HAL open science

CamemBERT: a Tasty French Language Model

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, Benoît Sagot

To cite this version:

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, et al.. CamemBERT: a Tasty French Language Model. ACL 2020 - 58th Annual Meeting of the Association for Computational Linguistics, Jul 2020, Seattle / Virtual, United States. 10.18653/v1/2020.acl-main.645 . hal-02889805

HAL Id: hal-02889805

https://inria.hal.science/hal-02889805v1

Submitted on 5 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CamemBERT: a Tasty French Language Model

Louis Martin[*1,2,3] **Benjamin Muller**[*2,3] **Pedro Javier Ortiz Suárez**[*2,3]
Yoann Dupont[3] **Laurent Romary**[2] **Éric Villemonte de la Clergerie**[2]
Djamé Seddah[2] **Benoît Sagot**[2]
[1]Facebook AI Research, Paris, France [2]Inria, Paris, France
[3]Sorbonne Université, Paris, France
louismartin@fb.com
{benjamin.muller, pedro.ortiz, laurent.romary,
eric.de_la_clergerie, djame.seddah, benoit.sagot}@inria.fr
yoa.dupont@gmail.com

Abstract

Pretrained language models are now ubiquitous in Natural Language Processing. Despite their success, most available models have either been trained on English data or on the concatenation of data in multiple languages. This makes practical use of such models—in all languages except English—very limited. In this paper, we investigate the feasibility of training monolingual Transformer-based language models for other languages, taking French as an example and evaluating our language models on part-of-speech tagging, dependency parsing, named entity recognition and natural language inference tasks. We show that the use of web crawled data is preferable to the use of Wikipedia data. More surprisingly, we show that a relatively small web crawled dataset (4GB) leads to results that are as good as those obtained using larger datasets (130+GB). Our best performing model CamemBERT reaches or improves the state of the art in all four downstream tasks.

1 Introduction

Pretrained word representations have a long history in Natural Language Processing (NLP), from non-contextual (Brown et al., 1992; Ando and Zhang, 2005; Mikolov et al., 2013; Pennington et al., 2014) to contextual word embeddings (Peters et al., 2018; Akbik et al., 2018). Word representations are usually obtained by training language model architectures on large amounts of textual data and then fed as an input to more complex task-specific architectures. More recently, these specialized architectures have been replaced altogether by large-scale pretrained language models which are *fine-tuned* for each application considered. This shift has resulted in large improvements in performance over a wide range of tasks (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019; Raffel et al., 2019).

These transfer learning methods exhibit clear advantages over more traditional task-specific approaches. In particular, they can be trained in an *unsupervized* manner, thereby taking advantage of the information contained in large amounts of raw text. Yet they come with implementation challenges, namely the amount of data and computational resources needed for pretraining, which can reach hundreds of gigabytes of text and require hundreds of GPUs (Yang et al., 2019; Liu et al., 2019). This has limited the availability of these state-of-the-art models to the English language, at least in the monolingual setting. This is particularly inconvenient as it hinders their practical use in NLP systems. It also prevents us from investigating their language modelling capacity, for instance in the case of morphologically rich languages.

Although multilingual models give remarkable results, they are often larger, and their results, as we will observe for French, can lag behind their monolingual counterparts for high-resource languages.

In order to reproduce and validate results that have so far only been obtained for English, we take advantage of the newly available multilingual corpora OSCAR (Ortiz Suárez et al., 2019) to train a monolingual language model for French, dubbed CamemBERT. We also train alternative versions of CamemBERT on different smaller corpora with different levels of homogeneity in genre and style in order to assess the impact of these parameters on downstream task performance. CamemBERT uses the RoBERTa architecture (Liu et al., 2019), an improved variant of the high-performing and widely used BERT architecture (Devlin et al., 2019).

We evaluate our model on four different downstream tasks for French: part-of-speech (POS) tagging, dependency parsing, named entity recognition (NER) and natural language inference (NLI).

[*]Equal contribution. Order determined alphabetically.

CamemBERT improves on the state of the art in all four tasks compared to previous monolingual and multilingual approaches including mBERT, XLM and XLM-R, which confirms the effectiveness of large pretrained language models for French.

We make the following contributions:

- First release of a monolingual RoBERTa model for the French language using recently introduced large-scale open source corpora from the Oscar collection and first outside the original BERT authors to release such a large model for an other language than English.[1]
- We achieve state-of-the-art results on four downstream tasks: POS tagging, dependency parsing, NER and NLI, confirming the effectiveness of BERT-based language models for French.
- We demonstrate that small and diverse training sets can achieve similar performance to large-scale corpora, by analysing the importance of the pretraining corpus in terms of size and domain.

2 Previous work

2.1 Contextual Language Models

From non-contextual to contextual word embeddings The first neural word vector representations were non-contextualized word embeddings, most notably word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fastText (Mikolov et al., 2018), which were designed to be used as input to task-specific neural architectures. Contextualized word representations such as ELMo (Peters et al., 2018) and flair (Akbik et al., 2018), improved the representational power of word embeddings by taking context into account. Among other reasons, they improved the performance of models on many tasks by handling words polysemy. This paved the way for larger contextualized models that replaced downstream architectures altogether in most tasks. Trained with language modeling objectives, these approaches range from LSTM-based architectures such as (Dai and Le, 2015), to the successful transformer-based architectures such as GPT2 (Radford et al., 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and more recently ALBERT (Lan et al., 2019) and T5 (Raffel et al., 2019).

[1]Released at: https://camembert-model.fr under the MIT open-source license.

Non-English contextualized models Following the success of large pretrained language models, they were extended to the multilingual setting with multilingual BERT (hereafter mBERT) (Devlin et al., 2018), a single multilingual model for 104 different languages trained on Wikipedia data, and later XLM (Lample and Conneau, 2019), which significantly improved unsupervized machine translation. More recently XLM-R (Conneau et al., 2019), extended XLM by training on 2.5TB of data and outperformed previous scores on multilingual benchmarks. They show that multilingual models can obtain results competitive with monolingual models by leveraging higher quality data from other languages on specific downstream tasks.

