BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla

Abhik Bhattacharjee\(^1\)\(^\ast\), Tahmid Hasan\(^1\), Wasi Uddin Islam\(^3\), Kazi Samin\(^1\), Md Saiful Islam\(^3\), Anindya Iqbal\(^1\), M. Sohel Rahman\(^1\), Rifat Shahriyar\(^1\)

Bangladesh University of Engineering and Technology (BUET)\(^1\), AWS AI Labs\(^2\), University of Rochester\(^3\)

abhik@ra.cse.buet.ac.bd, \{tahmidhasan,rifat\}@cse.buet.ac.bd

Abstract

In this work, we introduce BanglaBERT, a BERT-based Natural Language Understanding (NLU) model pretrained in Bangla, a widely spoken yet low-resource language in the NLP literature. To pretrain BanglaBERT, we collect 27.5 GB of Bangla pretraining data (dubbed ‘Bangla2B’) by crawling 110 popular Bangla sites. We introduce two downstream task datasets on natural language inference and question answering and benchmark on four diverse NLU tasks covering text classification, sequence labeling, and span prediction. In the process, we bring them under the first-ever Bangla Language Understanding Benchmark (BLUB). BanglaBERT achieves state-of-the-art results outperforming multilingual and monolingual models. We are making the models, datasets, and a leaderboard publicly available at https://github.com/csebuetnlp/banglabert to advance Bangla NLP.

1 Introduction

Despite being the sixth most spoken language in the world with over 300 million native speakers constituting 4% of the world’s total population,\(^1\) Bangla is considered a resource-scarce language. Joshi et al. (2020b) categorized Bangla in the language group that lacks efforts in labeled data collection and relies on self-supervised pretraining (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019) to boost the natural language understanding (NLU) task performances. To pretrain BanglaBERT, we collect 27.5 GB of Bangla pretraining data (dubbed ‘Bangla2B’) by crawling 110 popular Bangla sites.

We also introduce two datasets on Bangla Natural Language Inference (NLI) and Question Answering (QA), tasks previously unexplored in Bangla, and evaluate both pretrained models on four diverse downstream tasks on sentiment classification, NLI, named entity recognition, and QA. We bring these tasks together to establish the first-ever Bangla Language Understanding Benchmark (BLUB). We compare widely used multilingual models to BanglaBERT using BLUB and find that both models excel on all the tasks.

We summarize our contributions as follows:

1. We present two pretrained models in Bangla: BanglaBERT and BanglishBERT, and introduce new Bangla NLI and QA datasets.

2. We introduce the Bangla Language Understanding Benchmark (BLUB) and show that, in the supervised setting, BanglaBERT outperforms mBERT and XLM-R (base) by 6.8 and 4.3 BLUB scores, while in zero-shot cross-lingual transfer, BanglishBERT outperforms...
them by 15.8 and 10.8, respectively.

3. We provide the code, models, and a leaderboard to spur future research on Bangla NLU.

2 BanglaBERT

2.1 Pretraining Data

A high volume of good quality text data is a prerequisite for pretraining large language models. For instance, BERT (Devlin et al., 2019) is pretrained on the English Wikipedia and the Books corpus (Zhu et al., 2015) containing 3.3 billion tokens. Subsequent works like RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) used more extensive web-crawled data with heavy filtering and cleaning.

Bangla is a rather resource-constrained language in the web domain; for example, the Bangla Wikipedia dump from July 2021 is only 650 MB, 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 selected 110 Bangla websites by their Amazon Alexa rankings\textsuperscript{2} and the volume and quality of extractable texts by inspecting each website. The contents included encyclopedias, news, blogs, e-books, stories, social media/forums, etc.\textsuperscript{3} The amount of data totaled around 35 GB.

There are noisy sources of Bangla data dumps, a couple of prominent ones being OSCAR (Suárez et al., 2019) and CCNet (Wenzek et al., 2020). They contained many offensive texts; we found them infeasible to clean thoroughly. Fearing their potentially harmful impacts (Luccioni and Viviano, 2021), we opted not to use them. We further discuss ethical considerations at the end of the paper.

