Unsupervised Cross-lingual Representation Learning at Scale

Alexis Conneau[*] **Kartikay Khandelwal**[*]

Naman Goyal   **Vishrav Chaudhary**   **Guillaume Wenzek**   **Francisco Guzmán**

Edouard Grave   **Myle Ott**   **Luke Zettlemoyer**   **Veselin Stoyanov**

Facebook AI

Abstract

This paper shows that pretraining multilingual language models at scale leads to significant performance gains for a wide range of cross-lingual transfer tasks. We train a Transformer-based masked language model on one hundred languages, using more than two terabytes of filtered CommonCrawl data. Our model, dubbed *XLM-R*, significantly outperforms multilingual BERT (mBERT) on a variety of cross-lingual benchmarks, including +13.8% average accuracy on XNLI, +12.3% average F1 score on MLQA, and +2.1% average F1 score on NER. *XLM-R* performs particularly well on low-resource languages, improving 11.8% in XNLI accuracy for Swahili and 9.2% for Urdu over the previous XLM model. We also present a detailed empirical evaluation of the key factors that are required to achieve these gains, including the trade-offs between (1) positive transfer and capacity dilution and (2) the performance of high and low resource languages at scale. Finally, we show, for the first time, the possibility of multilingual modeling without sacrificing per-language performance; *XLM-R* is very competitive with strong monolingual models on the GLUE and XNLI benchmarks. We will make *XLM-R* code, data, and models publicly available.[1]

1   Introduction

The goal of this paper is to improve cross-lingual language understanding (XLU), by carefully studying the effects of training unsupervised cross-lingual representations at a very large scale. We present *XLM-R*, a transformer-based multilingual masked language model pre-trained on text in 100 languages, which obtains state-of-the-art performance on cross-lingual classification, sequence labeling and question answering.

Multilingual masked language models (MLM) like mBERT (Devlin et al., 2018) and XLM (Lample and Conneau, 2019) have pushed the state-of-the-art on cross-lingual understanding tasks by jointly pretraining large Transformer models (Vaswani et al., 2017) on many languages. These models allow for effective cross-lingual transfer, as seen in a number of benchmarks including cross-lingual natural language inference (Bowman et al., 2015; Williams et al., 2017; Conneau et al., 2018), question answering (Rajpurkar et al., 2016; Lewis et al., 2019), and named entity recognition (Pires et al., 2019; Wu and Dredze, 2019). However, all of these studies pre-train on Wikipedia, which provides a relatively limited scale especially for lower resource languages.

In this paper, we first present a comprehensive analysis of the trade-offs and limitations of multilingual language models at scale, inspired by recent monolingual scaling efforts (Liu et al., 2019). We measure the trade-off between high-resource and low-resource languages and the impact of language sampling and vocabulary size. The experiments expose a trade-off as we scale the number of languages for a fixed model capacity: more languages leads to better cross-lingual performance on low-resource languages up until a point, after which the overall performance on monolingual and cross-lingual benchmarks degrades. We refer to this tradeoff as the *curse of multilinguality*, and show that it can be alleviated by simply increasing model capacity. We argue, however, that this remains an important limitation for future XLU systems which may aim to improve performance with more modest computational budgets.

Our best model XLM-RoBERTa (*XLM-R*) outperforms mBERT on cross-lingual classification by up to 21% accuracy on low-resource languages like Swahili and Urdu. It outperforms the previous state of the art by 3.9% average accuracy on XNLI, 2.1%

---

average F1-score on Named Entity Recognition, and 8.4% average F1-score on cross-lingual Question Answering. We also evaluate monolingual fine tuning on the GLUE and XNLI benchmarks, where *XLM-R* obtains results competitive with state-of-the-art monolingual models, including RoBERTa (Liu et al., 2019). These results demonstrate, for the first time, that it is possible to have a single large model for all languages, without sacrificing per-language performance. We will make our code, models and data publicly available, with the hope that this will help research in multilingual NLP and low-resource language understanding.

2 Related Work

From pretrained word embeddings (Mikolov et al., 2013b; Pennington et al., 2014) to pretrained contextualized representations (Peters et al., 2018; Schuster et al., 2019) and transformer based language models (Radford et al., 2018; Devlin et al., 2018), unsupervised representation learning has significantly improved the state of the art in natural language understanding. Parallel work on cross-lingual understanding (Mikolov et al., 2013a; Schuster et al., 2019; Lample and Conneau, 2019) extends these systems to more languages and to the cross-lingual setting in which a model is learned in one language and applied in other languages.

Most recently, Devlin et al. (2018) and Lample and Conneau (2019) introduced mBERT and XLM - masked language models trained on multiple languages, without any cross-lingual supervision. Lample and Conneau (2019) propose translation language modeling (TLM) as a way to leverage parallel data and obtain a new state of the art on the cross-lingual natural language inference (XNLI) benchmark (Conneau et al., 2018). They further show strong improvements on unsupervised machine translation and pretraining for sequence generation. Separately, Pires et al. (2019) demonstrated the effectiveness of multilingual models like mBERT on sequence labeling tasks. Huang et al. (2019) showed gains over XLM using cross-lingual multi-task learning, and Singh et al. (2019) demonstrated the efficiency of cross-lingual data augmentation for cross-lingual NLI. However, all of this work was at a relatively modest scale, in terms of the amount of training data, as compared to our approach.

