Tweet Sentiment Analysis :

A Comparison of LSTM and BERT

Shih-Yu Lai[1], Shiuan-Chin Huang[2], Danhui Wu[3]

7183984563 9815633745 8038800437

University of Southern California

(**Contribution:** [1] Build the BERT model [2] Build the LSTM model [3] Write the paper)

Abstract: Sentiment analysis on social media data such as tweets has been very important and challenging task. The Long Short-Term Memory (LSTM) which improves original recurrent neural network (RNN) has been widely used for this task because it tackle the problem of vanishing and exploding gradients. Also, fine-tuning of pretrained Bidirectional Encoder Representations from Transformers (BERT) performs excellent on this task and achieves state-of-the-art performances. In this paper, we build a LSTM based model and a fine-tuned BERT model to predict the overall sentiment (positive, negative or neural) in a tweet dataset and compare the performance of these 2 models. The result shows the BERT model outperforms the LSTM model at the prediction accuracy.

Keyword: BERT, LSTM, Tweet Sentiment Analysis, NLP, Single-label multiclass classification

Introduction

Social networks have huge impact on the daily life of hundreds of millions of users[1] because they accelerate people to share opinions about social events, political movements, company strategies and product preferences[2]. However, most of these information are not “machine processable” because they are unstructured. As such, it garners increasing interests from scientific community how to capture and make use of public opinions provides by social networks[2]. The resulting emerging fields are opinion mining and sentiment analysis[2].Sentiment Analysis (SA) refers to the use of Natural Language Processing (NLP) to systematically identify, extract, quantify, and study affective states and subjective information[3]. Single-label multiclass (SLMC) classification is one kind of sentiment analysis tasks, which is the task of detecting, given an opinion-laden textual item (e.g.. product review, tweet, etc.), whether it expresses a positive, negative or neural opinion about a given entity[4].

Recently, neural networks approaches have achieved state-of-the-art performance in a variety of sentiment analysis tasks. In 1997, an alteration of RNN with Long Short-Term Memory units, or LSTM units [5], was proposed by the Sepp Hochreiter and Juergen Schmidhuber. Some errors back- propagate through time in general RNN. These LSTM units help to bypass these errors. In LSTM, however, information can only flows backward to forward, which limit LSTM-based model’s ability of understanding context. Transformer based language models such as Bidirectional Encoder Representations from Transformers (BERT) overcomes the this limit by performing the learning phase by scanning the span of text in both directions, from left to right and from right to left[6]. Moreover, BERT uses a "masked language model": during the training, random terms are masked in order to be predicted by the net[6]. These differences allow BERT to be the current state of the art language understanding model[7].

In this paper, we build a LSTM based neural network classifier and a fine-tuned BERT based classifier to predict the overall sentiment (positive, negative or neural) in a twee dataset and compare the performance of these 2 models.

Data

We use datasets from the Kaggle competition "Tweet Sentiment Extraction"[8]. In this paper, our goal is predicting the sentiment labels for tweets instead of extracting support phrases for sentiment labels as in the competition, so we delete the column of "selected_text" in the training set.

Source and Format

Files: train.csv (the training set) test.csv (the test set)
Columns: textID (unique ID for each piece of text); text (the text of the tweet); sentiment (the general

sentiment of the tweet)

Text Format: <id>,"<word or phrase that supports the sentiment>"

textID	text	sentiment
cb774db0d1	I`d have responded, if I were going	neutral
549e992a42	Sooo SAD I will miss you here in San Diego!!!	negative
088c60f138	my boss is bullying me...	negative
9642c003ef	what interview! leave me alone	negative
6e0c6d75b1	2am feedings for the baby are fun when he is all smiles and coos	positive

Table 1: Subset of training set of Kaggle competition" Tweet Sentiment Extraction".

Dataset	Positive	Negative	Neutral	Total
Training set	8582	7781	11118	27471
Test set	1103	1001	1430	3534

Table 2: Statistics of training set and test set of Kaggle competition" Tweet Sentiment Extraction".

Data preprocessing

Before we start model training, we preprocess the tweets (text column) in training set. The processing steps for LSTM model are as follows:

a. Remove all rows containing null values.
b. Remove all punctuations and special symbols except those that are meaningful for sentiment extraction (For example: **** in the sentence of "Son of ****" helps us predict the sentiment as negative.). As a result, we can obtain more precise training result.
c. Convert all characters to lower case so the same letter with different cases will be seen as the same.
d. Tokenize every tweet into separate word using the tokenizer in the deep learning library Keras.
e. Convert distinct words and labels into numeric values (for example: ‘son’, ‘you’, ‘whether’ are converted to number 1, 2,3). Note: we combine the distinct tokenized words in training set and test set before converting them to numbers in order to make sure these 2 data sets have same dimensions of tokenized text.

