BanglaBERT: Combating Embedding Barrier for Low-Resource Language Understanding

Abhik Bhattacharjee*, **Tahmid Hasan***, **Kazi Samin**, **M. Sohel Rahman**, **Anindya Iqbal and Rifat Shahriyar**
Bangladesh University of Engineering and Technology (BUET)
{abhik, samin}@ra.cse.buet.ac.bd,
{tahmidhasan, msrahman, anindya, rifat}@cse.buet.ac.bd

Abstract

Pre-training language models on large volume of data with self-supervised objectives has become a standard practice in natural language processing. However, most such state-of-the-art models are available in only English and other resource-rich languages. Even in multilingual models, which are trained on hundreds of languages, low-resource ones still remain underrepresented. Bangla, the seventh most widely spoken language in the world, is still low in terms of resources. Few downstream task datasets for language understanding in Bangla are publicly available, and there is a clear shortage of good quality data for pre-training. In this work, we build a Bangla natural language understanding model pre-trained on 18.6 GB data we crawled from top Bangla sites on the internet. We introduce a new downstream task dataset and benchmark on four tasks on sentence classification, document classification, natural language understanding, and sequence tagging. Our model outperforms multilingual baselines and previous state-of-the-art results by 1-6%. In the process, we identify a major shortcoming of multilingual models that hurt performance for low-resource languages that don't share writing scripts with any high resource one, which we name the 'Embedding Barrier'. We perform extensive experiments to study this barrier. We release all our datasets and pre-trained models to aid future NLP research on Bangla and other low-resource languages. Our code and data are available at `github.com/csebuetnlp/banglabert`.

1 Introduction

Self-supervised pre-training has enabled training deep neural language models (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019) by leveraging large amounts of unannotated data. This pre-training stage allows the models to learn general linguistic representations (Jawahar et al., 2019) that can later be fine-tuned on different downstream tasks (Wang et al., 2018; Rajpurkar et al., 2016; Tjong Kim Sang and De Meulder, 2003). With little task-specific supervision (Howard and Ruder, 2018), these models have been shown to achieve state-of-the-art results on many natural language understanding (NLU) tasks, that would have been impossible to do on these large models without pre-training. However, the field of natural language understanding is still restricted to only a few high-resource languages. Although there have been efforts to pre-train multilingual models (Pires et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020) that can generalize to hundreds of languages, they often lag in performance compared to language-specific models.

Bangla, the seventh most widely spoken language in the world, has unfortunately, still remained low in resources. This has been attributed to the unavailability of a variety of downstream task datasets and pretrained language models. With these shortcomings in mind, in this work, we build an NLU model for Bangla, pre-trained on 18.6 GB data (2.5B tokens) that are crawled from more than 50 popular Bangla websites. We introduce a new downstream task dataset and evaluate our model on four tasks on sentiment analysis, document classification, natural language inference, and named entity recognition. We outperform multilingual baselines and previous state-of-the-art results on all tasks by 1-6%.

While comparing with the multilingual models, we noticed that even though these models were pretrained with comparable Bangla texts to our model, the vocabulary dedicated to Bangla scripts were less than 1% of the total vocabulary. And since Bangla doesn't share its vocabulary or writing script with any other high-resource language, the

*These authors contributed equally to this work.

vocabulary is strictly limited within that bound. We name this phenomenon as the 'Embedding Barrier'. We conduct a series of experiments to show that

1. There is a considerable loss in performance (up to 4% accuracy) because of the Embedding Barrier.
2. Even a distant low-resource language having the same writing script as a high-resource one may benefit from borrowed tokens (albeit having different meanings) and their contextualized representations.
3. Similar languages may be hindered from having positive transfer because of them being written in different scripts.

Our study also calls for new ways to represent vocabularies of low-resource languages and multilingual models that can lift this barrier.