The benefits of scaling language model pretraining by increasing the size of the model as well as the training data has been extensively studied in the literature. For the monolingual case, Jozefowicz et al. (2016) show how large-scale LSTM models can obtain much stronger performance on language modeling benchmarks when trained on billions of tokens. GPT (Radford et al., 2018) also highlights the importance of scaling the amount of data and RoBERTa (Liu et al., 2019) shows that training BERT longer on more data leads to significant boost in performance. Inspired by RoBERTa, we show that mBERT and XLM are undertuned, and that simple improvements in the learning procedure of unsupervised MLM leads to much better performance. We train on cleaned CommonCrawls (Wenzek et al., 2019), which increase the amount of data for low-resource languages by two orders of magnitude on average. Similar data has also been shown to be effective for learning high quality word embeddings in multiple languages (Grave et al., 2018).

Several efforts have trained massively multilingual machine translation models from large parallel corpora. They uncover the high and low resource trade-off and the problem of capacity dilution (Johnson et al., 2017; Tan et al., 2019). The work most similar to ours is Arivazhagan et al. (2019), which trains a single model in 103 languages on over 25 billion parallel sentences. Siddhant et al. (2019) further analyze the representations obtained by the encoder of a massively multilingual machine translation system and show that it obtains similar results to mBERT on cross-lingual NLI. Our work, in contrast, focuses on the unsupervised learning of cross-lingual representations and their transfer to discriminative tasks.

3 Model and Data

In this section, we present the training objective, languages, and data we use. We follow the XLM approach (Lample and Conneau, 2019) as closely as possible, only introducing changes that improve performance at scale.

Masked Language Models. We use a Transformer model (Vaswani et al., 2017) trained with the multilingual MLM objective (Devlin et al., 2018; Lample and Conneau, 2019) using only monolingual data. We sample streams of text from each language and train the model to predict the masked tokens in the input. We apply subword tokenization directly on raw text data using Sentence Piece (Kudo and Richardson, 2018) with a unigram language model (Kudo, 2018). We sample batches
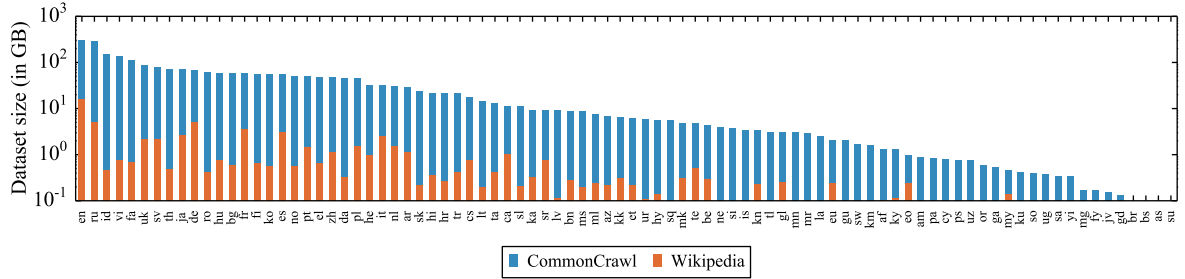
Figure 1: Amount of data in GiB (log-scale) for the 88 languages that appear in both the Wiki-100 corpus used for XLM-100, and the CC-100 used for XLM-R. CC-100 increases the amount of data by several orders of magnitude, in particular for low-resource languages.

from different languages using the same sampling distribution as Lample and Conneau (2019), but with $\alpha = 0.3$. Unlike Lample and Conneau (2019), we do not use language embeddings, which allows our model to better deal with code-switching. We use a large vocabulary size of 250K with a full softmax and train two different models: *XLM-R* $_{\text{Base}}$ (L = 12, H = 768, A = 12, 270M params) and *XLM-R* (L = 24, H = 1024, A = 16, 550M params). For all of our ablation studies, we use a BERT$_{\text{Base}}$ architecture with a vocabulary of 150K tokens. Appendix B goes into more details about the architecture of the different models referenced in this paper.

Scaling to a hundred languages. *XLM-R* is trained on 100 languages; we provide a full list of languages and associated statistics in Appendix A. Figure 1 specifies the iso codes of 88 languages that are shared across *XLM-R* and XLM-100, the model from Lample and Conneau (2019) trained on Wikipedia text in 100 languages.

Compared to previous work, we replace some languages with more commonly used ones such as romanized Hindi and traditional Chinese. In our ablation studies, we always include the 7 languages for which we have classification and sequence labeling evaluation benchmarks: English, French, German, Russian, Chinese, Swahili and Urdu. We chose this set as it covers a suitable range of language families and includes low-resource languages such as Swahili and Urdu. We also consider larger sets of 15, 30, 60 and all 100 languages. When reporting results on high-resource and low-resource, we refer to the average of English and French results, and the average of Swahili and Urdu results respectively.

Scaling the Amount of Training Data. Following Wenzek et al. (2019), we build a clean Com-

monCrawl Corpus in 100 languages. We use an internal language identification model in combination with the one from fastText (Joulin et al., 2017). We train language models in each language and use it to filter documents as described in Wenzek et al. (2019). We consider one CommonCrawl dump for English and twelve dumps for all other languages, which significantly increases dataset sizes, especially for low-resource languages like Burmese and Swahili.

Figure 1 shows the difference in size between the Wikipedia Corpus used by mBERT and XLM-100, and the CommonCrawl Corpus we use. As we show in Section 5.3, monolingual Wikipedia corpora are too small to enable unsupervised representation learning. Based on our experiments, we found that a few hundred MiB of text data is usually a minimal size for learning a BERT model.

4 Evaluation

We consider four evaluation benchmarks. For cross-lingual understanding, we use cross-lingual natural language inference, named entity recognition, and question answering. We use the GLUE benchmark to evaluate the English performance of *XLM-R* and compare it to other state-of-the-art models.

