Improving Bengali and Hindi Large Language Models

Arif Shahriar, Denilson Barbosa
Department of Computing Science
University of Alberta
{ashahri1, denilson}@ualberta.ca

Abstract

Despite being widely spoken worldwide, Bengali and Hindi are low-resource languages. The state-of-the-art in modeling such languages uses BERT and the Wordpiece tokenizer. We observed that the Wordpiece tokenizer often breaks words into meaningless tokens, failing to separate roots from affixes. Moreover, Wordpiece does not take into account fine-grained character-level information. We hypothesize that modeling fine-grained character-level information or interactions between roots and affixes helps with modeling highly inflected and morphologically complex languages such as Bengali and Hindi. We used BERT with two different tokenizers - a Unigram tokenizer and a character-level tokenizer and observed better performance. Then, we pretrained four language models accordingly - Bengali Unigram BERT, Hindi Unigram BERT, Bengali Character BERT, and Hindi Character BERT, and evaluated them for masked token detection, both in correct and erroneous settings, across many NLU tasks. We provide experimental evidence that Unigram and character-level tokenizers lead to better pretrained models for Bengali and Hindi, outperforming the previous state-of-the-art and BERT with Wordpiece vocabulary. We conduct the first study investigating the efficacy of different tokenization methods in modeling Bengali and Hindi.

Keywords: Language model, NLU, Low-resource languages, Self-supervised Pretraining, Tokenizer, BERT

1. Introduction

Bengali and Hindi are the sixth and fourth most spoken languages globally, with 234 and 345 million native speakers, respectively (Wikipedia). Speakers of these languages contribute to 7% of the world's population. Bengali is the national language of Bangladesh and the official language of two Indian states. Hindi is spoken mainly in northern India and is considered the common language of the Hind Belt region, covering most of India. Despite their popularity, these languages remain under-represented in computational resources, especially labeled datasets. Therefore, self-supervised pretraining is the best way towards useful and large language models for such languages (Joshi et al., 2020).

State-of-the-art monolingual BanglaBERT and multilingual BERT (Bhattacharjee et al., 2022; Devlin et al., 2019) trained Wordpiece tokenization system for modeling Bengali and Hindi. Both systems rely on the original Wordpiece tokenizer, which, to the best of our knowledge, was not developed with these languages in mind. In fact, we observed that Wordpiece splits words into tokens that have no meaning in isolation and fails to separate roots, suffixes, and prefixes. Thus, the Wordpiece tokenizer often fails to split words into natural tokens. Instead, it greedily selects the longest subword unit from the beginning and then repeats the same process until the end of the word. We found many situations where Wordpiece would pick a common root for its vocabulary during trainig while failing to use that token when encountering said root in a compounded and/or inflected form. One example was the root ‘পরিবহন’ (transport); when Wordpiece was faced with ‘গণপরিবহনের’ (of public transport), the resulting tokens would be sub-words [‘গণপ’, ‘রি’, ‘বহ’, ‘নের’] that did not include a root, suffix, or prefix. We ran into many other examples where Wordpiece produced “unnatural” tokens.

Motivated by that observation, we carefully considered design choices for building NLU models for Bengali and Hindi, including evaluating different tokenization methods to accommodate shared characteristics of these languages that make them different than high-resource languages like English. Bengali and Hindi originate from Sanskrit (Staal, 1963) and share high, non-linear inflections (Bhattacharya et al., 2005) and morphological complexity. Modified vowels, consonants, and many compound characters lead to morphological complexity. Moreover, the same root can have many inflected forms depending on adding different suffixes and prefixes (Rahman et al., 2022a).

Believing that better modeling fine-grained character-level information or interactions between roots and suffixes or prefixes would result in better models, we modified the BERT architecture with two different tokenizers - Unigram tokenizer and character-level tokenizer and observed better performance empirically. Then, we pretrained and evaluated language models based on these tokenizers on masked token detection, both in correct and in erroneous settings, among other common NLU tasks.

LREC-COLING 2024, pages 8719–8731
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Contributions

1. We show that BERT models using the Unigram tokenizer (Kudo, 2018) outperform BERT models with Wordpiece tokenizer for various tasks in both Bengali and Hindi.
2. We also show that a character-level tokenizer based on CharacterBERT (El Boukkouri et al., 2020), which is well suited to deal with compound characters and better learn intra and inter-word patterns by consulting characters in each word, also outperforms previous BERT models using Wordpiece.
3. We release the four pretrained models we experimented with – two for Bengali and two for Hindi – and achieved new state-of-the-art at the various benchmarks for language understanding.

2. Literature Review

Careful design choices for language models (LM) help achieve better performance in high-resource settings. Such improvements motivate us to identify overlooked design choices for low-resource languages like Bengali and Hindi.

Transformer-based (Vaswani et al., 2017) models with self-attention mechanisms remain state-of-the-art for language modeling tasks. GPT (Radford et al., 2019) adopted a generative pretraining objective to learn generalizable universal text representations from a large unlabeled corpus. The final aim of pretraining was to transfer learned knowledge to various downstream tasks. BERT (Devlin et al., 2019) introduced a masked language modeling (MLM) approach, which improved GPT's auto-regressive pretraining technique and leveraged bidirectional context for predicting masked tokens.