2 Related Works

Pretraining language models has brought about influential impact in natural language processing over the last decade. In the earlier days of deep learning, feed-forward neural networks were used to pretrain word vectors on large corpora (Mikolov et al., 2013; Pennington et al., 2014). These word embeddings attained from these networks were then used to initialize the embedding layers of more complex networks (Hochreiter and Schmidhuber, 1997) to be used on different tasks. ULMfit (Howard and Ruder, 2018) pre-trained all the weights of a neural network before using it on several downstream tasks. ELMo (Peters et al., 2018) introduced contextualized word embeddings, outperforming previous works. Radford et al. (2019) pretrained GPT using the decoder of the transformer (Vaswani et al., 2017) using both concepts of ULMfit and ELMo, and also showing scaling model size increases performance. BERT (Devlin et al., 2019) introduced masked language modeling (MLM) (Taylor, 1953) into the pretraining objective, achieving both effectiveness and efficiency in large language model pretraining using the encoder block of the Transformer. Several works (Liu et al., 2019; Yang et al., 2019; Raffel et al., 2020; Clark et al., 2020) followed further studying the objectives of BERT and improving performance on the downstream tasks.

Until recently, most of these models focused on English language. mBERT (Pires et al., 2019) followed by XLM (Conneau and Lample, 2019) pretrained on hundreds of languages using the Wikipedia dumps and Ccnet corpus (Wenzek et al., 2020), respectively. One of the first language-specific models other than English, CamemBERT (Martin et al., 2020), trained on French data. After that many works (Polignano et al., 2019; de Vries et al., 2019; Canete et al., 2020) did pretraining for other languages. XLM-RoBERTa (Conneau et al., 2020) explored multilingual pretraining at scale and showed cross-lingual effectiveness of multilingual models.

Although, there are multiple small pretrained models for Bangla available through the Huggingface Transformer Library (Wolf et al., 2020), none of them have pretrained on good quality data nor did any fine-tuning on downstream tasks.

3 Pre-training BanglaBERT

In this section, we describe the pretraining data, preprocessing steps, model architecture, and objectives of BanglaBERT:

3.1 Pretraining Data

A high volume of good quality text data is a prerequisite for pretraining large language models. For instance, BERT is pretrained on the English Wikipedia and the Books corpus (Zhu et al., 2015) containing about 3.3 billion tokens. Subsequent works like RoBERTa (Liu et al., 2019) and Xlnet (Yang et al., 2019) use an order of magnitude larger data crawled from the web that have passed through heavy filtering and cleaning.

Being a rather resource-constrained language in the web domain, the amount of data available for Bangla language is rather small, e.g., the Bangla Wikipedia dump from June 2020 is only 350 megabytes in size, two orders of magnitudes smaller than the English Wikipedia. As a result, we had to crawl the web extensively to collect our pretraining data. We narrowed down more than 50 Bangla websites from Amazon Alexa website rankings [1] and by the volume of extractable texts present there. The contents included encyclopedias, news sites, blogs, e-books, and story sites. We wrote custom crawlers for each website so that only the text contents would be extracted. The amount of extracted data totalled to around 30GB.

Although there are other noisy sources of data available (e.g., OSCAR Suárez et al., 2019), they

[1] `https://www.alexa.com/topsites/countries/BD`

contained a lot of offensive texts and repetitive contents. We found the OSCAR dataset to be too difficult to clean thoroughly, and hence opted not to use it.

3.2 Preprocessing

After obtaining the pretraining data, we performed thorough deduplication using fuzzy string similarity [2]. We filtered out non-Bangla pages using a language ID classifier (Joulin et al., 2017) and also removed non-textual data (e.g., HTML/Javascript tags) should they be present.

After the processing, the dataset was reduced to 18.6 GB with around 3.8M pages. We trained an Wordpiece (Wu et al., 2016) vocabulary of 32k tokens with the resulting corpus with a 400 character alphabet.

While creating a training sample, we kept the maximum sequence length to 512. We avoided truncating a sample midway through a sentence, removing the last sentence completely; and was careful about not to cross document boundaries while creating a sample.

