Natural Language Processing - Semantic Similarity

Himanshu Yadav
hy19558it2019@gmail.com
Bachelor of Technology - Information Technology, 4th year
Rajasthan Technical University
Kota, Rajasthan, India (324010)

1 ABSTRACT

Natural language processing (NLP) is an artificial intelligence discipline that focuses on the study and control of human language. In recent years, several transformer based models have emerged to aid in the development of better NLP applications. Transformers can provide high quality embeddings. In this assignment, we predict the semantic similarity for the given sentence and its reason. Alternatively, determine whether the given reason will satisfy the given text. To solve the problem, we fine-tuned the MPNet base model. To avoid overfitting, we also used data augmentation.

2 DATA INSIGHTS

The cosine similarity, which uses embeddings to increase accuracy in the supplied dataset, enables a stronger semantic comparison between the given text and reason.

Training dataset only contains one class, or "1" if text meets the reason. The right decision boundary must be selected in order to divide the outputs into two classes without categorising all inputs into a single category or class. The embedding is smoothed using mean pooling, which stops the Transformer from classifying data incorrectly.

Data augmentation can be utilised to give the classification models a better understanding of the dataset and prevent overfitting. It makes use of the tool Sentence Augmenter, which may be used to raise the quality of your sentence context tensors. This is accomplished by utilising word embeddings from a variety of sources, including WordNet, Word2Vec, and GloveVec. It permits word additions or deletions without changing the semantic meaning of the phrase.

During data augmentation, we transform the $(Text, Reason)$ and create $(Text', Reason')$ out of it, while still preserving the label $(Label)$.

$$(Text, Reason, Label) \implies (Text', Reason', Label) \tag{1}$$

3 TRAINING APPROACHES

MPNet combines strengths of masked and permuted language modeling for language understanding. BERT is one of the most successful pre-training models because it uses masked language modelling (MLM) but it fails to account for dependency among predicted tokens, XLNet employs permuted language modelling (PLM) for pre-training to address this issue. However, because XLNet does not use all of the position information in a sentence, there is a position discrepancy between pre-training and fine-tuning. On the other hand, MPNet, a novel pre-training method that combines the benefits of BERT and XLNet while avoiding their drawbacks. MPNet

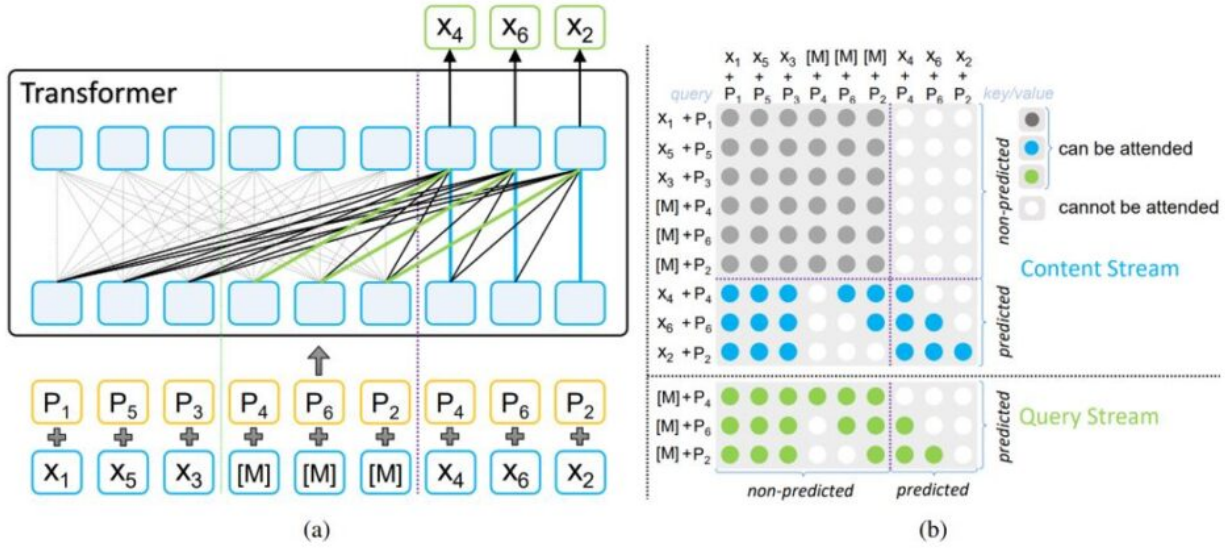

Figure 1: The model structure of MPNet (ref: here)

Table 1: Ablation Study Table

Base Model - MPNet						
Accuracy	**Precision Score**	**Recall Score**	**F1 Score**	**Brier Score**	**BCE Loss**	**MCC**
0.7529	0.6019	0.7644	0.6735	0.2471	8.5351	0.4881

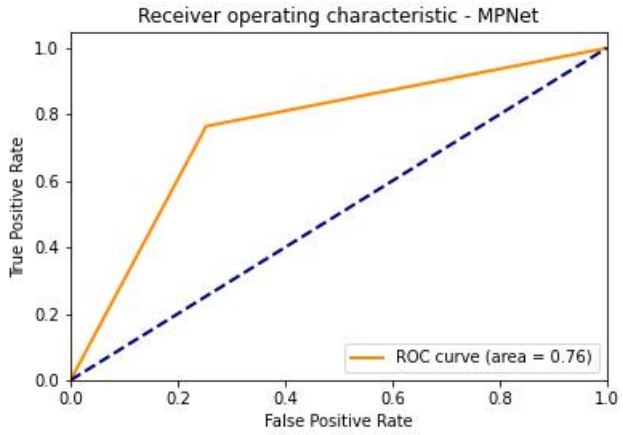

Figure 2: ROC Curve

takes advantage of predicted token dependency through permuted language modelling (vs. MLM in BERT) and uses auxiliary position information as input to make the model see a full sentence, reducing position discrepancy (vs. PLM in XLNet).After being trained on a large-scale dataset, MPNet was fine-tuned on a variety of downstream tasks (GLUE, SQuAD, and so on). Experiment results show that MPNet outperforms MLM and PLM by a large margin, as well as previous state-of-the-art pre-trained methods (e.g., BERT, XLNet, RoBERTa) in the same model setting.

4 ERROR ANALYSIS

The model precision score measures the proportion of positively predicted labels that are correct and the recall score represents the model's ability to correctly predict the positives out of actual positives. As in Table: 1, the precision score is low, and the recall score is high. This indicates that the model is slightly biased in favour of class "1" because it properly predicts class "1," which satisfies the text, and also incorrectly predicts some inputs from class "0." Low F1 score causes the BCE Loss to increase.
In Figure 2, The model will be able to identify between the positive class and the negative class 76% of the time as per the AUC of 0.7529, approx. 0.76. Hence, we see superior semantic comparison is being provided by this architecture.

5 CONCLUSION

In this assignment, we demonstrated how to modify a pre-trained transformer model to provide good embedding while also customising it to meet our specific needs. We used the data augmentation strategy to reduce over-fitting, and used mean pooling to smoothen the embeddings so that the MPNet transformer classify output into two classes instead of one. The results are shown in Table: 1.