TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/10 pp709–718
DOI: 10.26599/TST.2021.9010060
Volume 27, Number 4, August 2022

Pretrained Models and Evaluation Data for the Khmer Language

Shengyi Jiang, Sihui Fu, Nankai Lin*, and Yingwen Fu

Abstract: Trained on a large corpus, pretrained models (PTMs) can capture different levels of concepts in context and hence generate universal language representations, which greatly benefit downstream natural language processing (NLP) tasks. In recent years, PTMs have been widely used in most NLP applications, especially for high-resource languages, such as English and Chinese. However, scarce resources have discouraged the progress of PTMs for low-resource languages. Transformer-based PTMs for the Khmer language are presented in this work for the first time. We evaluate our models on two downstream tasks: Part-of-speech tagging and news categorization. The dataset for the latter task is self-constructed. Experiments demonstrate the effectiveness of the Khmer models. In addition, we find that the current Khmer word segmentation technology does not aid performance improvement. We aim to release our models and datasets to the community in hopes of facilitating the future development of Khmer NLP applications.

Key words: pretrained models; Khmer language; word segmentation; part-of-speech (POS) tagging; news categorization

1 Introduction

Pretrained models (PTMs) have greatly shaped the landscape of natural language processing (NLP). In general, PTMs are aimed at learning universal language representations by training models on large unannotated corpora[1] and then fine-tuning the learned representations for the tasks of interest. Extensive research[2–4] has demonstrated that the use of PTMs in a variety of downstream NLP tasks could bring remarkable improvements (and mostly achieve state-of-the-art performance) and demand a minimal amount of labeled data in supervised learning.

Although PTMs have been the default settings for most NLP applications, they generally require a large amount of computation to produce good results. As revealed by several reports[5–7], most massive network architectures are trained on unimaginably large corpora and use thousands of Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs). For high-resource languages, such as English and Chinese, the required resources may not be difficult to obtain. However, for low-resource languages, acquiring abundant unlabeled data is a major obstacle. Hence, NLP for minority language groups has yet to progress further.

In the current work, we attempt to advance the research on PTMs for the Khmer language. Utilizing publicly available Open Super-large Crawled Aggregated coRpus (OSCAR) and Wiki corpora, we train several Khmer PTMs under different settings. We then evaluate their performance on two downstream tasks: Part-of-speech tagging and news categorization. Whereas the former task adopts an open-source dataset, the latter uses a self-constructed one. The experimental results show the effectiveness of the Khmer PTMs. In addition, as Khmer is a language with no explicit delimiters between two words, we also exploit the impact of performing word segmentation on downstream tasks and find that the current Khmer segmentation technology

• Shengyi Jiang, Sihui Fu, Nankai Lin, and Yingwen Fu are with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510000, China. E-mail: neakail@outlook.com.
• Shengyi Jiang is also with Guangzhou Key Laboratory of Multilingual Intelligent Processing, Guangdong University of Foreign Studies, Guangzhou 510000, China.
* To whom correspondence should be addressed.
Manuscript received: 2021-05-15; revised: 2021-07-01; accepted: 2021-07-30

is not helpful. We plan to make all our models and datasets accessible to the community to serve as strong baselines and encourage future research on Khmer NLP.

Our paper is organized as follows: Section 2 briefly reviews the development of pretrained language models, as well as some published works related to Khmer NLP; Section 3 introduces the two PTMs employed in this work, the data source for pretraining, and the word segmentation tool; Section 4 describes two downstream tasks, as well as the datasets and evaluation metrics used; Section 5 details the experiments and analyzes the results; and Section 6 concludes our work.

2 Related Work

2.1 Pretrained language models

Generally, pretrained representations could be either context-free or contextual. Representatives of context-free models include word2vec[8] and Global Vectors for Word Representation (GloVe)[9], both of which only generate a single word embedding for each word in the vocabulary and ignore the fact that a word might have different meanings in different contexts. By contrast, contextual language models consider context information as the representation of a word depends on the other words in a sentence. Contextual models such as Embedding from Language Models (ELMo)[10] and Universal Language Model Fine-tuning (ULMFit)[11] have long used the unidirectional approach. In other words, during the pretraining phase, models are trained by predicting a word conditioned only on one side of the input sequence. Based on the architecture of a transformer[12], Bidirectional Encoder Representations from Transformers (BERT)[2] employs Masked Language Modeling (MLM) as one of its training objectives and first achieves bidirectional language understanding in the true sense. Several models, such as Robustly optimized BERT approach (RoBERTa)[3] and XLNet[5], have since been proposed to promote the success of BERT, but they usually require large networks and datasets to be effective. Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA)[13] introduces a relatively efficient pretraining approach involving the replacement of token detection, which helps it to produce a model that is comparable to or better than the best transformers with minimal computing power.