2.2 Pre-processing

We performed thorough deduplication on the pretraining data, removed non-textual contents (e.g., HTML/JavaScript tags), and filtered out non-Bangla pages using a language classifier (Joulin et al., 2017). After the processing, the dataset was reduced to 27.5 GB in size containing 5.25M documents having 306.66 words on average.

We trained a Wordpiece (Wu et al., 2016) vocabulary of 32k subword tokens on the resulting corpus with a 400 character alphabet, kept larger than the native Bangla alphabet to capture code-switching (Poplack, 1980) and allow romanized Bangla contents for better generalization. We limited the length of a training sample to 512 tokens and did not cross document boundaries (Liu et al., 2019) while creating a data point. After tokenization, we had 7.18M samples with an average length of 304.14 tokens and containing 2.18B tokens in total; hence we named the dataset ‘Bangla2B+’.

2.3 Pretraining Objective

Self-supervised pretraining objectives leverage unlabeled data. For example, BERT (Devlin et al., 2019) was pretrained with masked language modeling (MLM) and next sentence prediction (NSP). Several works built on top of this, e.g., RoBERTa (Liu et al., 2019) removed NSP and pretrained with longer sequences, SpanBERT (Joshi et al., 2020a) masked contiguous spans of tokens, while works like XLNet (Yang et al., 2019) introduced objectives like factorized language modeling.

We pretrained BanglaBERT using ELECTRA (Clark et al., 2020b), pretrained with the Replaced Token Detection (RTD) objective, where a generator and a discriminator model are trained jointly. The generator is fed as input a sequence with a portion of the tokens masked (15% in our case) and is asked to predict them using the rest of the input (i.e., standard MLM). The masked tokens are then replaced by tokens sampled from the generator’s output distribution for the corresponding masks, and the discriminator then has to predict whether each token is from the original sequence or not. After pretraining, the discriminator is used for fine-tuning. Clark et al. (2020b) argued that RTD back-propagates 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 performance to RoBERTa or XLNet with only a quarter of their training time. This computational efficiency motivated us to use ELECTRA for our implementation of BanglaBERT.

2.4 Model Architecture & Hyperparameters

We pretrained the base ELECTRA model (a 12-layer Transformer encoder with 768 embedding size, 768 hidden size, 12 attention heads, 3072 feed-forward size, generator-to-discriminator ratio $\frac{1}{3}$, 110M parameters) with 256 batch size for 2.5M steps on a v3-8 TPU instance on GCP. We used the Adam (Kingma and Ba, 2015) optimizer with a 2e-4 learning rate and linear warmup of 10k steps.
2.5 **BanglishBERT**

Often labeled data in a low-resource language for a task may not be available but be abundant in high-resource languages like English. In these scenarios, zero-shot cross-lingual transfer (Artetxe and Schwenk, 2019) provides an effective way to be still able to train a multilingual model on that task using the high-resource languages and be able to transfer to low-resource ones. To this end, we pretrained a bilingual model, named **BanglishBERT**, on Bangla and English together using the same set of hyperparameters mentioned earlier. We used the BERT pretraining corpus as the English data and trained a joint bilingual vocabulary (each language having $\sim 16k$ tokens). We upsampled the Bangla data during training to equal the participation of both languages.

3 **The Bangla Language Understanding Benchmark (BLUB)**

Many works have studied different Bangla NLU tasks in isolation, e.g., sentiment classification (Das and Bandyopadhyay, 2010; Sharfuddin et al., 2018; Tripto and Ali, 2018), semantic textual similarity (Shajalal and Aono, 2018), parts-of-speech (PoS) tagging (Alam et al., 2016), named entity recognition (NER) (Ashrafi et al., 2020). However, Bangla NLU has not yet had a comprehensive, unified study. Motivated by the surge of NLU research brought about by benchmarks in other languages, e.g., English (Wang et al., 2018), French (Le et al., 2020), Korean (Park et al., 2021), we establish the first-ever Bangla Language Understanding Benchmark (BLUB). NLU generally comprises three types of tasks: text classification, sequence labeling, and text span prediction. Text classification tasks can further be sub-divided into single-sequence and sequence-pair classification. Therefore, we consider a total of four tasks for BLUB. For each task type, we carefully select one downstream task dataset. We emphasize the quality and open availability of the datasets while making the selection. We briefly mention them below.