A few non-English monolingual models have been released: ELMo models for Japanese, Portuguese, German and Basque[2] and BERT for Simplified and Traditional Chinese (Devlin et al., 2018) and German (Chan et al., 2019).

However, to the best of our knowledge, no particular effort has been made toward training models for languages other than English at a scale similar to the latest English models (e.g. RoBERTa trained on more than 100GB of data).

BERT and RoBERTa Our approach is based on RoBERTa (Liu et al., 2019) which itself is based on BERT (Devlin et al., 2019). BERT is a multi-layer bidirectional Transformer encoder trained with a masked language modeling (MLM) objective, inspired by the Cloze task (Taylor, 1953). It comes in two sizes: the $\text{BERT}_{\text{BASE}}$ architecture and the $\text{BERT}_{\text{LARGE}}$ architecture. The $\text{BERT}_{\text{BASE}}$ architecture is 3 times smaller and therefore faster and easier to use while $\text{BERT}_{\text{LARGE}}$ achieves increased performance on downstream tasks. RoBERTa improves the original implementation of BERT by identifying key design choices for better performance, using dynamic masking, removing the next sentence prediction task, training with larger batches, on more data, and for longer.

3 Downstream evaluation tasks

In this section, we present the four downstream tasks that we use to evaluate CamemBERT, namely: Part-Of-Speech (POS) tagging, dependency parsing, Named Entity Recognition (NER) and Natural Language Inference (NLI). We also present the baselines that we will use for comparison.

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

Tasks POS tagging is a low-level syntactic task, which consists in assigning to each word its corresponding grammatical category. Dependency parsing consists in predicting the labeled syntactic tree in order to capture the syntactic relations between words.

For both of these tasks we run our experiments using the Universal Dependencies (UD)[3] framework and its corresponding UD POS tag set (Petrov et al., 2012) and UD treebank collection (Nivre et al., 2018), which was used for the CoNLL 2018 shared task (Seker et al., 2018). We perform our evaluations on the four freely available French UD treebanks in UD v2.2: GSD (McDonald et al., 2013), Sequoia[4] (Candito and Seddah, 2012; Candito et al., 2014), Spoken (Lacheret et al., 2014; Bawden et al., 2014)[5], and ParTUT (Sanguinetti and Bosco, 2015). A brief overview of the size and content of each treebank can be found in Table 1.

Treebank	#Tokens	#Sentences	Genres
GSD	389,363	16,342	Blogs, News Reviews, Wiki
Sequoia	68,615	3,099	Medical, News Non-fiction, Wiki
Spoken	34,972	2,786	Spoken
ParTUT	27,658	1,020	Legal, News, Wikis
FTB	350,930	27,658	News

Table 1: Statistics on the treebanks used in POS tagging, dependency parsing, and NER (FTB).

We also evaluate our model in NER, which is a sequence labeling task predicting which words refer to real-world objects, such as people, locations, artifacts and organisations. We use the French Treebank[6] (FTB) (Abeillé et al., 2003) in its 2008 version introduced by Candito and Crabbé (2009) and with NER annotations by Sagot et al. (2012). The FTB contains more than 11 thousand entity mentions distributed among 7 different entity types. A brief overview of the FTB can also be found in Table 1.

Finally, we evaluate our model on NLI, using the French part of the XNLI dataset (Conneau et al., 2018). NLI consists in predicting whether a hypothesis sentence is entailed, neutral or contradicts a premise sentence. The XNLI dataset is the extension of the Multi-Genre NLI (MultiNLI) corpus (Williams et al., 2018) to 15 languages by translating the validation and test sets manually into each of those languages. The English training set is machine translated for all languages other than English. The dataset is composed of 122k train, 2490 development and 5010 test examples for each language. As usual, NLI performance is evaluated using accuracy.

Baselines In dependency parsing and POS-tagging we compare our model with:

- *mBERT*: The multilingual cased version of BERT (see Section 2.1). We fine-tune mBERT on each of the treebanks with an additional layer for POS-tagging and dependency parsing, in the same conditions as our CamemBERT model.

- $XLM_{MLM\text{-}TLM}$: A multilingual pretrained language model from Lample and Conneau (2019), which showed better performance than mBERT on NLI. We use the version available in the Hugging's Face transformer library (Wolf et al., 2019); like mBERT, we fine-tune it in the same conditions as our model.

- *UDify* (Kondratyuk, 2019): A multitask and multilingual model based on mBERT, UDify is trained simultaneously on 124 different UD treebanks, creating a single POS tagging and dependency parsing model that works across 75 different languages. We report the scores from Kondratyuk (2019) paper.

- *UDPipe Future* (Straka, 2018): An LSTM-based model ranked 3rd in dependency parsing and 6th in POS tagging at the CoNLL 2018 shared task (Seker et al., 2018). We report the scores from Kondratyuk (2019) paper.

- *UDPipe Future + mBERT + Flair* (Straka et al., 2019): The original UDPipe Future implementation using mBERT and Flair as feature-based contextualized word embeddings. We report the scores from Straka et al. (2019) paper.

In French, no extensive work has been done on NER due to the limited availability of annotated corpora. Thus we compare our model with the only recent available baselines set by Dupont (2017), who trained both CRF (Lafferty et al., 2001) and

[3] https://universaldependencies.org

[4] https://deep-sequoia.inria.fr

[5] Speech transcript uncased that includes annotated disfluencies without punctuation

[6] This dataset has only been stored and used on Inria's servers after signing the research-only agreement.