Although pretrained language models are widely used in the field of NLP, their progress on the Khmer language remains stagnant mainly because of the scarcity of training data and the difficulty of compiling evaluation tasks. To address the research gap, we present the first BERT and ELECTRA models pretrained for the Khmer language.

2.2 Natural language processing for Khmer

Despite being a low-resource language, Khmer has attracted increasing attention in recent years. The research on Khmer is growing and now covers word segmentation[14], knowledge graph construction[15], parallel/comparable corpus construction[16, 17], named entity recognition[18, 19], etc. However, the data of most existing research are not publicly accessible. Hence, we could only choose part-of-speech (POS) tagging and text classification as the downstream tasks for subsequent model evaluation because the data for these two tasks are readily available. In the following, we briefly review related studies on Khmer POS tagging and text classification.

POS tagging for Khmer. Nou and Kameyama[20] designed a 27-tag scheme for Khmer and then developed a Khmer POS corpus, which includes 1298 sentences, along with a tagger built upon the transformation-based approach; they subsequently proposed a hybrid approach that combines rule-based and tri-gram models for unknown word POS guessing[21]. The Pan Asia Networking (PAN) Localization Project[22, 23] for the Khmer language also defined a 21-POS tag set and constructed a semi-automatic tagging corpus comprising 3998 sentences; the proposed POS tagger was based on the decision tree approach. Aiming at joint tokenization and POS tagging for low-resource languages. Ding et al.[24] introduced the NOVA annotation system and applied it to Khmer; they finally presented an annotation guideline with seven POS tags and a corpus with 20 106 annotated sentences. Thu et al.[25] also developed a manually tagged corpus with their own devised tag set; on the basis of this corpus, they systematically compared the performance of six well-known POS tagging methods so as to present a robust Khmer POS tagger. In the current work, we adopt the corpus of Thu et al.[25] as our evaluation data.

Text classification for Khmer. To the best of our knowledge, few scholars have conducted research on text classification for Khmer. Khoeurn and Kim[26] suggested a Khmer music ranking website on which the data are sourced from the posts and comments found on the Facebook pages of production companies. The basic

idea was to translate Khmer texts to English first and then conduct sentiment analysis on the translated texts to acquire the ranking. This study could be regarded as an initial attempt to perform sentiment analysis for Khmer texts via machine translation. Meanwhile, Ratanak[27] focused on the sentiment classification of Khmer comments on the news, and first attempted to identify the sentiments of texts at the sentence level and then made use of such clues to determine the sentiments at the document level. However, these works did not release the data. Moreover, we cannot find any publicly available data about Khmer text classification. Hence, we build a dataset from scratch.

3 Khmer Pretrained Models

3.1 BERT

In contrast to previous works on pretraining contextual representations that adopt unidirectional language modeling, BERT[2] takes advantage of its bidirectionality, which allows it to consider the full context of a word by looking at the words that precede and follow it. Its internal structure is actually the encoder part of a transformer, which could model dependencies within a long sequence while enabling efficient parallelization with the help of the multihead self-attention mechanism. Figure 1 presents a brief illustration of the architecture of BERT.

BERT models are usually first pre-trained on the enormous amount of unlabeled text from the web, and then fine-tuned for specific tasks which possess far less data. The pre-training of BERT involves two tasks: masked language modeling and next sentence prediction. In the MLM task, some percentage of the input tokens are masked randomly, and the model needs to predict these masked tokens. As for NSP, the model is asked to learn relationships between sentences, so as to tell whether Sentence B is the actual next sentence that follows Sentence A, or is just a random sentence from the corpus. The pre-training procedure for BERT is demonstrated in Fig. 2.

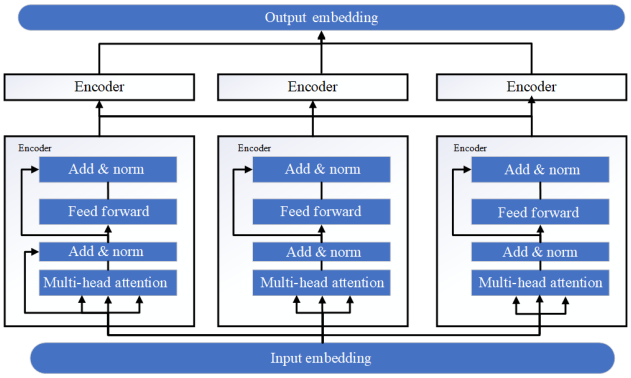

Fig. 1 Architecture of BERT.

