BERT Fine-Tuning for Amharic Sentiment Classification

Girma Neshir
IT doctoral program
Addis Ababa University
Ethiopia
girma.neshir@aau.edu.et

Solomon Atnafu
Department of Computer Science
Addis Ababa University
Ethiopia
satnafu@gmail.com

Andreas Rauber
TU Wien Informatics
Austria
rauber@ifs.tuwien.ac.at

Abstract

Transfer learning is getting great attention in advancing researches in the downstream tasks of Natural Language Processing(NLP) in a cost-effective and rapid way. This is because of the rapid development of context-based pre-trained language models. In this research, we develop the first Bidirectional Encoder Representations from Transformers(BERT) for the Amharic language, and then, this pre-trained BERT model is further fine-tuned for downstream tasks of Amharic sentiment classification. The fine-tuned BERT for Amharic sentiment classification outperformed with an accuracy of 95% with the condition of the insufficient labeled corpus. Other evaluation results are also achieving the best in performing sentiment classification of Amharic Facebook comments.

1 Introduction

BERT pre-trained models are used for our proposed work to perform sentiment classification. We are inspired by the works of (Devlin et al., 2018; McCormick and Ryan, 2019; McCormick, 2019; Choudhury) in that BERT pre-trained models are fine-tuned for downstream NLP tasks with insufficient labeled training data, faster computational speed, quicker development, and better results. In line with the framework of (Devlin et al., 2018), our proposed BERT architecture has 3 main layers. These include:The input and BERT encoder layers are trained with a large corpus and then the model is fine-tuned for different tasks by customizing the output layer. This minimizes the size of labeled data developing classifiers in the target task without requiring huge computational resources in shorter development times. As it does not include Amharic in the Huggingface pre-trained model library, we need to develop bert tokenizer and develop a pre-trained language model from scratch for later fine-tuning to carryout Amharic text classification tasks(i.e. Sentiment Analysis). In the next subsection, we present the steps to develop the BERT pre-trained model for Amharic.

2 Method

2.1 BERT Tokenizer of Amharic texts

Tokenizers encode inputs to the models. BERT pre-trained NLP models comprise their own tokenizers' library(huggingface, Accessed in 2020). However, the Amharic language is not included in any of BERT pre-trained NLP models. As a result, Amharic BERT word pieces tokenizer is also not present in any of the existing BERT tokenizer vocab.

For sentence sequence encoding, we need to determine the maximum length relying on the frequency distribution of lengths of training samples. We fix the maximum length of the input sequence to 512. For truncating, we set from the beginning as Amharic verbs are usually found at the end of a sentence. Padding set from the beginning of training samples avoids information loss.

2.2 BERT Model Building and Fine-Tuning

Nowadays, there are various context-based text representations including BERT, ELMO, ULMFiT, etc. For the Amharic Language model, we proposed BERT Architecture. The current model is available for only dominant languages. As mentioned earlier, it does not cover Amharic in the BERT pre-trained model library. We build an Amharic tokenizer and then build a small-scale pre-trained model for the Amharic language for further tasks to be carried out by fine-tuning it for sentiment classification.

Contributions: The contributions of the research are: (i) the development of the first BERT tokenizer and BERT Model for Amharic and (ii) the fine-tuning of the pre-trained BERT model for Amharic sentiment classification, (iii) the collection of different benchmark datasets for Amharic sentiment classification, (iv) The result is encouraging and starting to enhance the language model to test other tasks of the Amharic language, and (v) We got a notable performance improvement on sentiment classification of Amharic language where there is a lack of labeled corpus as compared with performing baseline models(SVM).

2.3 Experiment, Results and Discussion

Two experiments have been carried out. The detailed discussions of these are presented as follows.

Experiment I Building small-scale BERT pre-trained model: We developed a BERT model based on one of the popular pre-trained transformers hugging face library models. In this case, we adapted the BERT architecture configuration template using "BertForMaskedLM" as we suggest it for developing pre-trained language models for the new language. We train a "small-scale" Amharic language model with (84 million parameters = 6 layers, 768 hidden size, 12 attention heads) which is the same number of layers and attention heads as BERT(small). We then aim to fine-tune one of the downstream tasks of Amharic(i.e. sentiment classification under insufficient annotated sentiment corpus). For the pre-training experiment, we used google colab TPU of 25GB RAM and 99GB storage which is allocated for my account. We used 904MB of general domain text collections. The final pre-trained model will be publicly available for doing other downstream tasks of Amharic. We divided the data into training and validation data with 80:20 ratio. Finally, the pre-trained model is stored for further use.

Experiment II Fine-tuning LM model: For this experiment, PMO Facebook Labeled Amharic Comments(6652 samples) are used for fine-tuning Amharic sentiment classification. In this experiment, we used the BertForSequenceClassification model for the purpose of performing the Amharic sentiment classification. Results on the fine-tuning of Amharic Sentiment classification are discussed as follows. We achieved Matthews's Correlation coefficient (MCC) of 0.892 and accuracy of 0.95(1318 support) and the F-score of negative(0) and positive(1) classes of the BERT fine-tuned Amharic sentiment classifier is 97% and 93%, respectively. To test our BERT pre-trained transfer learning, we also used Matthews's correlation coefficient (MCC) as one of the model quality measurement metrics. MCC is used to evaluate the quality of models trained for binary and multiclass classification as we use it as balanced- measure models' performance trained on samples with distinct class sizes (imbalanced classes)(BogoToBogo, 2015; McCormick and Ryan, 2019). MCC values are ranging from -1(worst prediction) to +1(perfect prediction). According to the confusion matrix, only 32 of the negative test samples are wrongly classified as a positive category whereas 30 of the positive samples are incorrectly classified as a negative category. This is a dramatic improvement in performance Amharic sentiment classification when we compare it to performing the baseline result (i.e. SVM ACC.89%).We also inspected to trace the errors in predicting the sentiment of unseen texts. However, all the unseen Amharic text samples shown in Table 1 are correctly predicted.

texts(in Am-En)	original	predicted.
"በጣም ጥሩ ነው እንኩዋን ድሰ አላቹ በርቱ" /Congratulations, keep on, it is very good job./	1	1
"እባብን መቸም አምነህ በጉያህ አትይዘዉም አይታመንምና" /how you trust the snake in your pocket/	0	0
"ፈጣሪ ከናንተ ጋር ነዉና በርቱልንን-ንንን አንበሳው" /God bless you, keep on, you are our king./	1	1

Table 1: Model prediction of unseen texts

References

BogoToBogo. 2015. scikit-learn user guide release 0.16.1. http://github.com/scikit-learn.

Aniruddha Choudhury. Part 2: Bert fine-tuning tutorial with pytorch for text classification on the corpus of linguistic acceptability (cola) dataset. https://medium.com/@aniruddha.choudhury94/.

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

huggingface. Accessed in 2020. transformers. https://huggingface.co/transformers/.

Chris McCormick. 2019. Bert word embeddings tutorial. https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/.

Chris McCormick and Nick Ryan. 2019. Bert fine-tuning tutorial with pytorch. https://mccormickml.com/2019/07/22/BERT-fine-tuning/.