BiLSTM-CRF (Lample et al., 2016) architectures on the FTB and enhanced them using heuristics and pretrained word embeddings. Additionally, as for POS and dependency parsing, we compare our model to a fine-tuned version of mBERT for the NER task.

For XNLI, we provide the scores of mBERT which has been reported for French by Wu and Dredze (2019). We report scores from $\text{XLM}_{\text{MLM-TLM}}$ (described above), the best model from Lample and Conneau (2019). We also report the results of XLM-R (Conneau et al., 2019).

4 CamemBERT: a French Language Model

In this section, we describe the pretraining data, architecture, training objective and optimisation setup we use for CamemBERT.

4.1 Training data

Pretrained language models benefits from being trained on large datasets (Devlin et al., 2018; Liu et al., 2019; Raffel et al., 2019). We therefore use the French part of the OSCAR corpus (Ortiz Suárez et al., 2019), a pre-filtered and pre-classified version of Common Crawl.[7]

OSCAR is a set of monolingual corpora extracted from Common Crawl snapshots. It follows the same approach as (Grave et al., 2018) by using a language classification model based on the fastText linear classifier (Grave et al., 2017; Joulin et al., 2016) pretrained on Wikipedia, Tatoeba and SETimes, which supports 176 languages. No other filtering is done. We use a non-shuffled version of the French data, which amounts to 138GB of raw text and 32.7B tokens after subword tokenization.

4.2 Pre-processing

We segment the input text data into subword units using SentencePiece (Kudo and Richardson, 2018). SentencePiece is an extension of Byte-Pair encoding (BPE) (Sennrich et al., 2016) and WordPiece (Kudo, 2018) that does not require pre-tokenization (at the word or token level), thus removing the need for language-specific tokenisers. We use a vocabulary size of 32k subword tokens. These subwords are learned on 10^7 sentences sampled randomly from the pretraining dataset. We do not use subword regularisation (i.e. sampling from multiple possible segmentations) for the sake of simplicity.

[7] https://commoncrawl.org/about/

4.3 Language Modeling

Transformer Similar to RoBERTa and BERT, CamemBERT is a multi-layer bidirectional Transformer (Vaswani et al., 2017). Given the widespread usage of Transformers, we do not describe them here and refer the reader to (Vaswani et al., 2017). CamemBERT uses the original architectures of $\text{BERT}_{\text{BASE}}$ (12 layers, 768 hidden dimensions, 12 attention heads, 110M parameters) and $\text{BERT}_{\text{LARGE}}$ (24 layers, 1024 hidden dimensions, 16 attention heads, 335M parameters). CamemBERT is very similar to RoBERTa, the main difference being the use of whole-word masking and the usage of SentencePiece tokenization (Kudo and Richardson, 2018) instead of WordPiece (Schuster and Nakajima, 2012).

Pretraining Objective We train our model on the Masked Language Modeling (MLM) task. Given an input text sequence composed of N tokens $x_1, ..., x_N$, we select 15% of tokens for possible replacement. Among those selected tokens, 80% are replaced with the special <MASK> token, 10% are left unchanged and 10% are replaced by a random token. The model is then trained to predict the initial masked tokens using cross-entropy loss.

Following the RoBERTa approach, we dynamically mask tokens instead of fixing them statically for the whole dataset during preprocessing. This improves variability and makes the model more robust when training for multiple epochs.

Since we use SentencePiece to tokenize our corpus, the input tokens to the model are a mix of whole words and subwords. An upgraded version of BERT[8] and Joshi et al. (2019) have shown that masking whole words instead of individual subwords leads to improved performance. Whole-word Masking (WWM) makes the training task more difficult because the model has to predict a whole word rather than predicting only part of the word given the rest. We train our models using WWM by using whitespaces in the initial untokenized text as word delimiters.

WWM is implemented by first randomly sampling 15% of the words in the sequence and then considering all subword tokens in each of this 15% for candidate replacement. This amounts to a proportion of selected tokens that is close to the original 15%. These tokens are then either replaced by

[8] https://github.com/google-research/bert/blob/master/README.md

<MASK> tokens (80%), left unchanged (10%) or replaced by a random token.

Subsequent work has shown that the next sentence prediction (NSP) task originally used in BERT does not improve downstream task performance (Lample and Conneau, 2019; Liu et al., 2019), thus we also remove it.

Optimisation Following (Liu et al., 2019), we optimize the model using Adam (Kingma and Ba, 2014) ($\beta_1 = 0.9$, $\beta_2 = 0.98$) for 100k steps with large batch sizes of 8192 sequences, each sequence containing at most 512 tokens. We enforce each sequence to only contain complete paragraphs (which correspond to lines in the our pretraining dataset).

Pretraining We use the RoBERTa implementation in the fairseq library (Ott et al., 2019). Our learning rate is warmed up for 10k steps up to a peak value of 0.0007 instead of the original 0.0001 given our large batch size, and then fades to zero with polynomial decay. Unless otherwise specified, our models use the BASE architecture, and are pretrained for 100k backpropagation steps on 256 Nvidia V100 GPUs (32GB each) for a day. We do not train our models for longer due to practical considerations, even though the performance still seemed to be increasing.

4.4 Using CamemBERT for downstream tasks

We use the pretrained CamemBERT in two ways. In the first one, which we refer to as *fine-tuning*, we fine-tune the model on a specific task in an end-to-end manner. In the second one, referred to as *feature-based embeddings* or simply *embeddings*, we extract frozen contextual embedding vectors from CamemBERT. These two complementary approaches shed light on the quality of the pretrained hidden representations captured by CamemBERT.

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's architecture. Following the work done on BERT (Devlin et al., 2019), for sequence tagging and sequence labeling we append a linear layer that respectively takes as input the last hidden representation of the <s> special token and the last hidden representation of the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by Dozat and Manning (2017). We refer the reader to this article for more details on this module. We fine-tune on XNLI by adding a classification head composed of one hidden layer with a non-linearity and one linear projection layer, with input dropout for both.