For the BERT model, the processing steps are as follows:

a. Tokenize the input sentences with the BERT basic tokenizer to perform punctuation splitting, lower casing and invalid characters removal.
b. The input representation for each tweet is the sum of these token, segment, and position embeddings as the Figure 1 shows.

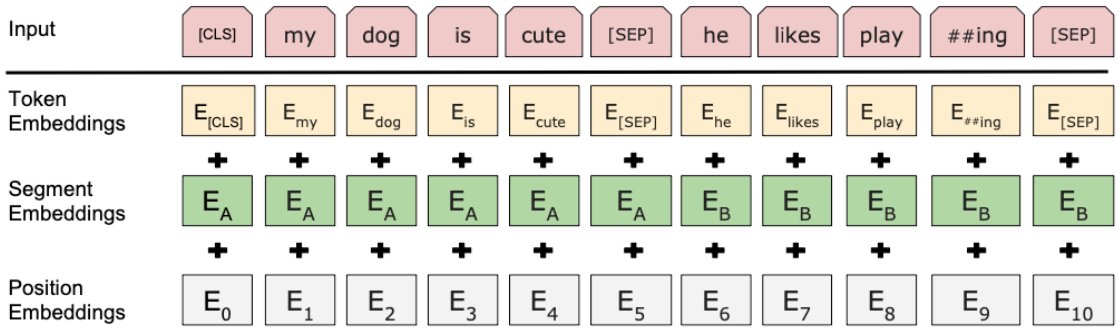

Figure 1: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings[6].

Methodology

Model Details

1. LSTM

Recurrent neural network (RNN) is well-suited for sentiment analysis. However, RNN are limited because of the problem of vanishing and exploding gradients. The Long Short-Term Memory (LSTM) can solve this problem and model the long-range dependency. Similar as RNN, LSTM also has a recurrent layer of many memory blocks, but LSTM have more controlling gates, which are the input gate, output gate and forget gate. These gates allow for a better control over the gradient flow and enable better preservation of long-range dependency.

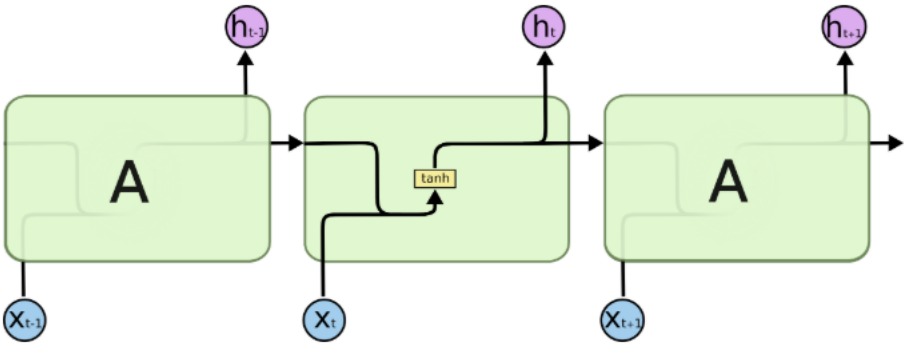

Figure 2: The repeating module in a standard RNN contains a single layer

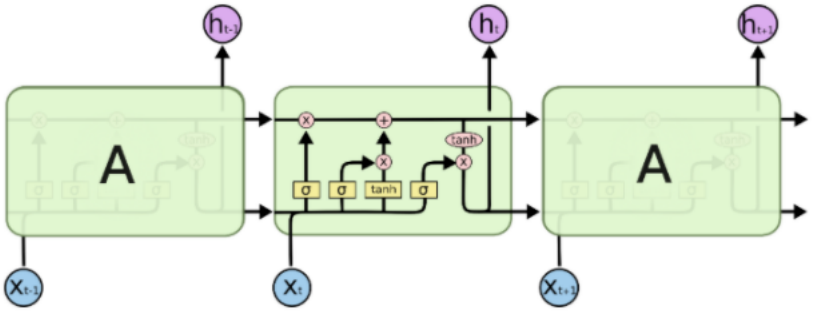

Figure 3: The repeating module in an LSTM contains four interactive layers

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a new language representation model created by Google AI Language team. BERT is designed to pretrain deep bidirectional representations from unlabeled text on both left and right context in all layers[6]. BERT is simple and powerful because by adding one additional output layer to the pre-trained BERT model we can create fine-tuned models which can be applied to various downstream NLP tasks, such as question answering and language inference[6].