3.2 ELECTRA

Although BERT achieves superior performance in many natural language understanding tasks, BERT models generally require a large number of parameters and extensive data to achieve high performance. In search of an alternative, Clark et al.[13] proposed an efficient PTM called ELECTRA. Unlike its predecessors that rely on MLM pretraining, ELECTRA adopts a novel approach called "replaced token detection" (RTD). Instead of masking a random selection of input tokens, this approach tries to construct a corrupted sequence by replacing some tokens in the original input with plausible alternatives sampled from a small generator (a transformer encoder). Then, a discriminator (also a transformer encoder) takes the corrupted sequence as input and identifies whether each token in it has been replaced by the generator or not. During the pretraining phase, the generator is trained jointly with the discriminator, with their combined loss minimized. As for fine-tuning, the generator is discarded, and the discriminator, i.e., the pretrained ELECTRA model, is retained. Similar to BERT, ELECTRA can then be applied to various language tasks. As RTD is defined over all input tokens rather than on masked tokens alone,

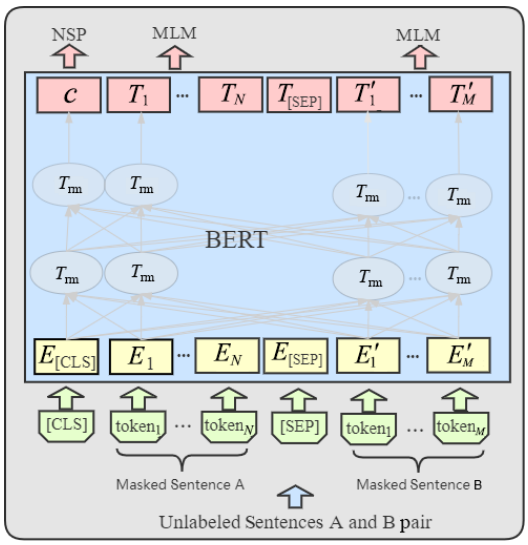

Fig. 2 Pretraining procedure for BERT. T_{rm} indicates transformer-encoder, [CLS] is the special symbol for classification output, and [SEP] is the special symbol to separate non-consecutive token sequences. E_i represents the input embedding of token i and $E_{[\mathrm{CLS}]}$ represents the input embedding of [CLS]. T_i represents the contextual representation of token i and C represents the representation of [CLS].

it is more efficient than MLM. In addition, it could help to mitigate pretraining/fine-tuning discrepancies. Figure 3 presents an overview of RTD.

3.3 Data for pretraining

To train our models, we try to collect texts from different sources. On the one hand, we use all the central Khmer data from the OSCAR corpus[‡], a large multilingual corpus whose texts come from the Common Crawl corpus[§]. Although Common Crawl comprises scraped data from the Internet and covers a wide range of topics, it distributes the data as a set of plain text files, each of which includes numerous documents that are written in different languages but lack any language information. Suárez et al.[28] proposed the goclassy architecture to perform language classification and filtering on the Common Crawl corpus and obtained the language-classified and ready-to-use OSCAR with 166 different languages available thus far.

Articles on Khmer Wikipedia[¶] are also used as part of our corpus for pretraining. We adopt the Khmer wiki data[‖] downloaded in January 2020 from the Wikipedia dumps. The data consist of 2536 documents written in Khmer. The statistics of all the data used for pretraining are shown in Table 1.

3.4 Word segmentation of Khmer language

In the writing system of the Khmer language, words within the same sentences or phrases are run together with no explicit delimiters among them. Unlike those in English or French, spaces in Khmer texts are not used as word boundary delimiters and usually serve as phrase delimiters for ease of reading. No standard rule indicates when to use or not use spaces. Chea et al.[29] pointed out that one challenge for Khmer word segmentation lies in the fact that a single sentence could be segmented in various ways with regard to its meaning in context. Hence, for Khmer NLP, one question needs to be answered: Is it necessary to conduct word segmentation on Khmer texts at the beginning? In other words, how beneficial is word segmentation for downstream tasks?

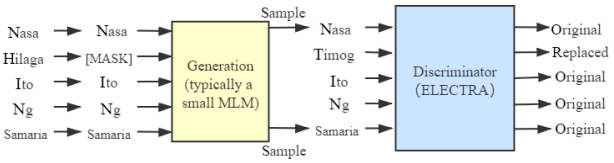

Fig. 3 Overview of replaced token detection.

Table 1 Statistics for pretraining of the corpus.

Source	Number of lines	Number of tokens	
		Unsegmented	Segmented
OSCAR	1 705 029	10 316 527	40 255 297
Wiki	333 060	2 539 906	8 725 557

To address this problem, we pretrain our models on unsegmented and segmented texts and then compare their performances on different downstream tasks. In this work, we employ the Khmer word segmenter[**] developed by Chea et al.[29]. Apart from segmenting single words, Chea et al.[29] proposed to segment three types of compound words: Those composed of two or more single words, those with specific prefixes, and those with specific suffixes. On the basis of such a segmentation scheme, Chea et al.[29] constructed a manually segmented corpus with 97 340 sentences. Then, the Khmer word segmenter was trained on this corpus by using the conditional random field model in a closed test. The precision, recall, and F-score were reported to be 0.986, 0.983, and 0.985, respectively. The statistics of our pretraining data after sentence and word segmentation are presented in Table 1.