We fine-tune CamemBERT independently for each task and each dataset. We optimize the model using the Adam optimiser (Kingma and Ba, 2014) with a fixed learning rate. We run a grid search on a combination of learning rates and batch sizes. We select the best model on the validation set out of the 30 first epochs. For NLI we use the default hyperparameters provided by the authors of RoBERTa on the MNLI task.[9] Although this might have pushed the performances even further, we do not apply any regularisation techniques such as weight decay, learning rate warm-up or discriminative fine-tuning, except for NLI. We show that fine-tuning CamemBERT in a straightforward manner leads to state-of-the-art results on all tasks and outperforms the existing BERT-based models in all cases. The POS tagging, dependency parsing, and NER experiments are run using Hugging Face's Transformer library extended to support CamemBERT and dependency parsing (Wolf et al., 2019). The NLI experiments use the fairseq library following the RoBERTa implementation.

Embeddings Following Straková et al. (2019) and Straka et al. (2019) for mBERT and the English BERT, we make use of CamemBERT in a feature-based embeddings setting. In order to obtain a representation for a given token, we first compute the average of each sub-word's representations in the last four layers of the Transformer, and then average the resulting sub-word vectors.

We evaluate CamemBERT in the embeddings setting for POS tagging, dependency parsing and NER; using the open-source implementations of Straka et al. (2019) and Straková et al. (2019).[10]

5 Evaluation of CamemBERT

In this section, we measure the performance of our models by evaluating them on the four aforementioned tasks: POS tagging, dependency parsing, NER and NLI.

[9] More details at https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md.

[10] UDPipe Future is available at https://github.com/CoNLL-UD-2018/UDPipe-Future, and the code for nested NER is available at https://github.com/ufal/acl2019_nested_ner.

Model	GSD		Sequoia		Spoken		ParTUT	
	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS
mBERT (fine-tuned)	97.48	89.73	98.41	91.24	96.02	78.63	97.35	91.37
$XLM_{MLM-TLM}$ (fine-tuned)	98.13	90.03	98.51	91.62	96.18	80.89	97.39	89.43
UDify (Kondratyuk, 2019)	97.83	91.45	97.89	90.05	96.23	80.01	96.12	88.06
UDPipe Future (Straka, 2018)	97.63	88.06	98.79	90.73	95.91	77.53	96.93	89.63
+ mBERT + Flair (emb.) (Straka et al., 2019)	97.98	90.31	**99.32**	93.81	**97.23**	81.40	97.64	92.47
CamemBERT (fine-tuned)	**98.18**	**92.57**	99.29	**94.20**	96.99	81.37	**97.65**	**93.43**
UDPipe Future + CamemBERT (embeddings)	97.96	90.57	99.25	93.89	97.09	**81.81**	97.50	92.32

Table 2: **POS** and **dependency parsing** scores on 4 French treebanks, reported on test sets assuming gold tokenization and segmentation (best model selected on validation out of 4). Best scores in bold, second best underlined.

Model	F1
SEM (CRF) (Dupont, 2017)	85.02
LSTM-CRF (Dupont, 2017)	85.57
mBERT (fine-tuned)	87.35
CamemBERT (fine-tuned)	89.08
LSTM+CRF+CamemBERT (embeddings)	**89.55**

Table 3: **NER** scores on the FTB (best model selected on validation out of 4). Best scores in bold, second best underlined.

Model	Acc.	#Params
mBERT (Devlin et al., 2019)	76.9	175M
$XLM_{MLM-TLM}$ (Lample and Conneau, 2019)	80.2	250M
$XLM-R_{BASE}$ (Conneau et al., 2019)	80.1	270M
CamemBERT (fine-tuned)	**82.5**	110M
Supplement: LARGE models		
$XLM-R_{LARGE}$ (Conneau et al., 2019)	85.2	550M
$CamemBERT_{LARGE}$ (fine-tuned)	**85.7**	335M

Table 4: **NLI** accuracy on the French XNLI test set (best model selected on validation out of 10). Best scores in bold, second best underlined.

POS tagging and dependency parsing For POS tagging and dependency parsing, we compare CamemBERT with other models in the two settings: *fine-tuning* and as *feature-based embeddings*. We report the results in Table 2.

CamemBERT reaches state-of-the-art scores on all treebanks and metrics in both scenarios. The two approaches achieve similar scores, with a slight advantage for the fine-tuned version of CamemBERT, thus questioning the need for complex task-specific architectures such as UDPipe Future.

Despite a much simpler optimisation process and no task specific architecture, fine-tuning CamemBERT outperforms UDify on all treebanks and sometimes by a large margin (e.g. +4.15% LAS on Sequoia and +5.37 LAS on ParTUT). CamemBERT also reaches better performance than other multilingual pretrained models such as mBERT and $XLM_{MLM-TLM}$ on all treebanks.

CamemBERT achieves overall slightly better results than the previous state-of-the-art and task-specific architecture UDPipe Future+mBERT +Flair, except for POS tagging on Sequoia and POS tagging on Spoken, where CamemBERT lags by 0.03% and 0.14% UPOS respectively. UDPipe Future+mBERT +Flair uses the contextualized string embeddings Flair (Akbik et al., 2018), which are in fact pretrained contextualized character-level word embeddings specifically designed to handle misspelled words as well as subword structures such as prefixes and suffixes. This design choice might explain the difference in score for POS tagging with CamemBERT, especially for the Spoken treebank where words are not capitalized, a factor that might pose a problem for CamemBERT which was trained on capitalized data, but that might be properly handle by Flair on the UDPipe Future+mBERT +Flair model.

Named-Entity Recognition For NER, we similarly evaluate CamemBERT in the fine-tuning setting and as input embeddings to the task specific architecture LSTM+CRF. We report these scores in Table 3.