RoBERTa (Liu et al., 2019) reimplemented BERT, thoroughly examined the impact of hyperparameter tuning and training set size, and concluded that BERT was undertrained. To address this issue, they used more data for training with longer sequences and larger batch sizes and applied a dynamic masking strategy. Like RoBERTa, El Boukkouri et al. (2020) proposed a variant of BERT named CharacterBERT, employing ELMO's (Peters et al., 2018) character-level CNN module to deal with the vocabularies of specialized domains like medical domain. Clark et al. (2020) improved the BERT pretraining objective using a generator and discriminator model, which allowed the model to learn faster from all tokens in the entire input sequence rather than a few masked tokens.

The lack of specialized LMs for low-resource languages like Bengali and Hindi forces NLP researchers working on downstream tasks (Ashrafi et al., 2020; Islam et al., 2021) to resort to fine-tuning multilingual pretrained language models (PLM). Devlin et al. (2019) released a multilingual BERT model. It was pretrained on a multilingual corpus obtained by concatenating Wikipedia pages encompassing 104 languages. Their model has a shared vocabulary that covers those 104 languages. Thus, the Wordpiece tokenizer was trained on all the languages. A RoBERTa-based multilingual model, XLM-RoBERTa (Conneau et al., 2020), was also pretrained with MLM objective on more than 2 TB of filtered common crawl data encompassing 100 languages. They used a vocabulary size of 250K and trained two different models, XLM-R base and XLM-R large. IndicBERT (Kakwani et al., 2020b) is another multilingual PLM developed for 11 major Indian languages. These multilingual models cover a wide range of languages. They are larger models, requiring more computational cost to fine-tune for target tasks due to their larger shared vocabulary size and increased model capacity.

There are very few models specific to Bengali and Hindi. Recently, Bhattacharjee et al. (2022) proposed a Bengali NLU model Bangla-BERT based on BERT. They pretrained the model using ELECTRA's (Clark et al., 2020) replaced token detection objective on a 27.5GB corpus crawled from popular websites. Like other BERT-based models developed for high-resource languages, they trained a Wordpiece tokenizer for modeling morphologically rich Bengali language. Moreover, Rahman et al. (2022a) have presented an analysis of different architectures like convolutional, recurrent, and transformer-based neural networks. They concluded their CNN-based CoCNN model could outperform other competitive architectures, like SOTA transformers for Bengali and Hindi. Although their work attempted to address the specific characteristics of Bengali and Hindi, the CoCNN model cannot be pretrained. Instead, it needs to be trained end-to-end for each downstream task. Therefore, it would not be able to transfer knowledge from large unlabeled corpora but only be limited to very few labeled datasets available for such low-resource languages.

3. Tokenizers

Tokenizers heavily influence transformer-based language models because the text encoding method determines how the language model will perform in various language understanding tasks. Therefore, designing an LM for a language requires a lot of research to choose a tokenizer well suited for the specific language characteristics. The language model cannot learn meaningful representation without understanding the language structure. The main objective is to find the most meaningful representations that represent the language well and make sense to the LM. We observe that BERT's original Wordpiece tokenizer cannot produce state-of-the-art results for Bengali and Hindi. Thus, we propose modifications to that model using two different tokenizers - Unigram tokenizer and character-level tokenizer, that improve end-task performance.

3.1. Bengali and Hindi Unigram BERT

Unigram (Kudo, 2018) is a subword tokenization algorithm. Unlike Wordpiece (Wu et al., 2016), it begins with a larger vocabulary and works with a top-down approach to iteratively reduce it to the final vocabulary. Before training the tokenizer, we apply normalization steps, including a few replacements and NFKC Unicode normalization. Two or more whitespaces were replaced with a single space like SentencePiece (Kudo and Richardson, 2018) algorithm. We also preserve the accents since vowel matras and diacritics are frequently used in Bengali and Hindi.

We use a Metaspace pre-tokenizer to replace single whitespaces with a specific character ('_') for the pre-tokenization step. We start with an initial seed vocabulary much larger than the target vocab size (30,522). Seed vocabulary includes all basic characters and the most common substrings (top 35%) obtained from the corpus. We include all possible characters besides the most common substrings because the model cannot otherwise tokenize potential out-of-vocabulary words. Out-of-vocabulary words occur when the tokenizer encounters words with subword tokens not present in the vocabulary.

At each iteration, the corpus-level loss was computed for the current vocabulary. Each word in the corpus was tokenized using the available vocabulary for calculating loss. The unigram model computed the probability for each subword token and multiplied all subword tokens' probability to get word-level probabilities. Like the Unigram language model, each token is considered independent of the previous tokens. Hence, each token's probability can be computed by dividing the token frequency by the sum of all tokens' frequencies in the corpus. During word tokenization, the Unigram model considers the best segmentation of the word into sub-word tokens with the highest probability. This best segmentation is efficiently done using the Viterbi (Viterbi, 1967) algorithm. As we consider tokens independent, word probability is the product of sub-word token probabilities. Afterward, word probabilities are multiplied by the word frequency to get final scores. Finally, the corpus-level loss is determined by applying the negative log-likelihood of these scores following Kudo (2018).