4 Evaluation Tasks

We evaluate our PTMs on two downstream NLP tasks: POS tagging and text classification. In the following sections, we briefly introduce each task, along with the evaluation datasets and procedures.

4.1 POS tagging

POS tagging refers to the process of determining the grammatical category of a word in a text according to its definition and context. It is usually regarded as a sequence labeling problem in which each token in a given input sequence is assigned a categorical label.

The dataset we use for this task comes from Thu et al.[25], they first collected 12 000 sentences from several Khmer websites and performed initial word segmentation using the segmenter mentioned in Section 3.4. Annotators were then asked to tag each word in the sentences under the guidance of the proposed 24-tag POS scheme and to fix the segmentation errors. In addition, Thu et al.[25] collected 1000 Khmer sentences to build a test set. Some statistics about this corpus are shown in Table 2.

The performance of POS tagging is evaluated using accuracy, precision, recall, and micro-F1 score, as

[‡]https://oscar-corpus.com/
[§]https://commoncrawl.org/
[¶]https://km.wikipedia.org/
[‖]https://github.com/phylypo/khmer-language-model-ulmfit

[**]https://github.com/VietHoang1512/khmer-nltk

Table 2 Statistics for POS corpus.

Type of dataset	Number of sentences	Number of words	Number of unique words
Training set	12 000	129 029	7624
Test set	1000	10 397	2743

provided by the seqeval[††] module. Accuracy refers to the ratio of the number of POS tags that a model correctly predicts to the number of all POS tags in the corpus. For each POS tag, its precision refers to the number of tokens correctly labeled as this tag (i.e., true positives, TPs) divided by the total number of tokens predicted by the model as having this tag (i.e., the sum of TPs and false positives, FPs, which refer to the items incorrectly predicted as having this tag). The recall is defined as the number of TPs divided by the total number of tokens that actually have this tag (i.e., the sum of TPs and false negatives, FNs, which are the tokens wrongly predicted as not having this tag). For the final results, seqeval gives the tag-wise precision and recall (i.e., micro-precision and micro-recall), which can be respectively calculated as follows:

$$P_{\text{micro}} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FP}_i} \tag{1}$$

$$R_{\text{micro}} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} \text{TP}_i + \sum_{i=1}^{n} \text{FN}_i} \tag{2}$$

where n refers to the number of POS tags in the corpus. As the harmonic mean of precision and recall, micro-F1 can be obtained by

$$F_{\text{micro}} = \frac{2P_{\text{micro}}R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}} \tag{3}$$

4.2 Text classification

Text classification is the task of assigning sentences or documents to predefined categories. In this work, we handle the problem of news categorization.

Given the absence of a publicly available dataset, we scrape some news articles written during 2010 to 2021 from VOA Khmer[※] to build our evaluation dataset. As the articles are sorted into different categories, we do not need to manually annotate the category for each news article; we simply adopt its classification scheme.

The whole dataset comprises 7166 Khmer news articles; each labeled as one of the following eight categories: Culture, economic, education, environment, health, politics, rights, and science. The dataset is divided into the training, validation, and tests with a ratio of 0.6:0.2:0.2. We should point out that, as different categories have significantly different numbers of articles, the division is conducted at the category level rather than on the whole dataset so as to preserve the percentage of samples for each category. The detailed statistics of our dataset for Khmer news categorization are presented in Table 3.

The classification performance is evaluated using macro F1-score and accuracy. With regard to each category, the F1-score is calculated as follows:

$$F_i = \frac{2P_i R_i}{P_i + R_i} \tag{4}$$

where P_i and R_i are obtained by

$$P_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FR}_i} \tag{5}$$

$$R_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{6}$$

Table 3 Statistics of our dataset for Khmer news categorization.

Category	Number of articles	Number of tokens (segmented)	Number of articles in the training set	Number of articles in the validation set	Number of articles in the test set
Education	568	80 646	340	114	114
Politics	1205	127 884	723	241	241
Economic	1150	137 157	690	230	230
Rights	1149	145 154	689	230	230
Health	1201	103 238	720	240	241
Environment	965	111 571	579	193	193
Science	266	30 139	159	53	54
Culture	662	93 023	396	133	133
Total	7166	828 812	4296	1434	1436

[††]https://github.com/SunYanCN/seqeval

[※]https://khmer.voanews.com

Thus, the macro F1-score is computed as follows:

$$F_{\text{macro}} = \frac{\sum_{i=1}^{n_c} F_i}{n_c} \tag{7}$$

where n_c is the number of categories. As for the accuracy, it is simply the ratio between the number of those correctly classified articles and the total number of articles.