3.3 Pretraining Model and Training Setup

We pretrained our model using the ELECTRA Clark et al. (2020) setup and objective. Instead of masked language modeling, ELECTRA is pretrained with Replaced Token Detection (RTD) objective. In this setup, a generator and a discriminator model are trained jointly. The generator is fed as input a sequence with some masked tokens and is asked to predict the masked tokens from the rest of the input (i.e., standard MLM, mentioned in section 2). For the discriminator input, the masked tokens are replaced by tokens sampled from the generator's output distribution for those masks. The discriminator is then asked to predict whether each token is from the original input or not. After pretraining, the discriminator is used for fine-tuning. The original authors argued that the RTD objective backpropagates loss from all tokens of a sequence, in contrast to 15% tokens of the MLM objective, giving the model more signals to learn from. Evidently, ELECTRA achieves comparable downstream task performance to RoBERTa or Xlnet with only a quarter of training time. Because of the computational efficiency, we used this model for our implementation of BanglaBERT.

We pretrained the base model (110M parameters) for about 534k steps (28 epochs) on an 8-GPU p3.x16large instance on AWS. We used the same hyperparameters (e.g., batch size, learning rate) proposed in the original work.

4 Downstream Tasks and Datasets

We fine-tuned our pretrained model on the four downstream tasks discussed below.

1. **Sentiment Classification**: For Bangla sentiment classification, we used the sentiment analysis dataset from (Sharfuddin et al., 2018). It has two labels: positive and negative, both classes are balanced. There were 8181 samples in total.

2. **Document Classification**: For document classification, we used the ai4bharat Bangla news article classification dataset (Kunchukuttan et al., 2020). There are two classes: sports and entertainment, both balanced. The authors provided a 11284-1411-1411 split.

3. **Natural Language Inference**: Due to the unavailability of any double sentence classification dataset in Bangla, we curated a natural language inference dataset. For this task, two sentences are given as input, and the model has to predict whether or not the second sentence is neutral, entailment, or contradictory to the first sentence. We used the same curation procedure as XNLI (Conneau et al., 2018): we translated the MultiNLI (Williams et al., 2018) training data using the English to Bangla translation model by (Hasan et al., 2020), and had the evaluation sets translated by experts.

4. **Named Entity Recognition**: For this task, we used the NER dataset from (Karim et al., 2019). A 64155-3565-3564 split was provided by the authors. The dataset had some annotation errors that didn't follow the IOB2 sequence labeling scheme [3]. We fixed the annotations before fine-tuning.

5 Experiments and Benchmark Results

In this section, we discuss the fine-tuning procedure of BanglaBERT. We also show comparative results with the multilingual baseline XLM-R and previous

[2] https://docs.python.org/3/library/difflib.html

[3] https://w.wiki/qsm

state-of-the-art works for the downstream tasks. All models were fine-tuned for three epochs on each task and the learning rate was tuned from the set **{1e-5, 2e-5, 3e-5, 4e-5, 5e-5}**. We show the test set score for the models that had the best results on the dev set. The results on different downstream tasks are shown below.

1. **Sentiment Classification**: Sharfuddin et al. (2018) did not provide any train-dev-test split with their dataset, unfortunately. Hence, we made a 80-10-10 split randomly. Positive and negative labels were split separately and then merged into respective sets to ensure all three sets remained balanced. As the dataset was collected from YouTube comments, there were lots on emojis present. We converted the emojis into texts using bnemo package[4]. We used accuracy as the performance metric. The results are shown in Table 1.

Model	Accuracy (%)
Sharfuddin et al. (2018)	85.67
XLM-R	87.54
BanglaBERT (ours)	**91.69**

Table 1: Performance on sentiment classification

2. **Document Classification**: For document classification, we show the baseline performance provided by Kunchukuttan et al. (2020) as well as the performance of XLM-R. While fine-tuning with XLM-R and BanglaBERT, we had to limit the maximum sequence length to 512. We show the performances in Table 2.

Model	Accuracy (%)
Kunchukuttan et al. (2020)	72.50
XLM-R	92.70
BanglaBERT (ours)	**98.92**

Table 2: Performance on document classification

3. **Natural Language Inference**: Since the NLI dataset is introduced by us, we use XLM-R as baseline for comparing with BanglaBERT. The results are shown in Table 3.

4. **Named Entity Recognition**: For NER, we compared our results with Ashrafi et al. (2020), who used an M-BERT backbone; along with XLM-R. As we had to correct some labels in the NER dataset, we reproduced the experiments of BANNER to ensure fair comparison. We followed a fine-tuning based approach for NER. We used micro-F1 as the scoring metric; and show both results in Table 4.