The training is based on an expectation maximization (EM) algorithm. It determines how much the loss will be increased for removing each token from the current vocabulary. At each step, we discard *p*% subword tokens having the most negligible impact on corpus-level loss. The *p* value for discarding tokens was selected based on corpus-level loss. We repeat these steps until we reach the desired 30.5K vocabulary size with scores to find the most probable splits into tokens. Therefore, the algorithm preserved the tokens mostly needed for tokenization purposes and dropped less needed ones belonging to the tail end of the distribution. During training, elementary tokens are not discarded, as the model cannot tokenize every word without all possible characters. Since our NLU model is based on BERT, we used similar special tokens ([UNK], [PAD], [CLS], [SEP], [MASK]) to indicate unknown, padding, classification, separator, and mask tokens.

3.1.1. Preliminary Experiments

We trained the Unigram tokenizer on Bengali pretraining data and compared it with the original BERT's Wordpiece tokenizer. We tokenized a few sample words and examined the differences (see Table 1).

Looking at the quality of sub-word tokens, we observed that the Unigram tokenizer always utilizes learned scores to find the most likely splits into tokens. It can separate both emphasizing suffixes ('ও') and modified vowels ('ে') in the first and second examples. However, the Wordpiece tokenizer fails to do that. In the third and fourth examples, the Unigram tokenizer can separate the roots ('অকপট', 'বিনম্র') from suffixes ('ভাবে','তা') in their inflected forms. In addition, Unigram tokenizer can separate the affix ('ের') in the second last example. This affix produces an inflectional form of the noun to denote a relationship with the next word in a sentence. Nevertheless, the Wordpiece tokenizer cannot identify roots, suffixes, or prefixes during word segmentation. In the last example, the Unigram tokenizer produces meaningful sub-word tokens ('অ', 'কাজ'). Here, the first prefix ('অ') indicates negation, thus negating the meaning of the next token ('কাজ'). On the contrary, in the last three examples, the Wordpiece tokenizer unnecessarily decomposes words into meaningless word pieces ('অক', 'াজ').

3.2. Bengali and Hindi Character BERT

Character-level tokenizer comprises a CNN module that produces a contextless token-level representation before feeding to the BERT model. It takes input as a sequence of characters and

Reference word	Unigram tokenizer	Wordpiece tokenizer
শিক্ষাপ্রতিষ্ঠানেও (also in educational institutions)	[শিক্ষাপ্রতিষ্ঠান, ে·, ও] [(educational institutions), (in), (also)]	[শিক্ষাপ্রতিষ্ঠানে, ও] [(in educational institutions) , (also)]
ভারোত্তোলনে (in weight-lifting)	[ভারোত্তোলন, ে·] [(weight-lifting), (in)]	[ভারোত্তোল, নে] [(-), (-)]
অকপটভাবে (frankly)	[অকপট, ভাবে] [(frank), (way)]	[অক, পট, ভাবে] [(_), (_), (way)]
বিনম্রতা (modesty)	[বিনম্র, তা] [(humble), (being)]	[বিন, ম্, রত, া] [(_), (_), (_), (_)]
গণপরিবহনের (of public transport)	[গণপরিবহন, ে·র] [(public transport), (of)]	[গণপ, রি, বহ, নের] [(_), (_), (_), (_)]
অকাজ (useless act)	[অ, কাজ] [(useless), (act)]	[অক, ·াজ] [(_). (_)]

Table 1: Comparison of Unigram tokenizer with Wordpiece Tokenizer. (_) indicates the translation of a token without meaning that does not indicate any root, suffix, prefix, or meaningful unit.

embeds each character into a fixed-sized *d*-dimensional vector representation. These embeddings are sequentially fed into seven 1D CNN layers with various filter sizes. The outputs of CNN layers are max-pooled and concatenated to build token-level representations. These concatenated representations go through Highway layers (Srivastava et al., 2015) incorporating nonlinearities with residual connections. Finally, outputs are projected to 768 dimensions, similar to BERT's embedding size. These token-level embeddings are fed to BERT's 12 encoder layers to produce contextual representations. Figure 1 depicts how the CNN module constructs token-level representation after attending to each character in the sequence.

In Bengali and Hindi, characters can combine to form modified vowels, consonants, and compound characters. The character-level tokenizer can model these intra-word interactions to learn fine-grained character-level information. Adding a single character, modified vowel, consonant, or compound character can lead to a different word. Thus, the original BERT's Wordpiece tokenizer will produce different sub-word tokens that can impact downstream task performance. For example, if we add a single character 'অ' at the beginning of the word 'কাজ' (act), it will become 'অকাজ' (useless act). The single character added here is a prefix that indicates the negation of the word 'কাজ' (act). The Wordpiece tokenizer will tokenize it into other meaningless subwords ['অক', '·াজ'] depending on the corpus, even if the meaningful subword unit 'কাজ' (act) is present in its vocabulary.