Cross-lingual Natural Language Inference (XNLI). The XNLI dataset comes with ground-truth dev and test sets in 15 languages, and a ground-truth English training set. The training set has been machine-translated to the remaining 14 languages, providing synthetic training data for these languages as well. We evaluate our model on cross-lingual transfer from English to other languages. We also consider three machine translation baselines: (i) *translate-test*: dev and test sets are machine-translated to English and a single English model is used (ii) *translate-train* (per-language):

the English training set is machine-translated to each language and we fine-tune a multiligual model on each training set (iii) *translate-train-all* (multi-language): we fine-tune a multilingual model on the concatenation of all training sets from translate-train. For the translations, we use the official data provided by the XNLI project.

Named Entity Recognition. For NER, we consider the CoNLL-2002 (Sang, 2002) and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) datasets in English, Dutch, Spanish and German. We fine-tune multilingual models either (1) on the English set to evaluate cross-lingual transfer, (2) on each set to evaluate per-language performance, or (3) on all sets to evaluate multilingual learning. We report the F1 score, and compare to baselines from Lample et al. (2016) and Akbik et al. (2018).

Cross-lingual Question Answering. We use the MLQA benchmark from Lewis et al. (2019), which extends the English SQuAD benchmark to Spanish, German, Arabic, Hindi, Vietnamese and Chinese. We report the F1 score as well as the exact match (EM) score for cross-lingual transfer from English.

GLUE Benchmark. Finally, we evaluate the English performance of our model on the GLUE benchmark (Wang et al., 2018) which gathers multiple classification tasks, such as MNLI (Williams et al., 2017), SST-2 (Socher et al., 2013), or QNLI (Rajpurkar et al., 2018). We use BERT$_{Large}$ and RoBERTa as baselines.

5 Analysis and Results

In this section, we perform a comprehensive analysis of multilingual masked language models. We conduct most of the analysis on XNLI, which we found to be representative of our findings on other tasks. We then present the results of *XLM-R* on cross-lingual understanding and GLUE. Finally, we compare multilingual and monolingual models, and present results on low-resource languages.

5.1 Improving and Understanding Multilingual Masked Language Models

Much of the work done on understanding the cross-lingual effectiveness of mBERT or XLM (Pires et al., 2019; Wu and Dredze, 2019; Lewis et al., 2019) has focused on analyzing the performance of fixed pretrained models on downstream tasks. In this section, we present a comprehensive study of different factors that are important to *pretraining*

large scale multilingual models. We highlight the trade-offs and limitations of these models as we scale to one hundred languages.

Transfer-dilution trade-off and Curse of Multilinguality. Model capacity (i.e. the number of parameters in the model) is constrained due to practical considerations such as memory and speed during training and inference. For a fixed sized model, the per-language capacity decreases as we increase the number of languages. While low-resource language performance can be improved by adding similar higher-resource languages during pretraining, the overall downstream performance suffers from this capacity dilution (Arivazhagan et al., 2019). Positive transfer and capacity dilution have to be traded off against each other.

We illustrate this trade-off in Figure 2, which shows XNLI performance vs the number of languages the model is pretrained on. Initially, as we go from 7 to 15 languages, the model is able to take advantage of positive transfer and this improves performance, especially on low resource languages. Beyond this point the *curse of multilinguality* kicks in and degrades performance across all languages. Specifically, the overall XNLI accuracy decreases from 71.8% to 67.7% as we go from XLM-7 to XLM-100. The same trend can be observed for models trained on the larger CommonCrawl Corpus.

The issue is even more prominent when the capacity of the model is small. To show this, we pretrain models on Wikipedia Data in 7, 30 and 100 languages. As we add more languages, we make the Transformer wider by increasing the hidden size from 768 to 960 to 1152. In Figure 4, we show that the added capacity allows XLM-30 to be on par with XLM-7, thus overcoming the curse of multilinguality. The added capacity for XLM-100, however, is not enough and it still lags behind due to higher vocabulary dilution (recall from Section 3 that we used a fixed vocabulary size of 150K for all models).

High-resource/Low-resource trade-off. The allocation of the model capacity across languages is controlled by several parameters: the training set size, the size of the shared subword vocabulary, and the rate at which we sample training examples from each language. We study the effect of sampling on the performance of high-resource (English and French) and low-resource (Swahili and
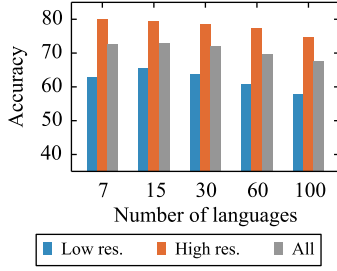
Figure 2: The transfer-interference trade-off: Low-resource languages benefit from scaling to more languages, until dilution (interference) kicks in and degrades overall performance.
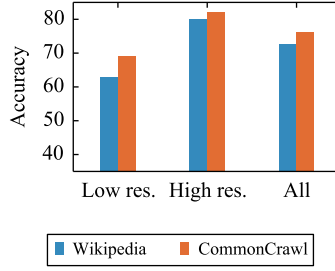

Figure 3: Wikipedia versus CommonCrawl: An XLM-7 obtains much better performance when trained on CC, in particular on low-resource languages.
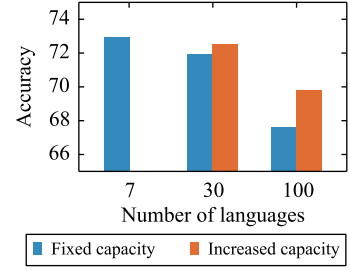