[4]https://pypi.org/project/bnemo/

Model	Accuracy (%)
XLM-R	75.02
BanglaBERT (ours)	**80.87**

Table 3: Performance on natural language inference

Model	Micro-F1
Ashrafi et al. (2020)	65.78
XLM-R	71.36
BanglaBERT	**72.11**

Table 4: Performance on named entity recognition

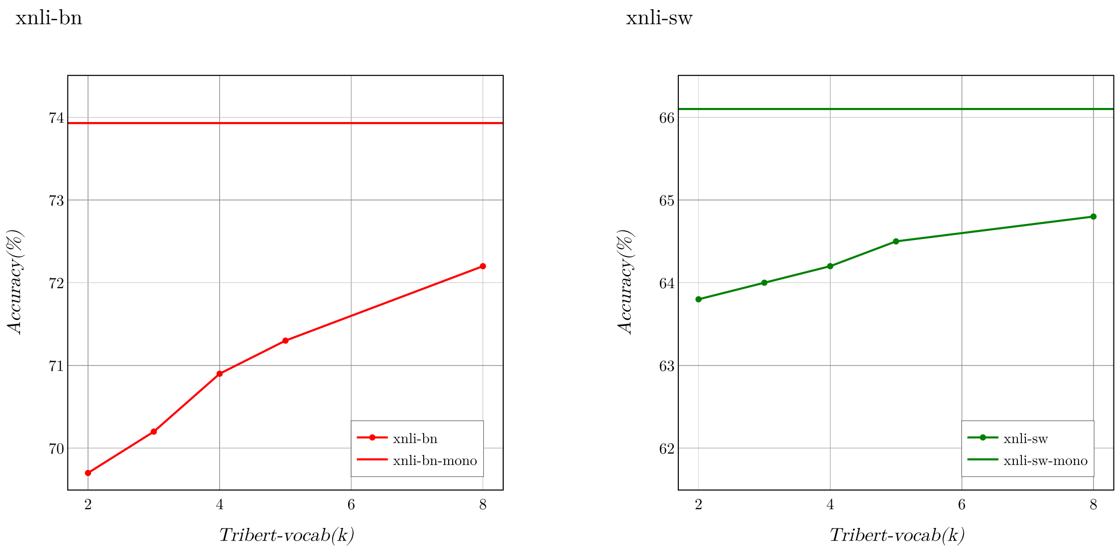
In all the downstream tasks, BanglaBERT performed better than baseline models and XLM-R by a considerable margin ranging from 1%-7%.

6 Addressing the Embedding Barrier

It was quite expected that BanglaBERT would perform better than the multilingual XLM-RoBERTa, since all model parameters were being used for one single language in our case. But XLM-RoBERTa had better scores on many tasks when compared with Greek (Koutsikakis et al., 2020) and Vietnamese BERT (Nguyen and Tuan Nguyen, 2020). This prompted us to further investigate the reason behind the under-performance.

Interestingly, we found that less than 2,500 tokens were allocated to Bangla vocabulary, an amount we believe too low to convey the morphological richness for Bangla language. Moreover, Bangla doesn't share its writing script with any high-resource language, which limits the usage of borrowed tokens. For instance, a Swahili Wikipedia dump of 36 megabytes tokenized on the XLM-RoBERTa vocabulary had more than 6500 tokens that had frequency over 100. The same size dump for Bangla has only about 2,100 tokens.

Although the pre-training corpus for Bangla used in XLM-RoBERTa is an order of magnitude bigger than Swahili, Swahili managed to make use of more tokens. Since Swahili uses Latin scripts, it can borrow tokens from other high and mid-resource languages that use Latin script (57 out



(a) caption-1

(b) caption-2

Figure 1: caption-3

of 100 languages supported in XLM-RoBERTa use Latin script and almost half of the total vocabularies are from the Latin block [5]. Although the borrowed tokens may not necessarily convey the same meaning across different languages, contextualization of the embedding representations (Peters et al., 2018) overcomes this problem. The absence of this opportunity puts low-resource languages that don't share their writing scripts with any high or mid-resource language at a great disadvantage.