We also include examples of how Wordpiece tokenizer might inaccurately split words with complex compound characters, a common feature in Bengali and Hindi. For example, Wordpiece breaks the word 'অনাবিষ্কৃত' (undiscovered) into meaningless subwords 'অনাব', 'ি·ষ', '·্কৃত'. Similarly, it splits the word 'ক্ষয়িষ্ণু' (decaying) into tokens 'ক্ষয়', 'ি·ষ', '·্ণ', '·ু'.

Moreover, the character-level tokenizer consults characters to produce a word-level representation. Therefore, it can have an open vocabulary not limited to Wordpiece vocabulary. Besides injecting character-level information, Bengali and Hindi Character BERT can learn local inter-word dependencies at the sentence level using encoder layers. Following the original BERT, we used padding, separator, classification, and mask tokens as special tokens and added positional embedding to the token embeddings. For MLM, we have constructed a temporary vocabulary comprising the top 30,522 tokens from the pretraining corpus. These tokens have been used as target labels.

4. Pretraining

4.1. Pretraining Objective

We pretrain the models with the masked language modeling (MLM) objective. During pretraining, 15% of the tokens are randomly selected for masking. The selected tokens are replaced using the [MASK] token. MLM cross-entropy loss was defined using the masked token prediction task. In addition, we randomize the masking pattern every time we feed a batch of sequences to the model. Random masking enables the model to learn efficiently from as many tokens as possible without duplicating data samples. We pretrained the Bengali and Hindi Character BERT and Unigram BERT models on each language separately. However, we do not train models on the next-sentence prediction (NSP) task, as the authors of the RoBERTa (Liu et al., 2019) paper show that removing next-sentence prediction loss does not hurt the BERT model's performance on downstream tasks.

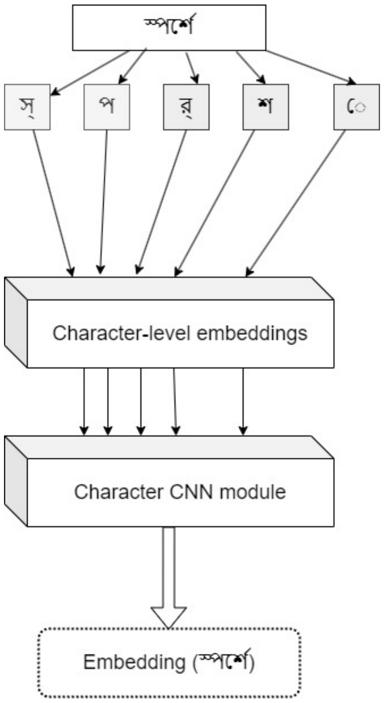

Figure 1: CNN module produces token-level representation after attending each character. Modified vowels and compound characters are highlighted in Grey color.

4.2. Model Architecture & Hyperparameters

Our models are based on the BERT base model and CharacterBERT model. Both models use a 12-layer encoder with 12 attention heads and 768 embedding sizes. We pretrained the models for 803,640 steps with a 256 batch size. We use 128 maximum sequence length due to GPU memory constraints. For optimization, we use the Adam optimizer (Kingma and Ba, 2015) with a 2e-5 learning rate, L_2 weight decay of 0.01. We linearly decay the learning rate with training steps. The Bengali and Hindi Unigram BERT model has 110M parameters. Moreover, the Bengali and Hindi Character BERT model has 104.6M parameters due to the CNN module's smaller character embeddings (16 dimensional).

5. Experiments

5.1. Datasets

5.1.1. Pretraining Datasets

Pretraining transformer-based model requires a large quantity of good-quality data. We used open-access data sets (Rahman et al., 2022b) that represent the contemporary Bengali and Hindi writing styles. In addition, we select data sources that cover various topics. Many noisy and unauthentic data sources, including offensive text and erroneous facts, are available. Therefore, most pretraining data were collected from sources such as online news portals, known as authentic data sources with minimum offensive texts.

Finally, we pretrained on articles from online news portals and Wikipedia articles. Bengali pretraining data was collected from Prothom Alo (between 2014 and 2017) (Rahman, 2017) and BDNews articles (between 2015 and 2017) (Khalidi, 2017). Hindi pretraining corpus encompasses Wikipedia articles (Gaurav, 2019), Hindi Oscar corpus (Thakur, 2019), HindiEnCorp 0.5 (Bojar et al., 2014) dataset, WMT Hindi news crawl data (Barrault et al., 2019). Since it is a sentence-level corpus, all documents were split into sentences. The Bengali and Hindi pretraining corpus had 6.69M and 8.57M samples, respectively, with a maximum sequence length of 128 tokens after tokenization.

Corpus	Training	Validation
BDNews24	446,984	111,747
Prothom Alo	1,080,000	270,000
Naya Diganta	-	100,000
Livehindustan	187,077	46,770
Hindi Patrika	-	100,000

Table 2: Model quality analysis dataset statistics (the number of training and validation samples). Naya Diganta and Hindi Patrika were entirely used for validation purposes.