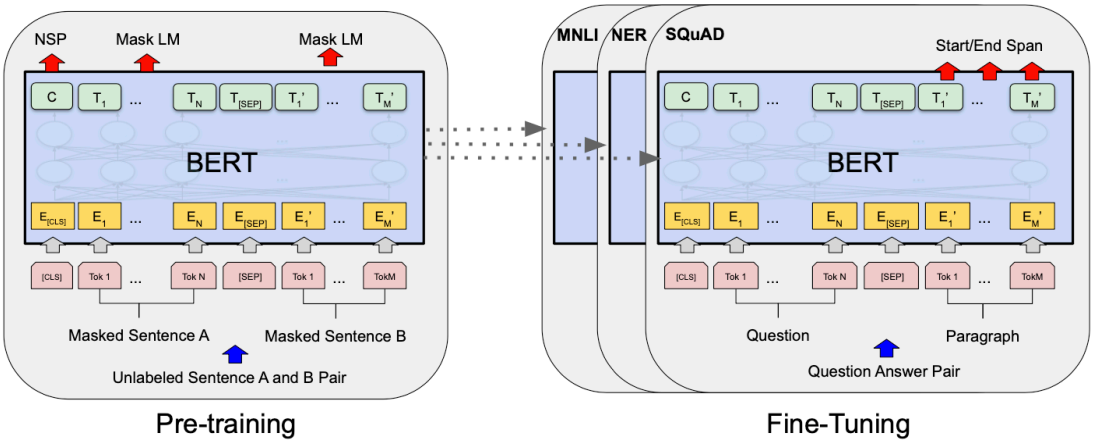

Figure 4: Overall pre-training and fine-tuning procedures for BERT[6].

The BERT model overcomes the limitation of current unidirectional language models by using a "masked language model" (MLM) pre-training objective. The masked language model randomly masks some tokens from the input in order to predict the original vocabulary id of the masked word only based on its context[6]. Outperforming than unidirectional left-to-right language model pre-training, the MLM allows us to pre-train a deep bidirectional Transformer because it enables the representation to fuse the left and the right context. Based on these advantages, we guess BERT will have better performance than LSTM model on the sentiment analysis task.

Training

1. LSTM

After reading and preprocessing training data in the method mentioned above. Then we split the training data into 2 parts --- one for training and one for validation comprising 80% and 20% of the dataset respectively. The validation set is used to tune the hyper-parameters. After data splitting, we import keras.models and choose the many to many model which is one kind of functional model. We select the functional model instead of the sequential model though the sequential model is broadly used in many classification problems. However, a Functional API has more flexibility in designing models. It is more complex, but it allows your model to share layers or design directed acyclic network graphs. Compared to sequential model, which is a linear stack of layers, we can define the relations between each layer for Functional API by ourselves. Then we start to train the model. We define 3 layers including a embedding layer, a LSTM layer and a Dense layer. The structure is indicated in the Figure 4. Then we applied the trained model to test data.