5 Experiment

In this part, we present the experimental setup and results for the Khmer PTMs. In particular, we pretrain the models on unsegmented and segmented texts and then apply them to the two tasks to explore the benefits of word segmentation.

5.1 Pretraining

All the models are trained on the pretrained data for ten epochs. Their learning rates gradually increase over the first 5000 steps to a peak value of 1×10^{-4}, after which they decline linearly. The weights are initialized randomly from a normal distribution with a mean of 0.0 and a standard deviation of 0.02. We build and train the BERT and ELECTRA tokenizers from scratch on our pretrained data, each of which has a vocabulary size of 32 000.

For the BERT models, the size of a minibatch is 128, with the maximum sequence length being 512. The BERT SMALL model has four layers with eight attention heads. The dimensions of the encoder and feedforward layers are 512 and 2048, respectively. The BERT BASE model has 12 layers with 12 attention heads, and the dimensions of the encoder and feedforward layers are 768 and 3072, respectively.

With regard to the ELECTRA models, the hyperparameters used in ELECTRA SMALL and ELECTRA BASE are the same as those in BERT SMALL and BERT BASE, except that the MLM probability for the ELECTRA models is 0.25 while that for the BERT models is 0.15.

Figure 4 illustrates the pretraining curves for each model. Given the same training time, the deep and wide models are more helpful in achieving a low training loss than the shallow models. Pretraining on segmented texts also aids the decrease of training loss. After 150 000 training steps, BERT BASE and ELECTRA BASE trained on segmented texts reach the lowest training loss.

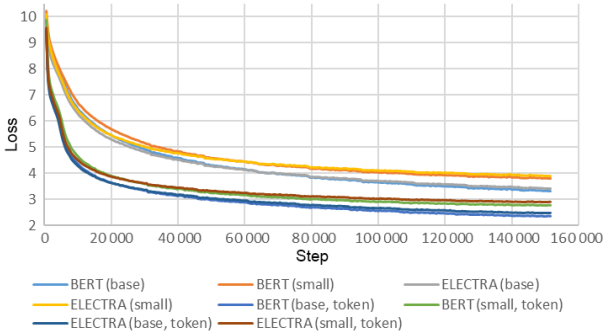

Fig. 4 Pretraining curves for all models, with training loss over the steps.

5.2 Fine-tuning

Fine-tuning is conducted by attaching relevant classifiers required for each task. After performing hyperparameter optimization, we employ the optimal setup for all evaluation experiments. While all other settings employed in both SMALL and BASE models are the same, the only difference is the peak value of the learning rate. The SMALL models take a peak value of 1×10^{-4}, but for BASE this value is smaller, which is 5×10^{-5}. From the perspective of evaluation tasks, the POS tagging models go through 10 epochs and use sequences of 200 tokens in a batch size of 32. In terms of the classification task, the models are fine-tuned on sequences of 128 tokens in the batch size of 32, for up to 3 epochs. We report the results of the models that perform best on the validation sets.

5.3 Results of POS tagging

In fine-tuning the models for POS tagging, we attach a linear classifier to the top layers of the PTMs to predict the POS tag for each token in the input sequence. Cross entropy is then used as the loss function. Table 4 shows the final results of our models on this task.

Apart from developing the Khmer POS corpus, Thu et al.[25] evaluated several POS tagging approaches; their released codes are run herein to reproduce the results reported in their paper as our baseline and thereby effectively assess the performance of our models. In the codes of Thu et al.[25], they provided the implementation of four approaches, namely, support vector machine (SVM), hidden Markov model (HMM), maximum entropy (MAX-Ent), and ripple-down rules (RDRs); for the former three methods, the features they adopted include the labels of the current word and its neighboring words. The paper of Thu et al.[25] only reported the accuracy of these methods, whereas the current work

Table 4 Results of POS tagging task on the test set. (%)

Model	Accuracy	Precision	Recall	F1 score
SVM	81.68	69.45	76.40	72.76
HMM	95.26	93.50	93.33	93.42
MAX-Ent	92.97	90.32	89.12	89.91
RDR	95.82	94.27	93.83	94.05
BERT (small)	96.10	93.40	93.14	93.27
BERT (base)	96.79	94.37	94.20	94.29
ELECTRA (small)	96.20	93.27	93.27	93.27
ELECTRA (base)	96.73	94.39	94.34	94.36
BERT (small, segmented)	96.83	94.42	94.23	94.33
BERT (base, segmented)	**97.03**	**94.77**	**94.94**	**94.86**
ELECTRA (small, segmented)	96.66	94.49	94.27	94.38
ELECTRA (base, segmented)	97.00	94.75	**94.94**	94.85

presents the micro-precisions, recalls, and F1 scores for a comprehensive comparison⋆.