In both scenarios, CamemBERT achieves higher F1 scores than the traditional CRF-based architectures, both non-neural and neural, and than fine-tuned multilingual BERT models.[11]

Using CamemBERT as embeddings to the traditional LSTM+CRF architecture gives slightly higher scores than by fine-tuning the model (89.08 vs. 89.55). This demonstrates that although CamemBERT can be used successfully without any task-specific architecture, it can still produce high quality contextualized embeddings that might be useful in scenarios where powerful downstream architectures exist.

[11] $XLM_{MLM-TLM}$ is a lower-case model. Case is crucial for NER, therefore we do not report its low performance (84.37%)

Natural Language Inference On the XNLI benchmark, we compare CamemBERT to previous state-of-the-art multilingual models in the fine-tuning setting. In addition to the standard CamemBERT model with a BASE architecture, we train another model with the LARGE architecture, referred to as $\text{CamemBERT}_{\text{LARGE}}$, for a fair comparison with $\text{XLM-R}_{\text{LARGE}}$. This model is trained with the CCNet corpus, described in Sec. 6, for 100k steps.[12] We expect that training the model for longer would yield even better performance.

CamemBERT reaches higher accuracy than its BASE counterparts reaching +5.6% over mBERT, +2.3 over $\text{XLM}_{\text{MLM-TLM}}$, and +2.4 over $\text{XLM-R}_{\text{BASE}}$. CamemBERT also uses as few as half as many parameters (110M vs. 270M for $\text{XLM-R}_{\text{BASE}}$).

$\text{CamemBERT}_{\text{LARGE}}$ achieves a state-of-the-art accuracy of 85.7% on the XNLI benchmark, as opposed to 85.2, for the recent $\text{XLM-R}_{\text{LARGE}}$.

CamemBERT uses fewer parameters than multilingual models, mostly because of its smaller vocabulary size (e.g. 32k vs. 250k for XLM-R). Two elements might explain the better performance of CamemBERT over XLM-R. Even though XLM-R was trained on an impressive amount of data (2.5TB), only 57GB of this data is in French, whereas we used 138GB of French data. Additionally XLM-R also handles 100 languages, and the authors show that when reducing the number of languages to 7, they can reach 82.5% accuracy for French XNLI with their BASE architecture.

Summary of CamemBERT's results CamemBERT improves the state of the art for the 4 downstream tasks considered, thereby confirming on French the usefulness of Transformer-based models. We obtain these results when using CamemBERT as a fine-tuned model or when used as contextual embeddings with task-specific architectures. This questions the need for more complex downstream architectures, similar to what was shown for English (Devlin et al., 2019). Additionally, this suggests that CamemBERT is also able to produce high-quality representations out-of-the-box without further tuning.

[12] We train our LARGE model with the CCNet corpus for practical reasons. Given that BASE models reach similar performance when using OSCAR or CCNet as pretraining corpus (Appendix Table 8), we expect an OSCAR LARGE model to reach comparable scores.

6 Impact of corpus origin and size

In this section we investigate the influence of the homogeneity and size of the pretraining corpus on downstream task performance. With this aim, we train alternative version of CamemBERT by varying the pretraining datasets. For this experiment, we fix the number of pretraining steps to 100k, and allow the number of epochs to vary accordingly (more epochs for smaller dataset sizes). All models use the BASE architecture.

In order to investigate the need for homogeneous clean data versus more diverse and possibly noisier data, we use alternative sources of pretraining data in addition to OSCAR:

- **Wikipedia**, which is homogeneous in terms of genre and style. We use the official 2019 French Wikipedia dumps[13]. We remove HTML tags and tables using Giuseppe Attardi's *WikiExtractor*.[14]

- **CCNet** (Wenzek et al., 2019), a dataset extracted from Common Crawl with a different filtering process than for OSCAR. It was built using a language model trained on Wikipedia, in order to filter out bad quality texts such as code or tables.[15] As this filtering step biases the noisy data from Common Crawl to more Wikipedia-like text, we expect CCNet to act as a middle ground between the unfiltered "noisy" OSCAR dataset, and the "clean" Wikipedia dataset. As a result of the different filtering processes, CCNet contains longer documents on average compared to OSCAR with smaller—and often noisier—documents weeded out.

Table 6 summarizes statistics of these different corpora.

In order to make the comparison between these three sources of pretraining data, we randomly sample 4GB of text (at the document level) from OSCAR and CCNet, thereby creating samples of both Common-Crawl-based corpora of the same size as the French Wikipedia. These smaller 4GB samples also provides us a way to investigate the impact

[13] https://dumps.wikimedia.org/backup-index.html.

[14] https://github.com/attardi/wikiextractor.

[15] We use the HEAD split, which corresponds to the top 33% of documents in terms of filtering perplexity.

Dataset	Size	GSD		Sequoia		Spoken		ParTUT		Average		NER	NLI
		UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	F1	Acc.
Fine-tuning													
Wiki	4GB	98.28	93.04	98.74	92.71	96.61	79.61	96.20	89.67	97.45	88.75	89.86	78.32
CCNet	4GB	98.34	93.43	98.95	93.67	96.92	**82.09**	96.50	**90.98**	97.67	**90.04**	90.46	**82.06**
OSCAR	4GB	98.35	93.55	98.97	93.70	96.94	81.97	96.58	90.28	97.71	89.87	90.65	81.88
OSCAR	138GB	**98.39**	**93.80**	**98.99**	**94.00**	**97.17**	81.18	**96.63**	90.56	**97.79**	89.88	**91.55**	81.55
Embeddings (with UDPipe Future (tagging, parsing) or LSTM+CRF (NER))													
Wiki	4GB	98.09	92.31	98.74	93.55	96.24	78.91	95.78	89.79	97.21	88.64	91.23	-
CCNet	4GB	**98.22**	**92.93**	99.12	94.65	97.17	**82.61**	**96.74**	89.95	97.81	90.04	**92.30**	-
OSCAR	4GB	98.21	92.77	99.12	**94.92**	97.20	82.47	**96.74**	**90.05**	**97.82**	**90.05**	91.90	-
OSCAR	138GB	98.18	92.77	**99.14**	94.24	**97.26**	82.44	96.52	89.89	97.77	89.84	91.83	-

Table 5: Results on the four tasks using language models pre-trained on data sets of varying homogeneity and size, reported on validation sets (average of 4 runs for POS tagging, parsing and NER, average of 10 runs for NLI).