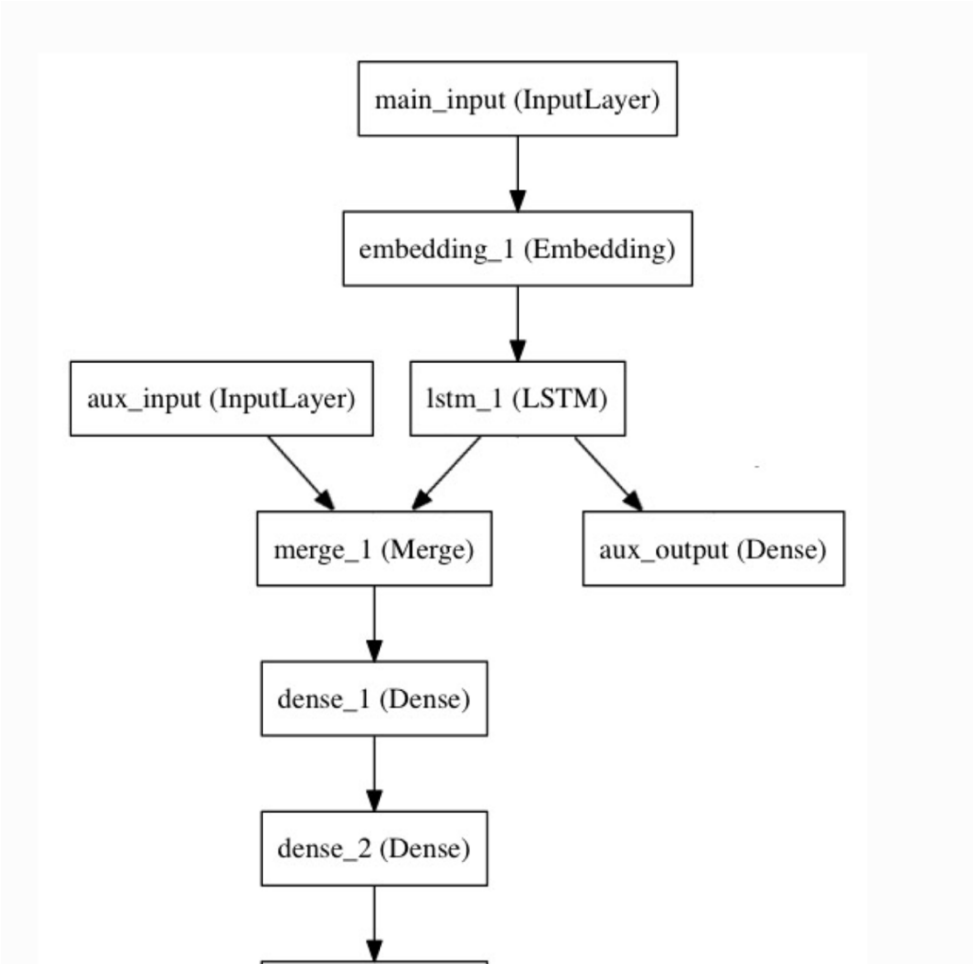

Figure 4: The layer structure of many to many model

2. BERT

There are different kinds of BERT models which can be classified into 2 categories --- the BERT $_{large}$ models and BERT $_{base}$ models. The BERT $_{base}$ model has 12 Transformer blocks, 12 self-attention heads, and 768 hidden dimension with a total parameters of 110M[6].We import the BERT $_{base\text{-}uncased}$ (one kind of BERT $_{base}$ models) as the underlying pre-trained BERT model using Pytorch. Then we read and preprocess training data in the method mentioned above. After converting all tweets into required format, we convert them back to original text to make sure the conversion process is correct. Then we split the training data into 2 parts --- one for training and one for validation comprising 90% and 10% of the dataset respectively. The validation set is used to tune the hyper-parameters. After data splitting, we start to train the model. We also do zero padding for all datasets to make sure that all batches have same dimensions. Hence, our model can focus on non-zero values during training process. Zero padding accelerate and optimize the learning process. That's why we find the accuracy is close to 100% when we only run 10 epochs. Then we applied the trained model to test data.

Result

We apply both the LSTM model and the BERT model to same training set and test set. The result are depicted in Table 3.

Model	Training set	Test set
LSTM	88%	68%
BERT	98%	79%

Table 3: Comparison of prediction accuracy of the LSTM model and the BERT model.

The prediction accuracy of test set of the BERT model is 10% higher than that of the LSTM model, aligned with the various strengths of BERT. The MLM of BERT allows us to pre-train a deep bidirectional Transformer while the LSTM can only implement unidirectional learning process. As a result, the BERT model can better utilize the information in the whole context of tweets and show higher prediction accuracy.

Conclusion

In this paper, we build a LSTM based model and a fine-tuned BERT model to predict the overall sentiment (positive, negative or neural) in a tweet dataset and compare the performance of these 2 models. The result shows the BERT model outperforms the LSTM model at the prediction accuracy of the overall sentiment of tweets.

Reference

1. Boyd, Danah M., and Nicole B. Ellison. "Social network sites: Definition, history, and scholarship." Journal of computer-mediated Communication 13.1 (2007): 210-230.
2. Cambria, Erik, et al. "New avenues in opinion mining and sentiment analysis." IEEE Intelligent systems 28.2 (2013): 15-21.
3. Garain, Avishek, and Sainik Kumar Mahata. "Sentiment analysis at sepln (tass)-2019: Sentiment analysis at tweet level using deep learning." arXiv preprint arXiv:1908.00321 (2019).
4. Gao, Wei, and Fabrizio Sebastiani. "From classification to quantification in tweet sentiment analysis." Social Network Analysis and Mining 6.1 (2016): 19.
5. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
6. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
7. Polignano, Marco, et al. "Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets." Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019). CEUR. 2019.
8. https://www.kaggle.com/c/tweet-sentiment-extraction/overview