In general, the PTMs outperform all the baseline methods. The best model, BERT BASE, shows an improved accuracy of 1.21 points and recall of 1.11 points relative to the best baseline, RDR. The base models obviously perform better than the small variants, but the improvements may not be significant for this task, especially with respect to accuracy. As for model selection, the gap between BERT and ELECTRA is quite small.

We generally presume POS tagging to be a token-based task, but the results show that segmentation does not greatly improve performance. Even the models pretrained on unsegmented texts are able to beat the baseline approaches. As the PTMs read an entire sequence at once and map it into intermediate representations, the result suggests that POS tagging could benefit from the information provided by the whole sequence. As segmentation is performed by the segmenter without any manual intervention, the negative impact of segmentation errors on the results deserves further investigation.

5.4 Results of text classification

As for the fine-tuning of the models for text classification, a linear classifier is also added to the PTMs to predict the category to which each input article belongs. The loss function is still cross entropy. Table 5 shows the overall results for each model on the test set. Note that the results concerning segmentation are all reported on the basis of the models pretrained and fine-tuned on the segmented training data; the others are based on the unsegmented data.

The results are consistent with those of POS tagging. In most cases, the base models beat the small variants, and the ELECTRA-based models perform slightly better than the BERT-based ones. Segmentation indeed enhances model performance, but its influence is still unremarkable. Hence, the current word segmentation for Khmer does not appear to be highly beneficial for downstream tasks.

As the F1 scores and accuracy are not satisfactory, we try to conduct an error analysis by checking the performance of the models with the help of a confusion matrix. For the four BASE models, we consider macro F1 scores on each category (Table 6) and suppose that the models may suffer from the class imbalance problem as their performance on the categories with the least number of articles (e.g., “Science” and “Education”) is relatively poor. To deal with the imbalanced data, we

Table 5 Results of the news categorization task on the test set. (%)

Model	Macro F1 score	ACC
BERT (small)	65.97	67.41
BERT (base)	66.99	68.66
ELECTRA (small)	66.89	68.18
ELECTRA (base)	68.22	69.29
BERT (small, segmented)	68.50	69.78
BERT (base, segmented)	67.46	69.08
ELECTRA (small, segmented)	67.97	68.94
ELECTRA (base, segmented)	**68.64**	**69.99**

Table 6 Macro F1 scores on each category for four PTMs. (%)

Model	Culture	Economic	Education	Environment	Health	Politics	Rights	Science
BERT (base)	70.21	67.24	**66.67**	80.68	76.17	62.64	**65.80**	57.43
ELECTRA (base)	68.59	67.49	66.00	79.51	76.52	61.64	63.95	**59.62**
BERT (base, segmented)	69.75	**69.05**	63.21	79.43	**78.37**	62.78	62.47	56.00
ELECTRA (base, segmented)	**70.97**	67.91	66.03	**81.17**	76.99	**62.95**	63.39	59.05

⋆ Although the accuracies of the other three models are close to the reported values, our result for SVM is quite at odds with the result of Thu et al.[25] (94.57%).

employ a simple yet effective informed undersampling method called EasyEnsemble[30], which samples several subsets from majority classes, trains a learner on each subset, and combines all weak learners into a final ensemble. In our experiments, we generate seven subsets, with each one satisfying an equal class distribution‡‡. Seven learners are trained, and their outputs are combined to obtain the ensemble results (Table 7). The results demonstrate that the ensemble method does help improve model performance as all models achieve high F1 scores and high accuracy. When we regard the macro F1 scores for each category for the four PTMs after utilizing EasyEnsemble (Table 8), we find that although the performance on "Science" is slightly low, our ensemble learners are still able to alleviate the imbalance issue to some extent.

We further conduct a case study and consider the top five mistakes on the test set. As Table 9 reveals, the major categories (e.g., "Politics" and "Economics") account for the most mistakes. Hence, the introduction of the undersampling method fails to lead to significant improvements. All models struggle to distinguish the major categories, especially "Politics" and "Rights". Such a result could shed some light on their relatively low scores for these two categories and on the fact that, in practice, the new articles under these categories tend to overlap one another and are more closely related than we think.

Table 7 Results of news classification task on the test set after utilizing EasyEnsemble.