Figure 4: Adding more capacity to the model alleviates the curse of multilinguality, but remains an issue for models of reasonable size.
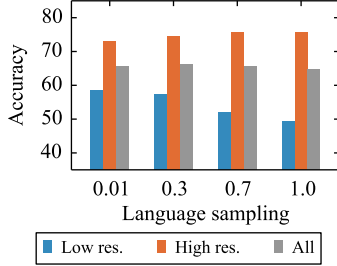

Figure 5: On the high-resource versus low-resource trade-off: impact of batch language sampling for XLM-100.
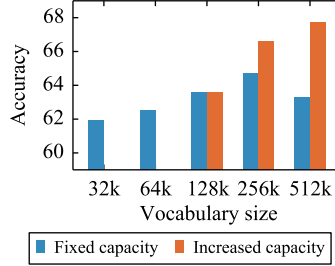

Figure 6: On the impact of vocabulary size at fixed capacity and with increasing capacity for XLM-100.
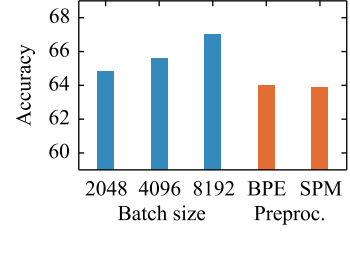

Figure 7: On the impact of large-scale training, and preprocessing simplification from BPE with tokenization to SPM on raw text data.

Urdu) languages for an XLM-100 model trained on Wikipedia (we observe a similar trend for the construction of the subword vocab). Specifically, we investigate the impact of varying the α parameter which controls the exponential smoothing of the language sampling rate. Similar to Lample and Conneau (2019), we use a sampling rate proportional to the number of sentences in each corpus. Models trained with higher values of α see batches of high-resource languages more often. Figure 5 shows that the higher the value of α, the better the performance on high-resource languages, and vice-versa. When considering overall performance, we found 0.3 to be an optimal value for α, and use this for *XLM-R*.

Importance of Capacity and Vocabulary Size. In previous sections and in Figure 4, we showed the importance of scaling the model size as we increase the number of languages. Similar to the overall model size, we argue that scaling the size of the shared vocabulary (the vocabulary capacity) can improve the performance of multilingual models on downstream tasks. To illustrate this effect, we train XLM-100 models on Wikipedia data with

different vocabulary sizes. We keep the overall number of parameters constant by adjusting the width of the transformer. Figure 6 shows that even with a fixed capacity, we observe a 2.8% increase in XNLI average accuracy as we increase the vocabulary size from 32K to 256K. This suggests that multilingual models can benefit from allocating a higher proportion of the total number of parameters to the embedding layer even though this reduces the size of the Transformer. With bigger models, we believe that using a vocabulary of up to 2 million tokens with an adaptive softmax (Grave et al., 2017; Baevski and Auli, 2018) should improve performance even further, but we leave this exploration to future work. For simplicity and given the computational constraints, we use a vocabulary of 250k for *XLM-R*.

We further illustrate the importance of this parameter, by training three models with the same transformer architecture (BERT$_{\text{Base}}$) but with different vocabulary sizes: 128K, 256K and 512K. We observe more than 3% gains in overall accuracy on XNLI by simply increasing the vocab size from 128k to 512k.

Importance of large-scale training with more data. As shown in Figure 1, the Common-Crawl Corpus that we collected has significantly more monolingual data than the previously used Wikipedia corpora. Figure 3 shows that for the same BERT$_{Base}$ architecture, all models trained on CommonCrawl obtain significantly better performance.

Apart from scaling the training data, Liu et al. (2019) also showed the benefits of training MLMs longer. In our experiments, we observed similar effects of large-scale training, such as increasing batch size (see Figure 7) and training time, on model performance. Specifically, we found that using validation perplexity as a stopping criterion for pretraining caused the multilingual MLM in Lample and Conneau (2019) to be under-tuned. In our experience, performance on downstream tasks continues to improve even after validation perplexity has plateaued. Combining this observation with our implementation of the unsupervised XLM-MLM objective, we were able to improve the performance of Lample and Conneau (2019) from 71.3% to more than 75% average accuracy on XNLI, which was on par with their supervised translation language modeling (TLM) objective. Based on these results, and given our focus on unsupervised learning, we decided to not use the supervised TLM objective for training our models.

Simplifying multilingual tokenization with Sentence Piece. The different language-specific tokenization tools used by mBERT and XLM-100 make these models more difficult to use on raw text. Instead, we train a Sentence Piece model (SPM) and apply it directly on raw text data for all languages. We did not observe any loss in performance for models trained with SPM when compared to models trained with language-specific preprocessing and byte-pair encoding (see Figure 7) and hence use SPM for *XLM-R*.

5.2 Cross-lingual Understanding Results

Based on these results, we adapt the setting of Lample and Conneau (2019) and use a large Transformer model with 24 layers and 1024 hidden states, with a 250k vocabulary. We use the multilingual MLM loss and train our *XLM-R* model for 1.5 Million updates on five hundred 32GB Nvidia V100 GPUs with a batch size of 8192. We leverage the SPM-preprocessed text data from Common-Crawl in 100 languages and sample languages with

$\alpha = 0.3$. In this section, we show that it outperforms all previous techniques on cross-lingual benchmarks while getting performance on par with RoBERTa on the GLUE benchmark.