5.1.2. Datasets For Model Quality Analysis

We experiment with publicly available datasets (Rahman et al., 2022b) for model quality analysis – three Bengali and two Hindi datasets. (see table 2). We chose newspaper articles to verify the effectiveness of models across various domains. Domains include politics, technology, sports, lifestyle, literature, and entertainment. Bengali datasets include online news website articles collected from Prothom Alo (Rahman, 2017), BDNews24 (Khalidi, 2017), and Naya Diganta (Mohiuddin, 2019). The Hindi datasets are also from online news portals – Hindi News LiveHindustan (Shekhar, 2018) and Patrika (Jain, 2018). Following Rahman et al. (2022b), we use two Bengali and Hindi datasets-Naya Diganta and Patrika as test sets.

We also tested our models with misspelled words. To do that, we replaced words in the Prothom Alo dataset with common misspellings identified by Rahman et al. (2022b). First, we randomly select a certain percentage of sentences in the validation set. Then, we transform the words in the selected sentences with their misspelled versions.

5.2. Performance Metrics

We use perplexity (PPL) obtained from the masked token prediction task to assess the quality of the model. To calculate the perplexity, we randomly select tokens from the sequences in the corpus for masking. Then, we compute the probability P of the model predicting token given the masked sequence. We calculate perplexity by multiplying the probabilities and dividing by the number of tokens. We consider log of probabilities to avoid numerical underflow. This PPL calculation approach is better suited to the BERT model than the next token prediction task as it measures the ability of the model to predict missing tokens anywhere in the sequence.

For downstream tasks such as text classification, named entity recognition, sentiment analysis, and natural language inference (NLI), we use F1 score and accuracy as performance metrics.

5.3. Model Training & Evaluation

For the masked token prediction task, we trained models with the same vocabulary size (30,522) for 24 epochs. We used 80% of the data for training and reported perplexity on 20% of validation data. We optimized the models using Adam optimizer with 2e-5 learning rate and decayed learning rate with linear learning rate scheduler. We used a batch size of 64 and trained the models using two GPU servers – one 16GB Tesla V100 GPU (51GB RAM) and another 11GB Nvidia GeForce RTX 2080 TI GPU (131GB RAM). For experimentation, we used frameworks and libraries like Transformers (Wolf et al., 2020a), Pytorch (Paszke et al., 2019), and Scikit-learn (Pedregosa et al., 2011).

5.4. Fine-tuning & Evaluation

We fine-tuned models for eight diverse downstream tasks in Bengali and Hindi. The fine-tuning was performed for 2-6 epochs, and learning rate was tuned from 1e-5 to 5e-5 range with a weight decay of 0.01. We used Adam optimizer and tuned the learning rate warmup ratio from 0 to 10% of the total steps. The batch size was chosen from {16, 32}, and we use a batch size of 32 for most of the tasks. The maximum length was restricted to 128 tokens for the fine-tuning experiments. We fine-tune pretrained models independently for each task.

We select the best model based on validation performance for public datasets with given test and validation splits. Then, we use the official test splits to report the final results. In addition, we perform a five-fold cross-validation for the data sets without an official test set and report mean results.

5.4.1. Downstream Tasks

We fine-tuned models on existing Bengali and Hindi benchmark datasets (Rahman et al., 2022b; Kakwani et al., 2020b) comprising basic NLU tasks. Basic NLU task includes question classification, sequence labeling, text classification, and sequence pair classification.

Question Classification The Bengali Question Classify dataset (Islam et al., 2016) consists of 3,330 question samples and six classes. The task is to classify a question into one of the six classes – numeric, human, location, abbreviation, entity, and descriptive type question.

Named Entity Recognition We chose the largest public NER dataset for Bengali (Karim et al., 2019). The dataset contains 71 thousand sentences properly annotated using the IOB tagging scheme. It was annotated using four course-grained tags: person (PER), location (LOC), organization (ORG), and object (OBJ) entity.

Article Genre Classification A news article is provided as input, and the task is to classify the article's genre or topic. We select two Bengali and Hindi datasets - Soham News Articles (Chatterjee, 2019) and BBC News (Kakwani et al., 2020b) with 14,106 and 4,333 articles, respectively.

Hate Speech Detection The task is to identify whether a post on social media contains hate speech and offensive content. Hindi Hate Speech dataset (HASOC, 2019) comprises 3,664 posts with binary classes - hate and offensive (HOF) and Non-hate and offensive (NOT).

Sentiment Analysis We used two Hindi datasets for sentiment analysis. The Hindi Product Review dataset (Rahman et al., 2022b) consists of 2,355 reviews. The task is to classify a review into positive or negative classes. Moreover, IIT-Patna movie (3,100 samples) (Akhtar et al., 2016) reviews can be categorized into three polarities- positive, negative, or neutral.

Natural Language Inference For NLI, we use the Choice of Plausible Alternative (COPA) task (Gordon et al., 2012). This task verifies models' capability for open-domain commonsense reasoning. We use COPA's Hindi-translated version (Kakwani et al., 2020b) with 899 multiple-choice questions on causal reasoning. The question serves as a premise, and the objective is to choose an alternative with a more plausible causal relation (cause or effect) to the premise.

Dataset	Error	Without Tokenizer	With Tokenizer	Improvement
BDNews	No	100.22	**48.60**	+51.50%
Prothom Alo	No	44.82	**26.05**	+41.88%
	10%	52.57	**29.27**	+44.32%
	20%	61.57	**32.95**	+46.48%
	30%	72.13	**37.15**	+48.50%
Naya Diganta	No	81.06	**45.78**	+43.52%

Table 3: Improvement resulted from Bengali Unigram BERT compared to baseline BERT.