(%)

Model	Macro F1 score	ACC
BERT (small)	67.05	68.38
BERT (base)	68.21	69.71
ELECTRA (small)	67.54	68.73
ELECTRA (base)	69.31	70.47
BERT (small, segmented)	68.64	69.78
BERT (base, segmented)	68.86	70.19
ELECTRA (small, segmented)	68.96	70.40
ELECTRA (base, segmented)	**69.42**	**70.61**

6 Conclusion

In this work, we present PTMs for the Khmer language for the first time by using BERT and ELECTRA. Considering the challenges presented by limited resources and the difficulty of compiling labeled data, we only apply the models to two downstream tasks, the dataset for one of which is self-constructed. The experimental results demonstrate the effectiveness of our Khmer PTMs. We also explore whether performing word segmentation exerts a positive influence on downstream tasks. Although the current Khmer word segmentation technology could offer some benefits, the improvements gained are not significant. By releasing our models and datasets to the community, we hope to advance the Khmer NLP research. For our future work, we will explore whether a more effective segmenter can

Table 8 Macro F1 scores on each category for four PTMs after utilizing EasyEnsemble.

(%)

Model	Culture	Economic	Education	Environment	Health	Politics	Rights	Science
BERT (base)	72.85	67.83	65.79	78.91	77.31	**63.86**	65.19	53.91
ELECTRA (base)	72.30	**70.00**	**68.47**	79.90	77.18	63.23	**65.78**	57.63
BERT (base, segmented)	**73.68**	69.64	66.97	80.10	**78.03**	62.24	64.18	56.00
ELECTRA (base, segmented)	73.40	69.74	67.28	**81.64**	77.85	62.18	64.75	**58.54**

Table 9 Top five mistakes on the test set for the four PTMs.

Model	Reference	Hypothesis	Frequency	Model	Reference	Hypothesis	Frequency
BERT (base)	Politics	Rights	48	BERT (base, segmented)	Politics	Rights	50
	Politics	Economics	28		Rights	Politics	38
	Rights	Politics	27		Politics	Economics	26
	Health	Economics	23		Economic	Environment	20
	Health	Politics	19		Rights	Economic	19
ELECTRA (base)	Politics	Rights	44	ELECTRA (base, segmented)	Politics	Rights	49
	Rights	Politics	31		Rights	Politics	29
	Politics	Economics	27		Politics	Economics	25
	Health	Economics	21		Rights	Economics	21
	Rights	Economics	19		Politics	Health	21

‡‡ For each subset, the sample ratios are 1.0 for the "Science" category, 0.3 for the "Culture" and "Education" categories, and 0.5 for all the others.

lead to even higher performance. We will also attempt to develop other Khmer NLP tasks, such as named entity recognition, natural language inference, and question answering.

Acknowledgment

This work was supported by the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2017KZDXM031) and Guangzhou Science and Technology Plan Project (No. 202009010021). We would like to extend our sincere gratitude to the anonymous reviewers for their insightful feedbacks.

References

[1] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, Pre-trained models for natural language processing: A survey, *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[3] Y. H. Liu, M. Ott, N. Goyal, J. F. Du, M. Joshi, D. Q. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv: 1907.11692, 2019.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, *OpenAI Blog*, vol. 1, no. 8, pp. 9–32, 2019.

[5] Z. L. Yang, Z. H. Dai, Y. M. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, XLNet: Generalized autoregressive pretraining for language understanding, presented at the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, Canada, 2019, pp. 5754–5764.

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv: 2005.14165, 2020.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, in *Proc. 1st Int. Conf. Learning Representations*, *ICLR* 2013, Scottsdale, AZ, USA, 2013, pp. 1–9.

[9] J. Pennington, R. Socher, and C. Manning, GloVe: Global vectors for word representation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1532–1543.

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, Deep contextualized word representations, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, New Orleans, LA, USA, 2018, pp. 2227–2237.

[11] J. Howard and S. Ruder, Universal language model fine-tuning for text classification, in *Proc. 56th Annu. Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018, pp. 328–339.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 1–15.

[13] K. Clark, M. T. Luong, Q. V. Le, and C. D. Manning, ELECTRA: Pre-training text encoders as discriminators rather than generators, in *Proc. 8th Int. Conf. Learning Representations*, *ICLR 2020*, Addis Ababa, Ethiopia, 2020, pp. 1–18.

[14] B. B. Yu, Y. Nuo, X. Yan, Q. L. Lei, G. Y. Xu, and F. Zhou, Segmentation and alignment of Chinese and Khmer bilingual names based on hierarchical dirichlet process, presented at Int. Conf. Mechatronics and Intelligent Robotics (ICMIR2018), Kunming, China, 2018, pp. 441–450.

[15] U. Phon and C. Pluempitiwiriyawej, Khmer WordNet construction, presented at the 5th Int. Conf. Information Technology (InCIT), Chonburi, Thailand, 2020, pp. 122–127.