XNLI. Table 1 shows XNLI results and adds some additional details: (i) the number of models the approach induces (#M), (ii) the data on which the model was trained (D), and (iii) the number of languages the model was pretrained on (#lg). As we show in our results, these parameters significantly impact performance. Column #M specifies whether model selection was done separately on the dev set of each language (N models), or on the joint dev set of all the languages (single model). We observe a 0.6 decrease in overall accuracy when we go from N models to a single model - going from 71.3 to 70.7. We encourage the community to adopt this setting. For cross-lingual transfer, while this approach is not fully zero-shot transfer, we argue that in real applications, a small amount of supervised data is often available for validation in each language.

XLM-R sets a new state of the art on XNLI. On cross-lingual transfer, *XLM-R* obtains 80.1% accuracy, outperforming the XLM-100 and mBERT open-source models by 9.4% and 13.8% average accuracy. On the Swahili and Urdu low-resource languages, *XLM-R* outperforms XLM-100 by 13.8% and 9.3%, and mBERT by 21.6% and 13.7%. While *XLM-R* handles 100 languages, we also show that it outperforms the former state of the art Unicoder (Huang et al., 2019) and XLM (MLM+TLM), which handle only 15 languages, by 4.7% and 5% average accuracy respectively. Using the multilingual training of translate-train-all, *XLM-R* further improves performance and reaches 82.4% accuracy, a new overall state of the art for XNLI, outperforming Unicoder by 3.9%. Multilingual training is similar to practical applications where training sets are available in various languages for the same task. In the case of XNLI, datasets have been translated, and translate-train-all can be seen as some form of cross-lingual data augmentation (Singh et al., 2019), similar to back-translation (Xie et al., 2019).

Named Entity Recognition. In Table 2, we report results of *XLM-R* and mBERT on CoNLL-2002 and CoNLL-2003. We consider the LSTM + CRF approach from Lample et al. (2016) and the Flair model from Akbik et al. (2018) as baselines. We evaluate the performance of the model

Model	D	#M	#lg	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
Fine-tune multilingual model on English training set (Cross-lingual Transfer)																			
Lample and Conneau (2019)	Wiki+MT	N	15	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
Huang et al. (2019)	Wiki+MT	N	15	85.1	79.0	79.4	77.8	77.2	77.2	76.3	72.8	73.5	76.4	73.6	76.2	69.4	69.7	66.7	75.4
Devlin et al. (2018)	Wiki	N	102	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
Lample and Conneau (2019)	Wiki	N	100	83.7	76.2	76.6	73.7	72.4	73.0	72.1	68.1	68.4	72.0	68.2	71.5	64.5	58.0	62.4	71.3
Lample and Conneau (2019)	Wiki	1	100	83.2	76.7	77.7	74.0	72.7	74.1	72.7	68.7	68.6	72.9	68.9	72.5	65.6	58.2	62.4	70.7
XLM-R$_{Base}$	CC	1	100	84.6	78.4	78.9	76.8	75.9	77.3	75.4	73.2	71.5	75.4	72.5	74.9	71.1	65.2	66.5	74.5
XLM-R	CC	1	100	88.8	83.6	84.2	82.7	82.3	83.1	80.1	79.0	78.8	79.7	78.6	80.2	75.8	72.0	71.7	80.1
Translate everything to English and use English-only model (TRANSLATE-TEST)																			
BERT-en	Wiki	1	1	88.8	81.4	82.3	80.1	80.3	80.9	76.2	76.0	75.4	72.0	71.9	75.6	70.0	65.8	65.8	76.2
Roberta	CC	1	1	_91.3_	82.9	84.3	81.2	81.7	83.1	78.3	76.8	76.6	74.2	74.1	77.5	70.9	66.7	66.8	77.8
Fine-tune multilingual model on each training set (TRANSLATE-TRAIN)																			
Lample and Conneau (2019)	Wiki	N	100	82.9	77.6	77.9	77.9	77.1	75.7	75.5	72.6	71.2	75.8	73.1	76.2	70.4	66.5	62.4	74.2
Fine-tune multilingual model on all training sets (TRANSLATE-TRAIN-ALL)																			
Lample and Conneau (2019)†	Wiki+MT	1	15	85.0	80.8	81.3	80.3	79.1	80.9	78.3	75.6	77.6	78.5	76.0	79.5	72.9	72.8	68.5	77.8
Huang et al. (2019)	Wiki+MT	1	15	85.6	81.1	82.3	80.9	79.5	81.4	79.7	76.8	78.2	77.9	77.1	80.5	73.4	73.8	69.6	78.5
Lample and Conneau (2019)	Wiki	1	100	84.5	80.1	81.3	79.3	78.6	79.4	77.5	75.2	75.6	78.3	75.7	78.3	72.1	69.2	67.7	76.9
XLM-R$_{Base}$	CC	1	100	84.3	80.1	81.2	79.8	79.2	80.6	78.1	77.0	75.9	79.5	77.2	79.6	74.9	70.2	70.8	77.9
XLM-R	CC	1	100	**88.7**	_85.2_	_85.6_	_84.6_	_83.6_	_85.5_	_82.4_	_81.6_	_80.9_	_83.4_	_80.9_	_83.3_	_79.8_	_75.9_	_74.3_	_82.4_

Table 1: **Results on cross-lingual classification.** We report the accuracy on each of the 15 XNLI languages and the average accuracy. We specify the dataset D used for pretraining, the number of models #M the approach requires and the number of languages #lg the model handles. Our *XLM-R* results are based on 5 different runs with different seeds. We show that using the translate-train-all approach which leverages training sets from multiple languages, *XLM-R* obtains a new state of the art on XNLI of 82.0% average accuracy. It also outperforms previous methods on cross-lingual transfer.