We further explored the vocabulary list of XLM-RoBERTa and found about 193k tokens out of 250k belong to the top five scripts (namely, Latin, Cyrillic, Arabic, CJK, and Devanagari). 21 languages didn't share their scripts with any of these languages, and consequently, their vocabularies were limited to 2.6k tokens on average. We name this problem as the 'Embedding Barrier'.

7 Simulating Embedding Barrier

In this section, we simulate the embedding barrier in a controlled environment. To see how the size of native vocabulary affects downstream performance, we pre-trained multiple multilingual ELECTRA models on Bangla, Swahili, and English data with different vocabulary size combinations. We chose Swahili as it is also a low-resource language, shares its writing script with English, and is a distant language from Bangla and English. Moreover, the XNLI dataset has support for Swahili, so we would have a common task to evaluate the languages on. We collected about 280 megabytes of Swahili data from the Swahili Wikipedia dump and by crawling about ten popular Swahili news and blog sites. We downsampled our Bangla corpus to have about the same size as its Swahili counterpart. We used similar sources in equal proportion so that we may have as much similarity as possible. We used the Books corpus (Zhu et al., 2015) and English Wikipedia dump as the English pre-training corpus.

We trained Wordpiece vocabularies of size 2k, 3k, 4k, 5k, and 8k on the Bangla and Swahili data separately. For each setup, we trained English vocabularies so that the total combined vocabulary for each case would remain constant to 32k. Thus, for instance, a combined vocabulary containing 2k Bangla vocabulary and 2k Swahili vocabulary would have approximately 28k (28611 in reality, because of common tokens between Swahili and English) tokens trained from the English corpus. We tokenized all training corpora on the combined vocabulary for each setup so that each language can borrow tokens from one another.

We trained five small ELECTRA (14M parameters) models on the five vocabularies for 250k steps using the same hyperparameters specified in the original work. Note than other than the vocabularies, training data and model setups are the same across all models. We also trained two models

[5] https://jrgraphix.net/r/Unicode/

solely on the Swahili and downsampled Bangla data to see how a monolingual model performs in comparison with the trilingual models (these models would thus have the full 32k vocabulary trained on the native corpus). We plot the performances of each model on the XNLI-sw and XNLI-bn dataset in Figure 1. The horizontal lines indicate the performances of the monolingual models.

From the figures, it can be seen that Bangla is affected more than Swahili due to the embedding barrier. The TriBERT-2k model has 4% lower accuracy than the monolingual model, while it trails only 2% in Swahili. As the vocabulary size increases, accuracy improves for both languages, but the curve is steeper for Bangla. This gives evidence that Bangla suffers more than Swahili from having a smaller number of vocabulary.

8 Borrowed Tokens or Positive Transfer?

It might be argued that the superior performance of Swahili in comparison with Bangla in the trilingual models is due to some other linguistic factors; not because of borrowed tokens (e.g., English and Swahili both have the same SVO typography). In order to validate that this is not the case, we trained two more models: one with the TriBERT-2k vocabulary of 32k tokens and another with a 2k vocabulary trained only on the Swahili corpus. If there were indeed positive transfer from English to Swahili, both of these models would perform worse than the TriBERT-2k models. We show the accuracy of all models in Table 5.

Model Name	Accuracy
TriBERT-2k	63.79
BERT-sw (Swahili vocab-2k)	63.43
BERT-sw (combined vocab-32k)	**64.73**

Table 5: Effect of borrowed tokens

As expected, BERT-sw model with 2k vocabulary performed the worst, because of the vocabulary being strictly limited. The BERT-sw model with combined vocabulary had the best performance as it was able to take advantage of the extra tokens borrowed from the English side. If the improvement in performance of Swahili in Section 7 was because of the positive transfer from the English side to Swahili, then the TriBERT-2k model would have performed better than Swahili model with combined vocabulary. This is a substantial proof that the advantage gained by Swahili is solely by being able to use the extra tokens from the English side.

We also probed into the preprocessed fine-tuning data of the BERT-sw model with the combined vocabulary and found that there were about 7.8k unique tokens in the fine-tuning data, thus enabling the data to have a higher vocabulary range.

