EXPLORING THE USE OF BERT FOR MEASURING SEMANTIC SIMILARITY IN NLP APPLICATIONS

AKHILANAND KUNDURTHI,

GRADUATE STUDENT IN COMPUTER SCIENCE AT UNIVERSITY OF WISCONSIN-MILWAUKEE.

ADVISOR: DR. ROHIT J KATE,

DEPARTMENT OF COMPUTER SCIENCE, UWM-MILWAUKEE.

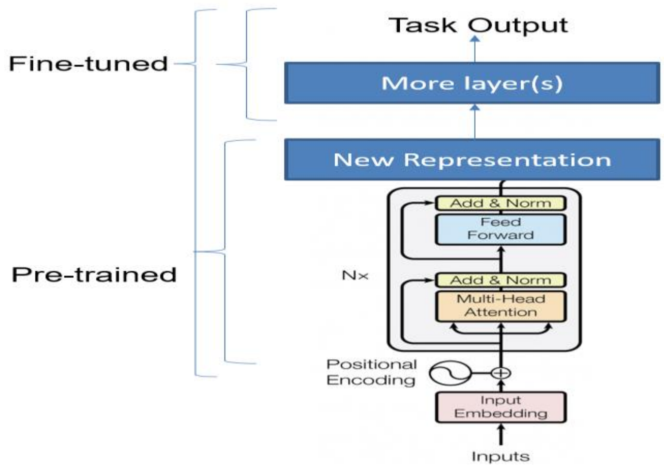
INTRODUCTION

Semantic Similarity is a measure of how similar two pieces of text are in terms of the meaning they convey. It measures the degree to which two concepts, ideas, or words are related semantically, i.e., in terms of their meaning. It is a measure of the closeness of their meanings, regardless of their syntactic or morphological differences. This project aims to build a model that can accurately predict the semantic sentence similarity between two sentences. I have used and fine-tuned a BERT model, which will take two sentences as inputs and output a real-valued similarity score for the two sentences. The applications of this project are machine translation, text classification, text entailment, plagiarism detection, information retrieval, dialogue systems. In the clinical domain, semantic text similarity can be used to make the clinical decision-making process easier and more efficient by detecting and eliminating redundant information present in the electronic health record, thereby reducing the cognitive burden of the clinician [2].

BERT MODEL:

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model developed by Google in 2018. It is a deep learning-based natural language processing (NLP) pre-training model built on top of the Transformer neural network architecture. BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to determine the intent. The model can be used to create state-of-the-art natural language processing systems for a variety of language understanding tasks. It generates context-based input representations, which can be used for NLP tasks like semantic textual similarity. For this task, the encoded sentence pairs are given as the inputs at the input embedding layer.

ARCHITECTURE OF BERT MODEL:



Pretraining and fine-tuning are two important steps in the development of a BERT model. Pretraining is the process of training a model on a large dataset with unlabeled data to learn language representations. Fine-tuning is the process of further adapting the model to specific tasks, such as semantic text similarity, with labeled data. During fine-tuning, the weights of the BERT model are adjusted to best solve the task at hand.

METHODOLOGY

DATASET:

The dataset is taken from the semantic text similarity (STS) benchmark, which gathers English datasets that have been utilized in the STS tasks held as a part of SemEval from 2012 to 2017. These datasets include sentences from image captions, news headlines, and user forums. The dataset consists of 5749 sentence pairs for training, 1500 sentence pairs for validation, and 1379 sentence pairs for testing.

LIBRARIES UTILIZED: NumPy, pandas, TensorFlow, transformers, keras.

TOOLS USED: Python 3.8

PREPROCESSING THE DATASET:

- The dataset contains various columns. For the task, only sentence 1 and sentence 2 columns have been chosen, and the score column as a target and extracted to a CSV file.

Sample Training Data:

Sentence 1	Sentence 2	Score
A plane is taking off.	An airplane is taking off.	5
A man is smoking.	A man is skating.	0.5
A man is spinning.	A man is dancing.	1
A woman is dancing.	A man is talking.	0

- From the BERT tokenizer's library, using batch_encode_plus, both sentences are encoded together and separated by [SEP] token. I have chosen the maximum number of tokens to be generated as 300 for efficient training. The encoded features are converted to a NumPy array using the pandas library.

Sample Tokenized Data:

```
['A kid plays with a toy phone.' 'A little boy plays with a toy phone.']
[ 101 1037 4845 3248 2007 1037 9121 3042 1012  102 1037 2210 2879 3248
 2007 1037 9121 3042 1012  102    0    0    0    0    0    0    0    0
```

```
['The woman is frying a breaded pork chop.'
 'A woman is cooking a breaded pork chop.']
[  101  1996  2450  2003 14744  2075  1037  7852  2098 15960 24494  1012
   102  1037  2450  2003  8434  1037  7852  2098 15960 24494  1012   102
```

BUILDING THE MODEL:

From the TensorFlow and Keras library, the pretrained Bert base uncased model, which is already pretrained on millions of datasets is loaded. The encoded features are given as the input to the loaded model. Since Bert generates sequence and pooled output, I used the sequence output for fine-tuning.