Dataset	Error	Without Tokenizer	With Tokenizer	Improvement
Live Hindustan	No	56.88	**23.27**	+59.10%
Hindi Patrika	No	92.03	**35.77**	+61.13%

Table 4: Improvement resulted from Hindi Unigram BERT compared to baseline BERT.

Dataset	Error	Without Tokenizer	With Tokenizer	Improvement
BDNews	No	100.22	**73.71**	+26.45%
Prothom Alo	No	44.82	**34.51**	+23.00%
	10%	52.57	**36.16**	+31.22%
	20%	61.57	**37.81**	+38.59%
	30%	72.13	**39.77**	+44.86%
Naya Diganta	No	81.06	**48.87**	+39.71%

Table 5: Improvement resulted from Bengali Character BERT compared to baseline BERT.

Dataset	Error	Without Tokenizer	With Tokenizer	Improvement
Live Hindustan	No	56.88	**42.99**	+24.42%
Hindi Patrika	No	92.03	**56.35**	+38.77%

Table 6: Improvement resulted from Hindi Character BERT compared to baseline BERT.

Discourse Analysis MIDAS Discourse dataset (Dhanwal et al., 2020) has 9,968 sentences from Hindi stories and is annotated with five discourse modes - descriptive, narrative, dialogue, argumentative, informative, and others. The task is to identify the modes of discourse at the sentence level.

Cloze-style Multiple Choice QA Cloze-style multiple choice QA (Kakwani et al., 2020b) evaluates whether an LM can serve the purpose of a knowledge base. 38,845 article samples are collected from Bengali Wikipedia. An entity is randomly masked in the article. The task is to predict the masked entity out of four possible candidates.

6. Results & Discussion

6.1. Comparison With Baseline Model

We trained original BERT and proposed models on five Bengali and Hindi datasets. We compared their PPL scores on a validation set in correct and erroneous settings. Perplexity and improvements are reported in tables 3, 4, 5, and 6. We report improvement relative to the baseline. Eight sets of PPL scores are used to perform a one-tailed paired t-test, and results in bold text indicate a statistically significant difference ($p < 0.005$). The PPL scores in the tables show that the Bengali and Hindi Unigram BERT and Character BERT outperform baseline BERT with Wordpiece vocabulary.

We also experiment with whether our pretrained models can deal with misspellings. Hence, we created noisy versions of the Prothom Alo validation set with incremental changes in noise levels. We transform words in *p*% sentences to create erroneous versions of the validation set. The misspelled words are formed by adding, removing, replacing characters, modifying vowels and consonants, or producing inflected forms of words by adding affixes.

With incremental changes in error percentage, the perplexity of Bengali and Hindi Unigram BERT and Character BERT increases slowly compared to the baseline BERT. In particular, Bengali and Hindi Character BERT outperform original BERT by a large margin in erroneous settings. Although the relative improvement is 23% in the correct validation set, the improvement becomes greater than 44% when a noise level of 30% is applied to the

Dataset	Original BERT	Bengali Unigram BERT	Bengali Character BERT
Question Classify	90.50	**97.22**	96.48

Table 7: Comparison of F1 score between proposed pretrained models and original BERT in Bengali downstream tasks. Original BERT results are from (Rahman et al., 2022a)

Dataset	Original BERT	Hindi Unigram BERT	Hindi Character BERT
Hate Speech Detection	77.00	**82.93**	82.77
Product Reviews	84.10	**89.57**	87.23
Average	80.55	**86.25**	85.00

Table 8: Comparison of F1 score between proposed pretrained models and original BERT in Hindi downstream tasks. Original BERT results are from (Rahman et al., 2022a)

Dataset	Indic BERT	Multilingual BERT	Bengali Unigram BERT	Bengali Character BERT
Named Entity Recognition	62.42	64.54	**71.49**	69.87
Soham News Article	78.45	80.23	91.28	**91.92**
Cloze-style QA	39.40	36.23	**56.16**	40.51
Average	60.09	60.33	**72.97**	67.43

Table 9: Accuracy comparison between proposed pretrained models and multilingual models in Bengali downstream tasks, except NER task, which compares F1 score. NER result, and the rest of the results for multilingual BERT and IndicBERT are published in (Ashrafi et al., 2020) and (Kakwani et al., 2020b), respectively.

Dataset	Indic BERT	Multilingual BERT	Hindi Unigram BERT	Hindi Character BERT
BBC News Classification	74.60	60.55	**76.67**	76.09
IITP Movie Reviews	59.03	56.77	**66.77**	63.87
Midas Discourse	78.44	71.20	**81.44**	79.44
COPA	51.22	54.78	**60.80**	56.12
Average	65.82	60.83	**71.42**	68.88

Table 10: Accuracy comparison between proposed pretrained models and multilingual models in Hindi downstream tasks. The multilingual BERT and IndicBERT results are published in (Kakwani et al., 2020b).