Model	train	#M	en	nl	es	de	Avg
Lample et al. (2016)	each	N	90.74	81.74	85.75	78.76	84.25
Akbik et al. (2018)	each	N	**93.18**	90.44	-	**88.27**	-
mBERT†	each	N	91.97	90.94	87.38	82.82	88.28
	en	1	91.97	77.57	74.96	69.56	78.52
XLM-R$_{Base}$	each	N	91.95	91.21	88.46	83.65	88.82
	en	1	91.95	77.83	76.24	69.70	78.93
	all	1	91.84	88.13	87.02	82.76	87.44
XLM-R	each	N	92.74	**93.25**	**89.04**	85.53	90.14
	en	1	92.74	81.00	76.44	72.27	80.61
	all	1	93.03	90.41	87.83	85.46	89.18

Table 2: **Results on named entity recognition** on CoNLL-2002 and CoNLL-2003 (F1 score). Results with † are from Wu and Dredze (2019). Note that mBERT and *XLM-R* do not use a linear-chain CRF, as opposed to Akbik et al. (2018) and Lample and Conneau (2019).

on each of the target languages in three different settings: (i) train on English data only (en) (ii) train on data in target language (each) (iii) train on data in all languages (all). Results of mBERT are reported from Wu and Dredze (2019). Note that we do not use a linear-chain CRF on top of *XLM-R* and mBERT representations, which gives an advantage to Akbik et al. (2018). Without the CRF, our *XLM-R* model still performs on par with the state of the art, outperforming Akbik et al. (2018) on Dutch by 2.84 points. On this task, *XLM-R* also outperforms mBERT by 2.1 F1 on average for cross-lingual transfer, and 1.86 F1 when trained on each language. Training on all languages leads

to an average F1 score of 89.18%, outperforming cross-lingual transfer approach by more than 8.5%.

Question Answering. We also obtain new state of the art results on the MLQA cross-lingual question answering benchmark, introduced by Lewis et al. (2019). We follow their procedure by training on the English training data and evaluating on the 7 languages of the dataset. We report results in Table 3. *XLM-R* obtains F1 and accuracy scores of 70.0% and 52.2% while the previous state of the art was 61.6% and 43.5%. *XLM-R* also outperforms mBERT by 12.3% F1-score and 10.6% accuracy. It even outperforms BERT-Large on English, confirming its strong monolingual performance.

5.3 Multilingual versus Monolingual

In this section, we present results of multilingual XLM models against monolingual BERT models.

GLUE: *XLM-R* versus RoBERTa. Our goal is to obtain a multilingual model with strong performance on both, cross-lingual understanding tasks as well as natural language understanding tasks for each language. To that end, we evaluate *XLM-R* on the GLUE benchmark. We show in Table 4, that *XLM-R* obtains better average dev performance than BERT$_{Large}$ by 1.3% and reaches performance on par with XLNet$_{Large}$. The RoBERTa model outperforms *XLM-R* by only 1.3% on average. We

Model	train	#lgs	en	es	de	ar	hi	vi	zh	Avg
BERT-Large[†]	en	1	80.2 / 67.4	-	-	-	-	-	-	-
mBERT[†]	en	102	77.7 / 65.2	64.3 / 46.6	57.9 / 44.3	45.7 / 29.8	43.8 / 29.7	57.1 / 38.6	57.5 / 37.3	57.7 / 41.6
XLM-15[†]	en	15	74.9 / 62.4	68.0 / 49.8	62.2 / 47.6	54.8 / 36.3	48.8 / 27.3	61.4 / 41.8	61.1 / 39.6	61.6 / 43.5
XLM-R$_{Base}$	en	100	77.8 / 65.3	67.2 / 49.7	60.8 / 47.1	53.0 / 34.7	57.9 / 41.7	63.1 / 43.1	60.2 / 38.0	62.9 / 45.7
XLM-R	en	100	**80.1 / 67.7**	**73.2 / 55.1**	**68.3 / 53.7**	**62.8 / 43.7**	**68.3 / 51.0**	**70.5 / 50.1**	**67.1 / 44.4**	**70.0 / 52.2**

Table 3: **Results on MLQA question answering** We report the F1 and EM (exact match) scores for zero-shot classification where models are fine-tuned on the English Squad dataset and evaluated on the 7 languages of MLQA. Results with † are taken from the original MLQA paper Lewis et al. (2019).

believe future work can reduce this gap even further by alleviating the curse of multilinguality and vocabulary dilution. These results demonstrate the possibility of learning one model for many languages while maintaining strong performance on per-language downstream tasks.

Model	#lgs	MNLI-m/mm	QNLI	QQP	SST	MRPC	STS-B	Avg
BERT$_{Large}$[†]	1	86.6/-	92.3	91.3	93.2	88.0	90.0	90.2
XLNet$_{Large}$[†]	1	89.8/-	93.9	91.8	95.6	89.2	91.8	92.0
RoBERTa[†]	1	90.2/90.2	94.7	92.2	96.4	90.9	92.4	92.8
XLM-R	100	88.4/88.5	93.1	92.2	95.1	89.7	90.4	91.5

Table 4: **GLUE dev results.** Results with † are from Liu et al. (2019). We compare the performance of *XLM-R* to BERT$_{Large}$, XLNet and Roberta on the English GLUE benchmark.