FINE-TUNING THE MODEL:

The sequence output is fed through bi-lstm layer. The bi-lstm layer output is fed through two pooling layers: a global average pooling layer to reduce the input data's dimensionality and noise, and a global max pooling layer to extract the most relevant features and those two layers outputs are concatenated. The concatenated output is fed through the dropout layer of dropout value 0.3 to prevent overfitting. Since our task is regression, the dropout layer is fed through a dense layer without an activation function which is linear by default. For delivering a meaningful improvement of results, the Adam optimizer with a learning rate of 3e-5 and metrics mean squared error [MSE] and loss mse have been used.

TRAINING THE MODEL: The model is compiled and trained from end-to-end with epochs of 15 with a batch size of 32.

TESTING THE MODEL: The model is evaluated on the test dataset. The predictions of the test data are extracted to a CSV file.

MODEL SUMMARY:

```
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_ids (InputLayer)         [(None, 300)]        0           []

 attention_masks (InputLayer)   [(None, 300)]        0           []

 token_type_ids (InputLayer)    [(None, 300)]        0           []

 bert (TFBertMainLayer)         TFBaseModelOutputWi  109482240   ['input_ids[0][0]',
                                thPoolingAndCrossAt               'attention_masks[0][0]',
                                tentions(last_hidde               'token_type_ids[0][0]']
                                n_state=(None, 300,
                                 768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 bidirectional (Bidirectional)  (None, 300, 128)     426496      ['bert[0][0]']

 global_average_pooling1d (Glob  (None, 128)         0           ['bidirectional[0][0]']
 alAveragePooling1D)

 global_max_pooling1d (GlobalMa  (None, 128)         0           ['bidirectional[0][0]']
 xPooling1D)

 concatenate (Concatenate)      (None, 256)          0           ['global_average_pooling1d[0][0]'
                                                                 , 'global_max_pooling1d[0][0]']

 dropout_37 (Dropout)           (None, 256)          0           ['concatenate[0][0]']

 dense (Dense)                  (None, 1)            257         ['dropout_37[0][0]']

==================================================================================================
```

The above picture shows that tokenized inputs of size 300 are given to the BERT model and fed through multiple layers, and at the dense layer, the similarity score is generated.

RESULTS

- The model has achieved a mean squared error [MSE] of 1.03 on the testing set when making predictions.
- Upon manual checking, of the 97 highly equivalent sentence pairs in the test set, the model has predicted 48 pairs as equivalent and 16 as roughly equivalent, which are 50% equivalent and 16.5% roughly equivalent.

- The model was able to accurately predict almost all the sentence pairs that had the lowest degree of similarity.
- The limitation of the model is that the training set does not have various high-similarity degree examples. As a result, the model is not able to accurately predict the most similar examples.

Table of Results

Sentence 1	Sentence 2	Actual Score	Predicted Score
Two men are fistfighting in a ring.	Two men fistfight in a ring.	5	5.082138
Senate confirms Janet Yellen as chair of US Federal Reserve.	Senate confirms Janet Yellen as next Federal Reserve Chair.	5	4.845632
A young pitcher is throwing the baseball.	A brown dog is walking on the grass beside a fence.	0	0.016776
Obama Struggles to Soothe Saudi Fears As Iran Talks Resume.	Myanmar Struggles to Finalize Voter Lists for Sunday Polls.	0	0.036723
A girl is styling her hair.	A girl is brushing her hair.	2.55	2.557038

CONCLUSION

In conclusion, this project has demonstrated the effectiveness of using a BERT model to accurately predict semantic sentence similarity between two sentences. This model can be fine-tuned and used for various other tasks, such as clinical STS mentioned in [2], and find the similarity to reduce the redundancy of the patient information, CORD19 STS said in [3] for building information retrieval engines calibrated precisely for COVID-19. Future work can focus on improving the model by adding more examples to the training set and training the model with more epochs. It may be necessary to adjust the model's parameters such as adding number of layers, the size of the hidden state, the number of attention heads, the number of tokens in the input, the type of pre-training, the learning rate, the optimizer, the batch size to improve the accuracy of predictions.

REFERENCES

[1]. Cer, D. (2017b, July 31). SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual. . . arXiv.org. https://arxiv.org/abs/1708.00055

[2]. Wang, Y., Afzal, N., Fu, S., Wang, L., Shen, F., Rastegar-Mojarad, M., & Liu, H. (2018). MedSTS: a resource for clinical semantic textual similarity. Language Resources and Evaluation, 54(1), 57–72. https://doi.org/10.1007/s10579-018-9431-1

[3]. Xiao Guo, Hengameh Mirzaalian, Ekraam Sabir, Ayush Jaiswal, & Wael AbdAlmageed. (2020). CORD19STS: COVID-19 Semantic Textual Similarity Dataset. Cornell University - ArXiv. https://doi.org/10.48550/arxiv.2007.02461