Corpus	Size	#tokens	#docs	Tokens/doc Percentiles:		
				5%	50%	95%
Wikipedia	4GB	990M	1.4M	102	363	2530
CCNet	135GB	31.9B	33.1M	128	414	2869
OSCAR	138GB	32.7B	59.4M	28	201	1946

Table 6: Statistics on the pretraining datasets used.

of pretraining data size. Downstream task performance for our alternative versions of CamemBERT are provided in Table 5. The upper section reports scores in the fine-tuning setting while the lower section reports scores for the embeddings.

6.1 Common Crawl vs. Wikipedia?

Table 5 clearly shows that models trained on the 4GB versions of OSCAR and CCNet (Common Crawl) perform consistently better than the the one trained on the French Wikipedia. This is true both in the fine-tuning and embeddings setting. Unsurprisingly, the gap is larger on tasks involving texts whose genre and style are more divergent from those of Wikipedia, such as tagging and parsing on the Spoken treebank. The performance gap is also very large on the XNLI task, probably as a consequence of the larger diversity of Common-Crawl-based corpora in terms of genres and topics. XNLI is indeed based on multiNLI which covers a range of genres of spoken and written text.

The downstream task performances of the models trained on the 4GB version of CCNet and OSCAR are much more similar.[16]

6.2 How much data do you need?

An unexpected outcome of our experiments is that the model trained "only" on the 4GB sample of OSCAR performs similarly to the standard CamemBERT trained on the whole 138GB OSCAR. The only task with a large performance gap is NER, where "138GB" models are better by 0.9 F1 points. This could be due to the higher number of named entities present in the larger corpora, which is beneficial for this task. On the contrary, other tasks don't seem to gain from the additional data.

In other words, when trained on corpora such as OSCAR and CCNet, which are heterogeneous in terms of genre and style, 4GB of uncompressed text is large enough as pretraining corpus to reach state-of-the-art results with the BASE architecure, better than those obtained with mBERT (pretrained on 60GB of text).[17] This calls into question the need to use a very large corpus such as OSCAR or CCNet when training a monolingual Transformer-based language model such as BERT or RoBERTa. Not only does this mean that the computational (and therefore environmental) cost of training a state-of-the-art language model can be reduced, but it also means that CamemBERT-like models can be trained for all languages for which a Common-Crawl-based corpus of 4GB or more can be created. OSCAR is available in 166 languages, and provides such a corpus for 38 languages. Moreover, it is possible that slightly smaller corpora (e.g. down to 1GB) could also prove sufficient to train high-performing language models. We obtained our results with BASE architectures. Further research is needed to confirm the validity of our findings on larger architectures and other more complex natural

[16] We provide the results of a model trained on the whole CCNet corpus in the Appendix. The conclusions are similar when comparing models trained on the full corpora: downstream results are similar when using OSCAR or CCNet.

[17] The OSCAR-4GB model gets slightly better XNLI accuracy than the full OSCAR-138GB model (81.88 vs. 81.55). This might be due to the random seed used for pretraining, as each model is pretrained only once.

language understanding tasks. However, even with a BASE architecture and 4GB of training data, the validation loss is still decreasing beyond 100k steps (and 400 epochs). This suggests that we are still under-fitting the 4GB pretraining dataset, training longer might increase downstream performance.

7 Discussion

Since the pre-publication of this work (Martin et al., 2019), many monolingual language models have appeared, e.g. (Le et al., 2019; Virtanen et al., 2019; Delobelle et al., 2020), for as much as 30 languages (Nozza et al., 2020). In almost all tested configurations they displayed better results than multilingual language models such as mBERT (Pires et al., 2019). Interestingly, Le et al. (2019) showed that using their FlauBert, a RoBERTa-based language model for French, which was trained on less but more edited data, in conjunction to CamemBERT in an ensemble system could improve the performance of a parsing model and establish a new state-of-the-art in constituency parsing of French, highlighting thus the complementarity of both models.[18] As it was the case for English when BERT was first released, the availability of similar scale language models for French enabled interesting applications, such as large scale anonymization of legal texts, where CamemBERT-based models established a new state-of-the-art on this task (Benesty, 2019), or the first large question answering experiments on a French Squad data set that was released very recently (d'Hoffschmidt et al., 2020) where the authors matched human performance using $\text{CamemBERT}_{\text{LARGE}}$. Being the first pre-trained language model that used the open-source Common Crawl Oscar corpus and given its impact on the community, CamemBERT paved the way for many works on monolingual language models that followed. Furthermore, the availability of all its training data favors reproducibility and is a step towards better understanding such models. In that spirit, we make the models used in our experiments available via our website and via the `huggingface` and `fairseq` APIs, in addition to the base CamemBERT model.

8 Conclusion

In this work, we investigated the feasibility of training a Transformer-based language model for languages other than English. Using French as an example, we trained CamemBERT, a language model based on RoBERTa. We evaluated CamemBERT on four downstream tasks (part-of-speech tagging, dependency parsing, named entity recognition and natural language inference) in which our best model reached or improved the state of the art in all tasks considered, even when compared to strong multilingual models such as mBERT, XLM and XLM-R, while also having fewer parameters.