XNLI: XLM versus BERT. A recurrent criticism against multilingual model is that they obtain worse performance than their monolingual counterparts. In addition to the comparison of *XLM-R* and RoBERTa, we provide the first comprehensive study to assess this claim on the XNLI benchmark. We extend our comparison between multilingual XLM models and monolingual BERT models on 7 languages and compare performance in Table 5. We train 14 monolingual BERT models on Wikipedia and CommonCrawl[2], and two XLM-7 models. We add slightly more capacity in the vocabulary size of the multilingual model for a better comparison. To our surprise - and backed by further study on internal benchmarks - we found that *multilingual models can outperform their monolingual BERT counterparts*. Specifically, in Table 5, we show that for cross-lingual transfer, monolingual baselines outperform XLM-7 for both Wikipedia and CC by 1.6% and 1.3% average accuracy. However, by making use of multilingual training (translate-train-all) and leveraging training sets coming from multiple languages, XLM-7 can outperform the BERT models: our XLM-7 trained

on CC obtains 80.0% average accuracy on the 7 languages, while the average performance of monolingual BERT models trained on CC is 77.5%. This is a surprising result that shows that the capacity of multilingual models to leverage training data coming from multiple languages for a particular task can overcome the capacity dilution problem to obtain better overall performance.

Model	D	#vocab	en	fr	de	ru	zh	sw	ur	Avg
Monolingual baselines										
BERT	Wiki	40k	84.5	78.6	80.0	75.5	77.7	60.1	57.3	73.4
	CC	40k	86.7	81.2	81.2	78.2	79.5	70.8	65.1	77.5
Multilingual models (cross-lingual transfer)										
XLM-7	Wiki	150k	82.3	76.8	74.7	72.5	73.1	60.8	62.3	71.8
	CC	150k	85.7	78.6	79.5	76.4	74.8	71.2	66.9	76.2
Multilingual models (translate-train-all)										
XLM-7	Wiki	150k	84.6	80.1	80.2	75.7	78	68.7	66.7	76.3
	CC	150k	**87.2**	**82.5**	**82.9**	**79.7**	**80.4**	**75.7**	**71.5**	**80.0**

Table 5: **Multilingual versus monolingual models (BERT-BASE).** We compare the performance of monolingual models (BERT) versus multilingual models (XLM) on seven languages, using a BERT-BASE architecture. We choose a vocabulary size of 40k and 150k for monolingual and multilingual models.

5.4 Representation Learning for Low-resource Languages

We observed in Table 5 that pretraining on Wikipedia for Swahili and Urdu performed similarly to a randomly initialized model; most likely due to the small size of the data for these languages. On the other hand, pretraining on CC improved performance by up to 10 points. This confirms our assumption that mBERT and XLM-100 rely heavily on cross-lingual transfer but do not model the low-resource languages as well as *XLM-R*. Specifically, in the translate-train-all setting, we observe that the biggest gains for XLM models trained on CC, compared to their Wikipedia counterparts, are on low-resource languages; 7% and 4.8% improvement on Swahili and Urdu respectively.

---

[2]For simplicity, we use a reduced version of our corpus by capping the size of each monolingual dataset to 60 GiB.

6 Conclusion

In this work, we introduced *XLM-R*, our new state of the art multilingual masked language model trained on 2.5 TB of newly created clean CommonCrawl data in 100 languages. We show that it provides strong gains over previous multilingual models like mBERT and XLM on classification, sequence labeling and question answering. We exposed the limitations of multilingual MLMs, in particular by uncovering the high-resource versus low-resource trade-off, the curse of multilinguality and the importance of key hyperparameters. We also expose the surprising effectiveness of multilingual models over monolingual models, and show strong improvements on low-resource languages.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Alexei Baevski and Michael Auli. 2018. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

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*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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)*.

Edouard Grave, Armand Joulin, Moustapha Cissé, Hervé Jégou, et al. 2017. Efficient softmax approximation for gpus. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1302–1310. JMLR. org.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pretraining with multiple cross-lingual tasks. *arXiv preprint arXiv:1909.00964*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Googles multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. 2017. Bag of tricks for efficient text classification. *EACL 2017*, page 427.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

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 (Volume 1: Long Papers)*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

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 arXiv:1907.11692*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *ACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf*.

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 preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Erik F Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *arXiv preprint cs/0209010*.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*.

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Arivazhagan, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2019. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. *arXiv preprint arXiv:1909.00437*.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. Xlda: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. 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*, pages 6000–6010.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of the 2nd Workshop on Evaluating Vector-Space Representations for NLP*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Appendix

A   Languages and statistics for CC-100 used by *XLM-R*

In this section we present the list of languages in the CC-100 corpus we created for training *XLM-R*. We also report statistics such as the number of tokens and the size of each monolingual corpus.

