large Corpora (CMLC-7) 2019*, page 9.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2089–2096. European Language Resources Association (ELRA).

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints*, page arXiv:1910.10683.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka, Jana Straková, and Jan Hajic. 2019. Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing. *CoRR*, abs/1908.07448.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. *arXiv e-prints*, page arXiv:1911.00359.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

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.

# A Appendix

## A.1 Number of training steps for each checkpoint and each corpus

| Language | 1 Epoch | 3 Epochs | 5 Epochs | 10 Epochs |
|---|---|---|---|---|
| *Wikipedia-Based ELMos* | | | | |
| Bulgarian | 6,268 | 18,804 | 31,340 | 62,680 |
| Catalan | 20,666 | 61,998 | 103,330 | 206,660 |
| Danish | 5,922 | 17,766 | 29,610 | 59,220 |
| Finnish | 8,763 | 26,289 | 43,815 | 87,630 |
| Indonesian | 7,891 | 23,673 | 39,455 | 78,910 |
| *OSCAR-Based ELMos* | | | | |
| Bulgarian | 143,169 | 429,507 | 715,845 | 1,431,690 |
| Catalan | 81,156 | 243,468 | 405,780 | 811,560 |
| Danish | 81,156 | 243,468 | 405,780 | 811,560 |
| Finnish | 181,230 | 543,690 | 906,150 | 1,812,300 |
| Indonesian | 263,830 | 791,490 | 1,319,150 | 2,638,300 |

Table 8: Number of training steps for each checkpoint, for the $\text{ELMo}_{\text{Wikipedia}}$ and $\text{ELMo}_{\text{OSCAR}}$ of each language.