Prothom Alo validation set. So, the relative improvement almost doubles, showing Bengali and Hindi Character BERT's advantage in erroneous settings. Bengali and Hindi Unigram BERT can also better adapt to noisy and inflected settings, as the relative improvement increases to 48% in the noisiest version from 41% in the correct Prothom Alo validation set.

6.2. Comparison in Downstream Tasks

We fine-tuned pretrained Bengali and Hindi Character BERT and Unigram BERT on three downstream tasks. We compared them with original BERT pretrained on the same pretraining data (Rahman et al., 2022a). There are no publicly available standard test splits for question classification, hate speech detection, and product review datasets. Thus, we perform 5-fold cross-validation to report mean F1 scores. The mean results in table 7 and 8 show that Bengali and Hindi Character BERT and Unigram BERT achieve robust performance over the original BERT in three downstream tasks. The Bengali and Hindi Unigram BERT model significantly improved over the original BERT. Moreover, character-aware Bengali and Hindi Character BERT marginally lag behind Bengali and Hindi Unigram BERT in three tasks.

We compare our pretrained models with two multilingual models. Table 9 and 10 compares published multilingual BERT and IndicBERT (Kakwani et al., 2020b) results with our pretrained models. However, the gold labels for the COPA test set are not publicly available. Therefore, we collected the test set with gold labels from the English COPA dataset (Roemmele et al., 2011). Then, we translated them to Hindi using manually translated annotations (Kakwani et al., 2020a). We fine-tuned pretrained multilingual BERT and IndicBERT on the COPA dataset and reported results on the translated test set. Previous work (Ashrafi et al., 2020) only published multilingual BERT results for the NER task. Hence, we fine-tuned IndicBERT on the NER dataset and reported results on the standard test set. We report macro-F1 for the NER task and accuracy for the other tasks.

Bengali and Hindi Unigram BERT outperform multilingual BERT and IndicBERT, achieving average scores of 72.97 and 71.42, respectively. Bengali and Hindi Character BERT outperform multilingual BERT and IndicBERT, achieving average scores of 67.43 and 68.88, respectively. Our pretrained models' performance is more robust for four text classification tasks than for other tasks. For challenging tasks like COPA and Cloze-style QA, our models, especially Bengali and Hindi Unigram BERT, improve by a large margin of up to 9 points and 19 points in COPA and QA datasets, respectively. For Wikipedia-based datasets like Clozed-style QA, our pretrained models can improve task performance over multilingual BERT pretrained on the Wikipedia corpus. Bengali and Hindi Character BERT and Unigram BERT are comparable, but Unigram BERT is consistently better in most tasks.

7. Error Analysis

We also experimented with how the Unigram tokenizer performs in scenarios where Benali and Hindi have multiple morphological forms, such as classifier suffixes and affixes indicating different verb tenses. For example, the Unigram tokenizer can separate classifier suffixes 'টি', 'গুলো', 'খানা' from roots 'বালক', 'আম, 'কাগজ' in words 'বালকটি' (the boy), 'আমগুলো' (mangoes), 'কাগজখানা (the papers), respectively. Here, the classifier suffixes indicate singular, specific, and concrete forms of nouns. Moreover, it can break down words 'হারছিলাম' (was losing), 'হেরেছিলাম' (lost), 'হারতাম' (used to lose) into suffixes 'ছিলাম', 'তাম', and roots 'হার', 'হেরে'. Here, the affixes indicate past progressive tense, past perfect tense, and simple past tense, respectively, and the root shows an example of nonlinear inflection depending on the addition of affix and tense.

We also investigated failure cases, specifically those the unigram and character level tokenizer fail to capture instead of general BERT problems. The Unigram tokenizer fails to break down transliterated words properly. For example, 'ইউনিফাইড' is the transliteration of the word 'Unified' in English. In the NER dataset, this word appears as the sports team's name, 'ৰাংলাদশে ইউনিফাইড দল' (Bangladesh Unified Team). Another challenge for the tokenizer was tokenizing multi-word expressions. For example, the Unigram tokenizer cannot properly tokenize the multi-word expression 'উত্তর-পশ্চিমাঞ্চলের' (north-western region). Nevertheless, it can properly tokenize each word of the multi-word expression when it occurs in isolation. Moreover, both the tokenizer fails to properly tokenize English words appearing frequently in the Hate Speech dataset. Since the observations are collected from social media, it is common for social media posts to have mixed script text. However, our tokenizers and models are pre-trained mainly on monolingual data, which does not contain mixed scripts like English and Hindi. For example, Bengali and Hindi Unigram BERT does not have all English alphabets in its vocabulary, thus leading to an unknown ([UNK]) token. Although Bengali and Hindi characters BERT do not rely on fixed vocabulary, such mixed script data was not seen during training.

8. Conclusion & Future Work

To address the specific needs of Sanskrit-originated languages, we have presented Bengali and Hindi Unigram and character-level tokenizers and pretrained models. We show that these tokenizers are well suited for low-resource Bengali and Hindi languages by demonstrating improved performance in a diverse set of downstream tasks. Both tokenizers, especially Bengali and Hindi character-level tokenizer, show robustness over the original BERT tokenizer in highly erroneous and inflected settings.