Our experiments demonstrate that using web crawled data with high variability is preferable to using Wikipedia-based data. In addition we showed that our models could reach surprisingly high performances with as low as 4GB of pretraining data, questioning thus the need for large scale pretraining corpora. This shows that state-of-the-art Transformer-based language models can be trained on languages with far fewer resources than English, whenever a few gigabytes of data are available. This paves the way for the rise of monolingual contextual pre-trained language-models for under-resourced languages. The question of knowing whether pretraining on small domain specific content will be a better option than transfer learning techniques such as fine-tuning remains open and we leave it for future work.

Pretrained on pure open-source corpora, CamemBERT is freely available and distributed with the MIT license via popular NLP libraries (`fairseq` and `huggingface`) as well as on our website `camembert-model.fr`.

Acknowledgments

We want to thank Clémentine Fourrier for her proofreading and insightful comments, and Alix Chagué for her great logo. This work was partly funded by three French National funded projects granted to Inria and other partners by the Agence Nationale de la Recherche, namely projects PARSITI (ANR-16-CE33-0021), SoSweet (ANR-15-CE38-0011) and BASNUM (ANR-18-CE38-0003), as well as by the last author's chair in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

[18] We refer the reader to (Le et al., 2019) for a comprehensive benchmark and details therein.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. *Building a Treebank for French*, pages 165–187. Kluwer, Dordrecht.

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.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Rachel Bawden, Marie-Amélie Botalla, Kim Gerdes, and Sylvain Kahane. 2014. Correcting and validating syntactic dependency in the spoken French treebank rhapsodie. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2320–2325, Reykjavik, Iceland. European Language Resources Association (ELRA).

Michaël Benesty. 2019. Ner algo benchmark: spacy, flair, m-bert and camembert on anonymizing french commercial legal cases.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Marie Candito and Benoit Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proc. of IWPT'09*, Paris, France.

Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, and Éric Villemonte de la Clergerie. 2014. Deep syntax annotation of the sequoia french treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 2298–2305. European Language Resources Association (ELRA).

Marie Candito and Djamé Seddah. 2012. Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN, Grenoble, France, June 4-8, 2012*, pages 321–334.

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. ArXiv preprint : 1911.02116.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. ArXiv preprint 2001.06286.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume*

1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Martin d'Hoffschmidt, Maxime Vidal, Wacim Belblidia, and Tom Brendlé. 2020. Fquad: French question answering dataset. *arXiv preprint arXiv:2002.06071*.

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*. OpenReview.net.

Yoann Dupont. 2017. Exploration de traits pour la reconnaissance d'entit'es nomm'ees du français par apprentissage automatique. In *24e Conf'erence sur le Traitement Automatique des Langues Naturelles (TALN)*, page 42.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In (Korhonen et al., 2019), pages 3651–3657.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

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

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

Daniel Kondratyuk. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *CoRR*, abs/1904.02099.

Anna Korhonen, David R. Traum, and Lluís Màrquez, editors. 2019. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Anne Lacheret, Sylvain Kahane, Julie Beliao, Anne Dister, Kim Gerdes, Jean-Philippe Goldman, Nicolas Obin, Paola Pietrandrea, and Atanas Tchobanov. 2014. Rhapsodie: a prosodic-syntactic treebank for spoken French. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 295–301, Reykjavik, Iceland. European Language Resources Association (ELRA).

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

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 preprint 1909.11942.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2019. Flaubert: Unsupervised language model pre-training for french. ArXiv : 1912.05372.

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. ArXiv preprint 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*. ArXiv preprint : 1911.03894.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. 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, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

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, Kyung-Tae 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.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific BERT models. *CoRR*, abs/2003.02912.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Matthew E. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 2227–2237. 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).

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? arXiv:1906.01502.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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 preprint 1910.10683.

Benoît Sagot, Marion Richard, and Rosa Stern. 2012. Annotation référentielle du corpus arboré de Paris 7 en entités nommées (referential named entity annotation of the paris 7 french treebank) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN, Grenoble, France, June 4-8, 2012*, pages 535–542. ATALA/AFCP.

Manuela Sanguinetti and Cristina Bosco. 2015. PartTUT: The Turin University Parallel Treebank. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, volume 589 of *Studies in Computational Intelligence*, pages 51–69. Springer.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Amit Seker, Amir More, and Reut Tsarfaty. 2018. Universal morpho-syntactic parsing and the contribution of lexica: Analyzing the onlp lab submission to the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 208–215.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer 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.

Milan Straka, Jana Straková, and Jan Hajic. 2019. Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing. ArXiv preprint 1908.07448.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In (Korhonen et al., 2019), pages 5326–5331.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip

Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. ArXiv preprint 1912.07076.

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 preprint 1911.00359.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv preprint 1910.03771.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. *CoRR*, abs/1904.09077.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Appendix

In the appendix, we analyse different design choices of CamemBERT (Table 8), namely with respect to the use of whole-word masking, the training dataset, the model size, and the number of training steps in complement with the analyses of the impact of corpus origin an size (Section 6. In all the ablations, all scores come from at least 4 averaged runs. For POS tagging and dependency parsing, we average the scores on the 4 treebanks. We also report all averaged test scores of our different models in Table 7.

A Impact of Whole-Word Masking

In Table 8, we compare models trained using the traditional subword masking with whole-word masking. Whole-Word Masking positively impacts downstream performances for NLI (although only by 0.5 points of accuracy). To our surprise, this Whole-Word Masking scheme does not benefit much lower level task such as Name Entity Recognition, POS tagging and Dependency Parsing.

B Impact of model size

Table 8 compares models trained with the BASE and LARGE architectures. These models were trained with the CCNet corpus (135GB) for practical reasons. We confirm the positive influence of larger models on the NLI and NER tasks. The LARGE architecture leads to respectively 19.7% error reduction and 23.7%. To our surprise, on POS tagging and dependency parsing, having three time more parameters doesn't lead to a significant difference compared to the BASE model. Tenney et al. (2019) and Jawahar et al. (2019) have shown that low-level syntactic capabilities are learnt in lower layers of BERT while higher level semantic representations are found in upper layers of BERT. POS tagging and dependency parsing probably do not benefit from adding more layers as the lower layers of the BASE architecture already capture what is necessary to complete these tasks.