[16] H. Y. Chi, X. Yan, S. Y. Li, F. Zhou, G. Y. Xu, and L. Zhang, The acquisition of Khmer-Chinese parallel sentence pairs from comparable corpus based on manhattan-BiGRU model, presented at the 2020 Chinese Control and Decision Conf., Hefei, China, 2020, pp. 4801–4805.

[17] S. Ning, X. Yan, Y. Nuo, F. Zhou, Q. Xie, and J. P. Zhang, Chinese-Khmer parallel fragments extraction from comparable corpus based on dirichlet process, *Procedia Comput. Sci.*, vol. 166, pp. 213–221, 2020.

[18] H. S. Pan, X. Yan, Z. T. Yu, and J. Y. Guo, A Khmer named entity recognition method by fusing language characteristics, presented at the 26th Chinese Control and Decision Conf., Changsha, China, 2014, pp. 4003–4007.

[19] X. H. Liu, X. Yan, G. Y. Xu, Z. T. Yu, and G. S. Qin, Khmer-Chinese bilingual LDA topic model based on dictionary, *Int. J. Comput. Sci. Math.*, vol. 10, no. 6, pp. 557–565, 2019.

[20] C. Nou and W. Kameyama, Khmer POS Tagger: A transformation-based approach with hybrid unknown word handling, presented at the Int. Conf. Semantic Computing (ICSC 2007), Irvine, CA, USA, 2007, pp. 482–492.

[21] C. Nou and W. Kameyama, Hybrid approach for Khmer unknown word POS guessing, presented at the 2007 IEEE Int. Conf. Information Reuse and Integration, Las Vegas, NV, USA, 2007, pp. 215–220.

[22] PAN Localization Cambodia (PLC) of IDRC, Part of speech template, https://www.dit.gov.bt/sites/default/files/PartOfSpeech.pdf, 2007.

[23] PAN Localization Cambodia (PLC) of IDRC,

Khmer automatic Pos tagging, https://moam.info/research-report-on-khmer-automatic-pos-pan-localization_5a22d8711723ddefdcf2139f.html, 2008.

[24] C. C. Ding, M. Utiyama, and E. Sumita, NOVA: A feasible and flexible annotation system for joint tokenization and part-of-speech tagging, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 18, no. 2, p. 17, 2019.

[25] Y. K. Thu, V. Chea, and Y. Sagisaka, Comparison of six POS tagging methods on 12K sentences Khmer language POS tagged corpus, in *Proc. 1st Regional Conf. Optical Character Recognition and Natural Language Processing Technologies for ASEAN Languages* (*ONA 2017*), Phnom Penh, Cambodia, 2017, pp. 1–12.

[26] S. Khoeurn and Y. S. Kim, Sentiment analysis engine for Cambodian music industry re-building, *J. Korea Soc. Simul.*, vol. 26, no. 4, pp. 23–34, 2017.

[27] T. Ratanak, A study on the sentiment classification for Khmer comments on news, (in Chinese), Master dissertation, Kunming Univ. Sci. Technol., Kunming, China, 2017.

[28] P. J. O. Suárez, B. Sagot, and L. Romary, Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures, in *Proc. 22nd Workshop on Challenges in the Management of Large Corpora* (*CMLC-7*), Mannheim, Germany, 2019, pp. 9–16.

[29] V. Chea, Y. K. Thu, C. C. Ding, M. Utiyama, A. Finch, and E. Sumita, Khmer word segmentation using conditional random fields, in *Khmer Natural Language Processing*, Phnom Penh, Cambodia, 2015, pp. 62–69.

[30] X. Y. Liu, J. X. Wu, and Z. H. Zhou, Exploratory undersampling for class-imbalance learning, *IEEE Trans. Syst., Man, Cybern., Part B* (*Cybern.*), vol. 39, no. 2, pp. 539–550, 2009.

Shengyi Jiang received the PhD degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2005. He is a professor at Guangdong University of Foreign Studies, Guangzhou, China. He is mainly engaged in data mining and natural language processing.

Sihui Fu received the MS degree in management science and engineering from Guangdong University of Foreign Studies, Guangzhou, China, in 2019. She has published papers in the International Conference on Language Resources and Evaluation. She is mainly engaged in data mining and natural language processing.

Nankai Lin received the BS degree in software engineering from Guangdong University of Foreign Studies, Guangzhou, China, in 2019. Now he is pursuing the master degree in cyberspace security at Guangdong University of Foreign Studies, Guangzhou, China. He is mainly engaged in data mining and natural language processing.

Yingwen Fu received the BS degree in software engineering from Guangdong University of Foreign Studies, Guangzhou, China, in 2020. She is now pursuing the MS degree at Guangdong University of Foreign Studies, Guangzhou, China. She has published papers in *Journal of Intelligent and Fuzzy Systems*. She is mainly engaged in data mining and natural language processing.