In future research, we plan to run experiments on more Bengali and Hindi NLU benchmarks and pretrain models on more data with longer sequences. We can use a more efficient pretraining objective like replaced token detection or span masking instead of tokens. Fusing both tokenizers or applying similar tokenization strategies for Bengali and Hindi natural language generation models is also possible. It will also be an interesting avenue to explore how the tokenizer performs if built upon an N-gram language model instead of a Unigram language model. Future research can apply and compare similar tokenizers for more recent and larger NLU models (He et al., 2021; Liu et al., 2019).

9. Ethical considerations

We carefully considered data sources with minimum offensive texts for the pretraining corpus. However, unpleasant content like objectionable text and sociocultural or stereotypical biases might exist. Such biases can contribute to biased word representations and have a negative impact, especially for text generation purposes.

10. Code Availability

Our code and pretrained models are available at `https://github.com/arif-shahriar-anik/Carefully-Chosen-Transformer-Architecture-Improves-Sanskrit-Originated-Language-Modeling`.

11. Bibliographical References

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan. The COLING 2016 Organizing Committee.

Imranul Ashrafi, Muntasir Mohammad, Arani Shawkat Mauree, Galib Md. Azraf Nijhum, Redwanul Karim, Nabeel Mohammed, and Sifat Momen. 2020. Banner: A cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Samit Bhattacharya, Monojit Choudhury, Sudeshna Sarkar, and Anupam Basu. 2005. Inflectional morphology synthesis for bengali noun, pronoun and verb systems. In *Proc. of the National Conference on Computer Processing of Bangla (NCCPB 05)*, pages 34–43.

Atira S Bick, Gadi Goelman, and Ram Frost. 2011. Hebrew brain vs. english brain: Language modulates the way it is processed. *Journal of Cognitive Neuroscience*, 23(9):2280–2290.

Ondřej Bojar, Vojtěch Diatka, Pavel Straňák, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp 0.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Soham Chatterjee. 2019. Classification: Bengali news articles (indicnlp).

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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. *Advances in neural information processing systems*, 32.

Pieter Delobelle and Bettina Berendt. 2019. Time to take emoji seriously: They vastly improve casual conversational models.

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.

Swapnil Dhanwal, Hritwik Dutta, Hitesh Nankani, Nilay Shrivastava, Yaman Kumar, Junyi Jessy Li, Debanjan Mahata, Rakesh Gosangi, Haimin Zhang, Rajiv Ratn Shah, and Amanda Stent. 2020. An annotated dataset of discourse modes in Hindi stories. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1191–1196, Marseille, France. European Language Resources Association.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gaurav. 2019. Hindi wikipedia articles - 172k.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Wei Guo and Aylin Caliskan. 2021. Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 122–133, New York, NY, USA. Association for Computing Machinery.

HASOC. 2019. Hasoc: hindi hate speech dataset.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md. Aminul Islam, Md. Fasihul Kabir, Khandaker Abdullah-Al-Mamun, and Mohammad Nurul Huda. 2016. Word/phrase based answer type classification for bengali question answering system. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 445–448.

Bhuwnesh Jain. 2018. Rajashthan patrika epaper:hindi.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020a. Indicglue datasets.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020b. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

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

Toufique Imrose Khalidi. 2017. Bd news articles.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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, Melbourne, Australia. Association for Computational Linguistics.

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

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen Mckeown, and Tatsunori B Hashimoto. 2023. When do pretraining biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European*

Chapter of the Association for Computational Linguistics, pages 3198–3211.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Alamgir Mohiuddin. 2019. Daily nayadiganta.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Valeria A. Pfeifer, Emma L. Armstrong, and Vicky Tzuyin Lai. 2022. Do all facial emojis communicate emotion? the impact of facial emojis on perceived sender emotion and text processing. *Computers in Human Behavior*, 126:107016.

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.

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

Chowdhury Rahman, MD. Hasibur Rahman, Mohammad Rafsan, Mohammed Eunus Ali, Samiha Zakir, and Rafsanjani Muhammod. 2022a. CNN for modeling Sanskrit originated Bengali and Hindi language. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 47–56, Online only. Association for Computational Linguistics.

Chowdhury Rafeed Rahman, MD. Hasibur Rahman, Mohammad Rafsan, Samiha Zakir, Mohammed Eunus Ali, and Rafsanjani Muhammod. 2022b. CNN for Modeling Sanskrit Originated Bengali and Hindi Language Dataset.

Matiur Rahman. 2017. Prothom alo bangla news paper.

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.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Shashi Shekhar. 2018. Hindi news livehindustan.

Manjira Sinha, Tirthankar Dasgupta, and Anupam Basu. 2016. Effect of syntactic features in bangla sentence comprehension. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 275–284.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks.

Johan Frederik Staal. 1963. Sanskrit and sanskritization. *The Journal of Asian Studies*, 22(3):261–275.

Abhishek Thakur. 2019. Hindi oscar corpus.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

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.

Wikipedia. List of languages by number of native speakers — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=List_of_languages_by_number_of_native_speakers&oldid=1175765045`. [Online; accessed 9-October-2023].

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. 2020a. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020b. Huggingface's transformers: State-of-the-art natural language processing.

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 preprint arXiv:1609.08144*.

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 *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.