| Task                          | Corpus       | |Train| | |Dev| | |Test| | Metric     | Domain          |
|------------------------------|--------------|------------------|------|------|------|----------------|----------------|
| Sentiment Classification     | SentNoB      | 12,575           | 1,567| 1,567| Macro-F1   | Social Media   |
| Natural Language Inference   | BNLI         | 381,449          | 2,419| 4,895| Accuracy   | Miscellaneous  |
| Named Entity Recognition     | MultiCoNER   | 14,500           | 800  | 800  | Micro-F1   | Miscellaneous  |
| Question Answering           | BQA, TyDiQA  | 127,771          | 2,502| 2,504| EM/F1      | Wikipedia      |

Table 1: Statistics of the Bangla Language Understanding Evaluation (BLUB) benchmark.

1. **Single-Sequence Classification** Sentiment classification is perhaps the most-studied Bangla NLU task, with some of the earlier works dating back over a decade (Das and Bandyopadhyay, 2010). Hence, we chose this as the single-sequence classification task. However, most Bangla sentiment classification datasets are not publicly available. We could only find two public datasets: **BYSa** (Tripto and Ali, 2018) and **SentNoB** (Islam et al., 2021). We found BYSA to have many duplications. Even worse, many duplicates had different labels. SentNoB had better quality and covered a broader set of domains, making the classification task more challenging. Hence, we opted to use the latter.

2. **Sequence-pair Classification** In contrast to single-sequence classification, there has been a dearth of sequence-pair classification works in Bangla. We found work on semantic textual similarity (Shajalal and Aono, 2018), but the dataset is not publicly available. As such, we curated a new Bangla Natural Language Inference (**BNLI**) dataset for sequence-pair classification. We chose NLI as the representative task due to its fundamental importance in NLU. Given two sentences, a premise and a hypothesis as input, a model is tasked to predict whether or not the hypothesis is entailment, contradiction, or neutral to the premise. We used the same curation procedure as the XNLI (Conneau et al., 2018) dataset: 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 expert human translators. Due to the possibility of the incursion of errors during automatic translation, we used the Language-Agnostic BERT Sentence Embeddings (LaBSE) (Feng et al., 2020) of the translations and original sentences to compute their similarity and discarded all sentences below a similarity threshold of 0.70. Moreover, to ensure good-quality human translation, we used similar quality assurance strategies as Guzmán et al. (2019).
<table>
<thead>
<tr>
<th>Models</th>
<th>Params.</th>
<th>SC</th>
<th>NLI</th>
<th>NER</th>
<th>QA</th>
<th>BLUB Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zero-shot cross-lingual transfer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT</td>
<td>180M</td>
<td>27.05</td>
<td>62.22</td>
<td>39.27</td>
<td>59.01/64.18</td>
<td>50.35</td>
</tr>
<tr>
<td>XLM-R (base)</td>
<td>270M</td>
<td>42.03</td>
<td>72.18</td>
<td>45.37</td>
<td>55.03/61.83</td>
<td>55.29</td>
</tr>
<tr>
<td>XLM-R (large)</td>
<td>550M</td>
<td>49.49</td>
<td>78.13</td>
<td>56.48</td>
<td>71.13/77.70</td>
<td>66.59</td>
</tr>
<tr>
<td>BanglishBERT</td>
<td>110M</td>
<td>48.39</td>
<td>75.26</td>
<td>55.56</td>
<td>72.87/78.63</td>
<td>66.14</td>
</tr>
<tr>
<td><strong>Supervised fine-tuning</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT</td>
<td>180M</td>
<td>67.59</td>
<td>75.13</td>
<td>68.97</td>
<td>67.12/72.64</td>
<td>70.29</td>
</tr>
<tr>
<td>XLM-R (base)</td>
<td>270M</td>
<td>69.54</td>
<td>78.46</td>
<td>73.32</td>
<td>68.09/74.27</td>
<td>72.82</td>
</tr>
<tr>
<td>XLM-R (large)</td>
<td>550M</td>
<td>70.97</td>
<td>82.40</td>
<td>78.39</td>
<td>73.15/79.06</td>
<td>76.79</td>
</tr>
<tr>
<td>IndicBERT</td>
<td>18M</td>
<td>68.41</td>
<td>77.11</td>
<td>54.13</td>
<td>50.84/57.47</td>
<td>61.59</td>
</tr>
<tr>
<td>sahajBERT</td>
<td>18M</td>
<td>71.12</td>
<td>76.92</td>
<td>70.94</td>
<td>65.48/70.69</td>
<td>71.03</td>
</tr>
<tr>
<td>BanglishBERT</td>
<td>110M</td>
<td>70.61</td>
<td>80.95</td>
<td>76.28</td>
<td>72.43/78.40</td>
<td>75.73</td>
</tr>
<tr>
<td>BanglaBERT</td>
<td>110M</td>
<td><strong>72.89</strong></td>
<td><strong>82.80</strong></td>
<td><strong>77.78</strong></td>
<td><strong>72.63/79.34</strong></td>
<td><strong>77.09</strong></td>
</tr>
</tbody>
</table>