9 Do Similar Languages with Different Scripts Allow Positive Transfer?

From the previous two sections, it is quite clear that having the same writing scripts as a high-resource language puts even a distant and low-resource one at a good advantage. However, it may be possible that low-resource languages with unique scripts may still have positive transfer from similar languages that have different scripts. To show that this is not the case, we again trained a small ELECTRA model on three languages. This time we replaced Swahili with Hindi, a language from the same language branch as Bangla. We collected our Hindi pretraining data from the Hindi Wikipedia dump and by crawling two Hindi blogs and news sites. We again subsampled our data to have the same size as the experiments done in Section 7. We trained a TriBERT-2k model in this manner. We show the comparison of this model and TriBERT-2k from section 7 in Table 6.

Model Name	Accuracy
TriBERT-2k (bn-sw-en)	69.74
TriBERT-2k (bn-hi-en)	69.86

Table 6: Positive transfer from similar languages

To our surprise, even Hindi provided little to no positive transfer, because of having a different writing script. Though all the parameters of XLM-R is shared by all languages, because of the embeddings being different from the beginning, the model failed to transfer knowledge between the tokens of the two languages.

10 Discussion

We understand that large multilingual language models are limited by their capacity, and the embedding layer has to be kept limited to a fixed vocabulary (250k in the case of XLM-R). But languages with unique scripts that don't share their scripts with other languages suffer the most here. Not only do they have a limited native vocabulary, they also cannot borrow tokens from other languages. From section 7, we see that the limited capacity of vocabulary does indeed have a more severe effect on low-resource languages with unique scripts, and borrowed tokens can indeed help alleviate this barrier to some extent. And even similar languages are prevented from having positive transfer for having different writing scripts. So, these type of languages are left to fend for themselves.

11 Conclusion and Future Works

In this work, we pre-trained a Bangla natural language understanding model from 18.6GB data we collected from crawling over 50 sites. We introduced a downstream dataset on natural language inference and benchmarked on four tasks, setting new state-of-the-art results in all of them. While making the benchmark comparison, we identified a unique problem endemic to low-resource languages that don't share their writing scripts with any high or mid-resource language: the embedding barrier. We performed several controlled experiments to study the barrier in depth. We showed that even a low-resource language having the same writing scripts as a high-resource language can take advantage of borrowed tokens through contextualization of their embeddings, and it is through borrowed tokens that they don't suffer heavily from embedding barrier. We further showed that similar languages can be hindered from positive transfer due to having different writing scripts.

We believe that our contributions in this work will help the Bangla NLP community in using transfer learning in other natural language understanding tasks by using our pretrained BanglaBERT. We further hope that our studies will call for new methods to tackle the embeddding barrier for low-resource languages.

In future, we would like to fine-tune our model on other NLU tasks, as well as use the pretrained wrights to initialize NLG models. We would like to investigate the embedding barrier further, and design new methods to lift this barrier.

Acknowledgements

We would like to thank Intelligent Machines Limited for providing cloud support.

References

I. Ashrafi, M. Mohammad, A. S. Mauree, G. M. A. Nijhum, R. Karim, N. Mohammed, and S. Momen. 2020. Banner: A cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

José Canete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. In *Proceedings of the Practical ML for Developing Countries Workshop at ICLR 2020, PML4DC*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel 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*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

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

Redwanul Karim, MA Islam, Sazid Rahman Simanto, Saif Ahmed Chowdhury, Kalyan Roy, Adnan Al Neon, Md Hasan, Adnan Firoze, Rashedur M Rahman, et al. 2019. A step towards information extraction: Named entity recognition in bangla using deep learning. *Journal of Intelligent & Fuzzy Systems*, 37(6):7401–7413.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-bert: The greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 110–117. Association for Computing Machinery.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

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*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online. Association for Computational Linguistics.

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

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

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019*, CEUR Workshop Proceedings.

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. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

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.

Abdullah Aziz Sharfuddin, Md Nafis Tihami, and Md Saiful Islam. 2018. A deep recurrent neural network with bilstm model for sentiment classification. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.

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

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*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, page 6000–6010, Long Beach, California, USA.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA. IEEE Computer Society.