C Impact of training dataset

Table 8 compares models trained on CCNet and on OSCAR. The major difference between the two datasets is the additional filtering step of CCNet that favors Wikipedia-Like texts. The model pretrained on OSCAR gets slightly better results on POS tagging and dependency parsing, but gets a larger +1.31 improvement on NER. The CCNet model gets better performance on NLI (+0.67).

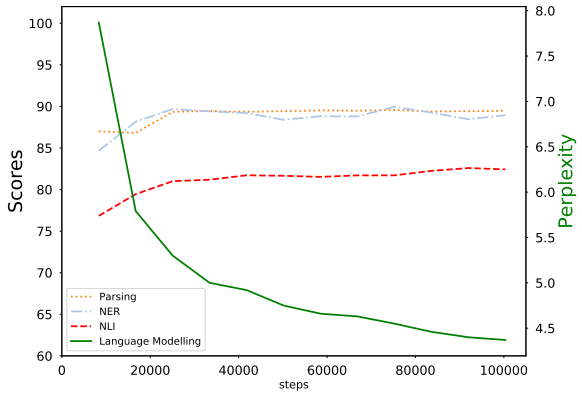

Figure 1: Impact of number of pretraining steps on downstream performance for CamemBERT.

.

D Impact of number of steps

Figure 1 displays the evolution of downstream task performance with respect to the number of steps. All scores in this section are averages from at least 4 runs with different random seeds. For POS tagging and dependency parsing, we also average the scores on the 4 treebanks.

We evaluate our model at every epoch (1 epoch equals 8360 steps). We report the masked language modelling perplexity along with downstream performances. Figure 1, suggests that the more complex the task the more impactful the number of steps is. We observe an early plateau for dependency parsing and NER at around 22k steps, while for NLI, even if the marginal improvement with regard to pretraining steps becomes smaller, the performance is still slowly increasing at 100k steps.

In Table 8, we compare two models trained on CCNet, one for 100k steps and the other for 500k steps to evaluate the influence of the total number of steps. The model trained for 500k steps does not increase the scores much from just training for 100k steps in POS tagging and parsing. The increase is slightly higher for XNLI (+0.84).

Those results suggest that low level syntactic representation are captured early in the language model training process while it needs more steps to extract complex semantic information as needed for NLI.

Dataset	Masking	Arch.	#Steps	GSD		Sequoia		Spoken		ParTUT		NER	NLI
				UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	F1	Acc.
Fine-tuning													
OSCAR	Subword	Base	100k	**98.25**	92.29	99.25	93.70	96.95	79.96	97.73	**92.68**	89.23	81.18
OSCAR	Whole-word	Base	100k	98.21	92.30	99.21	94.33	96.97	80.16	**97.78**	92.65	89.11	81.92
CCNET	Subword	Base	100k	98.02	92.06	**99.26**	94.13	96.94	80.39	97.55	92.66	89.05	81.77
CCNET	Whole-word	Base	100k	98.03	**92.43**	99.18	94.26	96.98	80.89	97.46	92.33	89.27	81.92
CCNET	Whole-word	Base	500k	98.21	**92.43**	99.24	**94.60**	96.69	**80.97**	97.65	92.48	89.08	83.43
CCNET	Whole-word	Large	100k	98.01	91.09	99.23	93.65	**97.01**	80.89	97.41	92.59	**89.39**	**85.29**
Embeddings (with UDPipe Future (tagging, parsing) or LSTM+CRF (NER))													
OSCAR	Subword	Base	100k	**98.01**	90.64	**99.27**	94.26	97.15	**82.56**	**97.70**	92.70	**90.25**	-
OSCAR	Whole-word	Base	100k	97.97	90.44	99.23	93.93	97.08	81.74	97.50	92.28	89.48	-
CCNET	Subword	Base	100k	97.87	**90.78**	99.20	94.33	**97.17**	82.39	97.54	92.51	89.38	-
CCNET	Whole-word	Base	100k	97.96	90.76	99.23	**94.34**	97.04	82.09	97.39	**92.82**	89.85	-
CCNET	Whole-word	Base	500k	97.84	90.25	99.14	93.96	97.01	82.17	97.27	92.28	89.07	-
CCNET	Whole-word	Large	100k	**98.01**	90.70	99.23	94.01	97.04	82.18	97.31	92.28	88.76	-

Table 7: Performance reported on **Test sets** for all trained models (**average** over multiple fine-tuning seeds).

Dataset	Masking	Arch.	#Param.	#Steps	UPOS	LAS	NER	XNLI
Masking Strategy								
OSCAR	Subword	Base	110M	100k	97.78	89.80	**91.55**	81.04
OSCAR	Whole-word	Base	110M	100k	**97.79**	**89.88**	91.44	**81.55**
Model Size								
CCNet	Whole-word	Base	110M	100k	97.67	89.46	90.13	82.22
CCNet	Whole-word	Large	335M	100k	**97.74**	**89.82**	**92.47**	**85.73**
Dataset								
CCNet	Whole-word	Base	110M	100k	97.67	89.46	90.13	**82.22**
OSCAR	Whole-word	Base	110M	100k	**97.79**	**89.88**	**91.44**	81.55
Number of Steps								
CCNet	Whole-word	Base	110M	100k	**98.04**	89.85	90.13	82.20
CCNet	Whole-word	Base	110M	500k	97.95	**90.12**	91.30	**83.04**

Table 8: Comparing scores on the **Validation sets** of different design choices. POS tagging and parsing datasets are averaged. (average over multiple fine-tuning seeds).