Table 2: Performance comparison of pretrained models on different downstream tasks. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling (Koehn, 2004).

3. Sequence Labeling In this task, all words of a text sequence have to be classified. Named Entity Recognition (NER) and Parts-of-Speech (PoS) tagging are two of the most prominent sequence labeling tasks. We chose the Bangla portion of SemEval 2022 MultiCoNER (Malmasi et al., 2022) dataset for BLUB.

4. Span Prediction Extractive question answering is a standard choice for text span prediction. Similar to BNLI, we machine-translated the SQuAD 2.0 (Rajpurkar et al., 2018) dataset and used it as the training set (BQA). For validation and test, We used the Bangla portion of the TyDiQA\(^5\) (Clark et al., 2020a) dataset. We posed the task analogous to SQuAD 2.0: presented with a text passage and a question, a model has to predict whether or not it is answerable. If answerable, the model has to find the minimal text span that answers the question.

We present detailed statistics of the BLUB benchmark in Table 1.

4 Experiments & Results

Setup We fine-tuned BanglaBERT and BanglishBERT on the four downstream tasks and compared them with several multilingual models: mBERT (Devlin et al., 2019), XLM-R base and large (Conneau et al., 2020), and IndicBERT (Kakwani et al., 2020), a multilingual model for Indian languages; and sahajBERT (Diskin et al., 2021), an ALBERT-based (Lan et al., 2020) PLM for Bangla. All pretrained models were fine-tuned for 3-20 epochs with batch size 32, and the learning rate was tuned from \(2e^{-5}, 3e^{-5}, 4e^{-5}, 5e^{-5}\). The final models were selected based on the validation performances after each epoch. We performed fine-tuning with three random seeds and reported their average scores in Table 2. We reported the average performance of all tasks as the BLUB score.

Zero-shot Transfer We show the zero-shot cross-lingual transfer results of the multilingual models fine-tuned on the English counterpart of each dataset (SentNob has no English equivalent; hence we used the Stanford Sentiment Treebank (Socher et al., 2013) for the sentiment classification task) in Table 2. In zero-shot transfer setting, BanglishBERT achieves strong cross-lingual performance over similar-sized models and falls marginally short of XLM-R (large). This is an expected outcome since cross-lingual effectiveness depends explicitly on model size (K et al., 2020).