ISO code	Language	Tokens (M)	Size (GiB)	ISO code	Language	Tokens (M)	Size (GiB)
af	Afrikaans	242	1.3	lo	Lao	17	0.6
am	Amharic	68	0.8	lt	Lithuanian	1835	13.7
ar	Arabic	2869	28.0	lv	Latvian	1198	8.8
as	Assamese	5	0.1	mg	Malagasy	25	0.2
az	Azerbaijani	783	6.5	mk	Macedonian	449	4.8
be	Belarusian	362	4.3	ml	Malayalam	313	7.6
bg	Bulgarian	5487	57.5	mn	Mongolian	248	3.0
bn	Bengali	525	8.4	mr	Marathi	175	2.8
-	Bengali Romanized	77	0.5	ms	Malay	1318	8.5
br	Breton	16	0.1	my	Burmese	15	0.4
bs	Bosnian	14	0.1	my	Burmese	56	1.6
ca	Catalan	1752	10.1	ne	Nepali	237	3.8
cs	Czech	2498	16.3	nl	Dutch	5025	29.3
cy	Welsh	141	0.8	no	Norwegian	8494	49.0
da	Danish	7823	45.6	om	Oromo	8	0.1
de	German	10297	66.6	or	Oriya	36	0.6
el	Greek	4285	46.9	pa	Punjabi	68	0.8
en	English	55608	300.8	pl	Polish	6490	44.6
eo	Esperanto	157	0.9	ps	Pashto	96	0.7
es	Spanish	9374	53.3	pt	Portuguese	8405	49.1
et	Estonian	843	6.1	ro	Romanian	10354	61.4
eu	Basque	270	2.0	ru	Russian	23408	278.0
fa	Persian	13259	111.6	sa	Sanskrit	17	0.3
fi	Finnish	6730	54.3	sd	Sindhi	50	0.4
fr	French	9780	56.8	si	Sinhala	243	3.6
fy	Western Frisian	29	0.2	sk	Slovak	3525	23.2
ga	Irish	86	0.5	sl	Slovenian	1669	10.3
gd	Scottish Gaelic	21	0.1	so	Somali	62	0.4
gl	Galician	495	2.9	sq	Albanian	918	5.4
gu	Gujarati	140	1.9	sr	Serbian	843	9.1
ha	Hausa	56	0.3	su	Sundanese	10	0.1
he	Hebrew	3399	31.6	sv	Swedish	77.8	12.1
hi	Hindi	1715	20.2	sw	Swahili	275	1.6
-	Hindi Romanized	88	0.5	ta	Tamil	595	12.2
hr	Croatian	3297	20.5	-	Tamil Romanized	36	0.3
hu	Hungarian	7807	58.4	te	Telugu	249	4.7
hy	Armenian	421	5.5	-	Telugu Romanized	39	0.3
id	Indonesian	22704	148.3	th	Thai	1834	71.7
is	Icelandic	505	3.2	tl	Filipino	556	3.1
it	Italian	4983	30.2	tr	Turkish	2736	20.9
ja	Japanese	530	69.3	ug	Uyghur	27	0.4
jv	Javanese	24	0.2	uk	Ukrainian	6.5	84.6
ka	Georgian	469	9.1	ur	Urdu	730	5.7
kk	Kazakh	476	6.4	-	Urdu Romanized	85	0.5
km	Khmer	36	1.5	uz	Uzbek	91	0.7
kn	Kannada	169	3.3	vi	Vietnamese	24757	137.3
ko	Korean	5644	54.2	xh	Xhosa	13	0.1
ku	Kurdish (Kurmanji)	66	0.4	yi	Yiddish	34	0.3
ky	Kyrgyz	94	1.2	zh	Chinese (Simplified)	259	46.9
la	Latin	390	2.5	zh	Chinese (Traditional)	176	16.6

Table 6: **Languages and statistics of the CC-100 corpus.** We report the list of 100 languages and include the number of tokens (Millions) and the size of the data (in GiB) for each language. Note that we also include romanized variants of some non latin languages such as Bengali, Hindi, Tamil, Telugu and Urdu.

B   Model Architectures and Sizes

As we showed in section 5, capacity is an important parameter for learning strong cross-lingual representations. In the table below, we list multiple monolingual and multilingual models used by the research community and summarize their architectures and total number of parameters.

Model	#lgs	tokenization	L	H_m	H_{ff}	A	V	#params
BERT$_{Base}$	1	WordPiece	12	768	3072	12	30k	110M
BERT$_{Large}$	1	WordPiece	24	1024	4096	16	30k	335M
mBERT	104	WordPiece	12	768	3072	12	110k	172M
RoBERTa$_{Base}$	1	BPE	12	768	3072	8	50k	125M
RoBERTa	1	BPE	24	1024	4096	16	50k	355M
XLM-15	15	BPE	12	1024	4096	8	95k	250M
XLM-17	17	BPE	16	1280	5120	16	200k	570M
XLM-100	100	BPE	16	1280	5120	16	200k	570M
Unicoder	15	BPE	12	1024	4096	8	95k	250M
XLM-R $_{Base}$	100	SPM	12	768	3072	12	250k	270M
XLM-R	100	SPM	24	1024	4096	16	250k	550M
GPT2	1	bBPE	48	1600	6400	32	50k	1.5B
wide-mmNMT	103	SPM	12	2048	16384	32	64k	3B
deep-mmNMT	103	SPM	24	1024	16384	32	64k	3B
T5-3B	1	WordPiece	24	1024	16384	32	32k	3B
T5-11B	1	WordPiece	24	1024	65536	32	32k	11B

Table 7: **Details on model sizes.** We show the tokenization used by each Transformer model, the number of layers L, the number of hidden states of the model H_m, the dimension of the feed-forward layer H_{ff}, the number of attention heads A, the size of the vocabulary V and the total number of parameters #params. For Transformer encoders, the number of parameters can be approximated by $4LH_m^2 + 2LH_mH_{ff} + VH_m$. GPT2 numbers are from Radford et al. (2019), mm-NMT models are from the work of Arivazhagan et al. (2019) on massively multilingual neural machine translation (mmNMT), and T5 numbers are from Raffel et al. (2019). While *XLM-R* is among the largest models partly due to its large embedding layer, it has a similar number of parameters than XLM-100, and remains significantly smaller that recently introduced Transformer models for multilingual MT and transfer learning. While this table gives more hindsight on the difference of capacity of each model, note it does not highlight other critical differences between the models.