Supervised Fine-tuning In the supervised fine-tuning setup, BanglaBERT outperformed multilingual models and monolingual sahajBERT on all the tasks, achieving a BLUB score of 77.09, even coming head-to-head with XLM-R (large). On the other hand, BanglishBERT marginally lags behind BanglaBERT and XLM-R (large). BanglaBERT is not only superior in performance but also substantially compute- and memory-efficient. For instance, it may seem that sahajBERT is more efficient than BanglaBERT due to its smaller size, but it takes 2-3.5x time and 2.4-3.33x memory as BanglaBERT.
Sample efficiency  It is often challenging to annotate training samples in real-world scenarios, especially for low-resource languages like Bangla. So, in addition to compute- and memory-efficiency, sample-efficiency (Howard and Ruder, 2018) is another necessity of PLMs. To assess the sample efficiency of BanglaBERT, we limit the number of training samples and see how it fares against other models. We compare it with XLM-R (large) and plot their performances on the SC and NLI tasks for different sample size in Figure 1.

Results show that when we have fewer number of samples (≤ 1k), BanglaBERT has substantially better performance (2-9% on SC and 6-10% on NLI with \( p < 0.05 \)) over XLM-R (large), making it more practically applicable for resource-scarce downstream tasks.

5 Conclusion & Future Works

Creating language-specific models is often infeasible for low-resource languages lacking ample data. Hence, researchers are compelled to use multilingual models for languages that do not have strong pretrained models. To this end, we introduced BanglaBERT and BanglishBERT, two NLU models in Bangla, a widely spoken yet low-resource language. We presented new downstream datasets on NLI and QA, and established the BLUB benchmark, setting new state-of-the-art results with BanglaBERT. In future, we will include other Bangla NLU benchmarks (e.g., dependency parsing (de Marneffe et al., 2021)) in BLUB and investigate the benefits of initializing Bangla NLG models from BanglaBERT.

Acknowledgements

We would like to thank the Research and Innovation Centre for Science and Engineering (RISE), BUET, for funding the project, and Intelligent Machines Limited and Google TensorFlow Research Cloud (TRC) Program for providing cloud support.

Ethical Considerations

Dataset and Model Release  The Copy Right Act, 2000\(^8\) of Bangladesh allows reproduction and public release of copy-right materials for non-commercial research purposes. As a transformative research work, we will release BanglaBERT under a non-commercial license. Furthermore, we will release only the pretraining data for which we know the distribution will not cause any copyright infringement. The downstream task datasets can all be made publicly available under a similar non-commercial license.

Quality Control in Human Translation  Translations were done by expert translators who provide translation services for renowned Bangla newspapers. Each translated sentence was further assessed for quality by another expert. If found to be of low quality, it was again translated by the original translator. The sample was then discarded altogether if found to be of low quality again. Fewer than 100 samples were discarded in this process. Translators were paid as per standard rates in local currencies.

Text Content  We tried to minimize offensive texts in the pretraining data by explicitly crawling the sites where such contents would be nominal. However, we cannot guarantee that there is absolutely no objectionable content present and therefore recommend using the model carefully, especially for text generation purposes.

\(^6\)We present a detailed comparison in the Appendix.  
\(^7\)Results for the other tasks can be found in the Appendix.  
\(^8\)http://bdlaws.minlaw.gov.bd/act-details-846.html
References


Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D Manning. 2020b. ELECTRA: pre-training text encoders as discriminators rather than generators. In 8th International Conference on Learning Representations, ICLR 2020, April, 2020, Online.


Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman,


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.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pages 1–6. IEEE.


Appendix

Pretraining Data Sources

We used the following sites for data collection. We categorize the sites into six types:

Encyclopedia:
- bn.banglapedia.org
- bn.wikipedia.org
- songramernotebook.com

News:
- anandabazar.com
- arthoniteerkagoj.com
- bangla.24livenews.com
- bangla.bdnews24.com
- bangla.dhakatribune.com
- bangla.hindustantimes.com
- bangladeshakhela.com
- banglanews24.com
- banglatribune.com
- bbc.com
- bd-journal.com
- bd-pratidin.com
- bd24live.com
- bengali.indianexpress.com
- bigganprojekti.com
- bonikbarta.net
- chakarianews.com
- channelionline.com
- ctgtimes.com
- ctn24.com
- daily-bangladesh.com
- dailyagnishikha.com
- dainikazadi.net
- dainikdinkal.net
- dailyfulki.com
- dailyinqilab.com
- dailynayadiganta.com
- dailysangram.com
- dailysylhet.com
- dainikamadershomoy.com
- dainikhshika.com
- dhakardak-bd.com
- dmpnews.org
- dw.com
- eisamay.indiatimes.com
- ittefaq.com.bd
- jagonews24.com
- jugantor.com
- kalerkantho.com
- manobkantha.com.bd
- mzamin.com
- ntvbd.com
- onnodristy.com
- pavilion.com.bd
- prothomalo.com
- protidinersangbad.com
- risingbd.com
- rtvonline.com
- samakal.com
- sangbadpratidin.in
- somoyerkonchosor.com
- somoynews.tv
- tbsnews.net
- teknafe.com
- thedailystar.net
- voabangla.com
- zeeindia.net
- zoombangla.com

Blogs:
- amrabondhu.com
- banglalblog.in
- bigganblog.org
- biggani.org
- bigyan.org.in
- bishorgo.com
- cadetcollegeblog.com
- choturmatrik.com
- horoppa.wordpress.com
- muktangon.blogspot.com
- roar.media/bangla
- sachalayatan.com
- shodalap.org
- shopnobaz.net
- somewhereinblog.net
- subeen.com
- tunerpage.com
- tutobd.com

E-books/Stories:
- banglaepub.github.io
- bengali.pratilipi.com
- bn.wikisource.org
- ebanglalibrary.com
- eboipotro.github.io
- golpokobita.com
- kaliokalam.com
- shirisherdalpala.net
- tagoreweb.in

Social Media/Forums:
- banglacricket.com
- bn.globalvoices.org
- helpfulhub.com
- nirbik.com
- pchelplinebd.com
- techtunes.io
Miscellaneous:
- banglasonglyric.com
- bdlaws.minlaw.gov.bd
- bdup24.com
- bengalisongslyrics.com
- dakghar.org
- gdn8.com
- gunijan.org.bd
- hrw.org
- jakir.me
- jhankarmahbub.com
- jw.org
- lyricsbangla.com
- neonaloy.com
- porjotonlipi.com
- sasthabangla.com
- tanzil.net

We wrote custom crawlers for each site above (except the Wikipedia dumps).

Additional Sample Efficiency Tests
We plot the the sample efficiency results of the NER and QA tasks in Figure 2.

Figure 2: Sample-efficiency tests with NER and QA.

Similar results are also observed here for the NER task, where BanglaBERT is more sample-efficient when we have \( \leq 1k \) training samples. In the QA task however, both models have identical performance for all sample counts.

Compute and Memory Efficiency Tests
To validate that BanglaBERT is more efficient in terms of memory and compute, we measured each model’s training time and memory usage during the fine-tuning of each task. All tests were done on a desktop machine with an 8-core Intel Core-i7 11700k CPU and NVIDIA RTX 3090 GPU. We used the same batch size, gradient accumulation steps, and sequence length for all models and tasks for a fair comparison. We use relative time and memory (GPU VRAM) usage considering those of BanglaBERT as units. The results are shown in Table 3. (We mention the upper and lower values of the different tasks for each model)

<table>
<thead>
<tr>
<th>Model</th>
<th>Time</th>
<th>Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT</td>
<td>1.14x-1.92x</td>
<td>1.12x-2.04x</td>
</tr>
<tr>
<td>XLM-R (base)</td>
<td>1.29-1.81x</td>
<td>1.04-1.63x</td>
</tr>
<tr>
<td>XLM-R (large)</td>
<td>3.81-4.49x</td>
<td>4.44-5.55x</td>
</tr>
<tr>
<td>SahajBERT</td>
<td>2.40-3.33x</td>
<td>2.07-3.54x</td>
</tr>
<tr>
<td>BanglaBERT</td>
<td>1.00x</td>
<td>1.00x</td>
</tr>
</tbody>
</table>

Table 3: Compute and memory efficiency tests