Entailment Reasoning with Fine-Tuned Contextualized Embeddings using Natural Language Inference

Mukul Puranik
Univ. Southern California
Los Angeles, CA
mpuranik@usc.edu
USC ID: 9943870051

Akshay Bhadra
Univ. Southern California
Los Angeles, CA
bhadra@usc.edu
USC ID: 3484524247
(DriveLink)

Khyati Suratwala
Univ. Southern California
Los Angeles, CA
suratwal@usc.edu
USC ID: 3811385001

Abstract

The task at hand is to perform Natural Language Inference on a pair of sentences, to understand the relation between the two sentences. In Natural Language Inference, we define if the statement entails, contradicts or has a neutral relation with the hypothesis. This has applications like information retrieval, semantic understanding and context question answering. Succeeding at such tasks does not require a system to evaluate complex optimizations but only that which encompass the meaning or better semantically represent the data (i.e their lexical and compositional semantics).

1 Introduction

The task here is to successfully have representations of both the given data and the hypothesis and check if their representations share context in different dimensional space. If they do, then the statement entails the hypothesis, if they have opposite context, then the statement contradicts the hypothesis, but if they dont have similar or contradicting contexts, then they might have a neutral relation.

The challenges of NLI are quite different from those encountered in formal deduction: the emphasis is on informal reasoning, lexical-semantic knowledge, and variability of linguistic expression. So to tackle some of the challenges, our focus is to mainly pack as much context in sentences and match that with the context generated by the hypothesis, if it is similar or contradictory, then we can easily decode that the sentences entail, contradict or have a neutral relation.

2 Baseline

Fine tuned BERT (Bidirectional Encoder Representations from Transformers)Devlin et al. (2018) is used as a baseline for the task. BERT is a pretrained model that

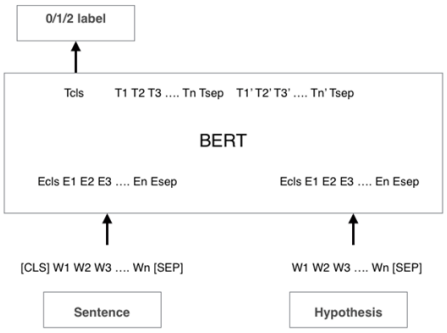

Figure 1: BERT Finetuned Model for NLI

is used for different tasks like classification, entity recognition, question answering using transfer learning. BERT has given state-of-art results for many tasks by fine tuning on task specific data using less space and time. BERT is trained on a huge generalised data to give a pretrained model with weights. A new layer can be added on top of BERT and trained on a small task specific dataset. The whole model is trained on this dataset only for a few epochs. The lower layers are finetuned according to the new data. We can also freeze some layers while training for the task, but this can decrease the accuracy, if the data used for BERT and the task are very different.

For our task, we have used MNLI dataset (Williams et al., 2018) for fine tuning BERT for predicting if the sentences entail, contradict or have a neutral relation. We examined the whole dataset and found only some specific columns to be appropriate to achieve our baseline BERT model. We decided to use *sentence1*, *sentence2* and *'gold_label'* columns from the dataset for our NLI task. The final layer will have 3 out-features at the end, corresponding to the 3 results. The baseline model is

trained for 10 epochs. The MNLI data is preprocessed to convert the data in a format that can be fed to BERT.

The whole MNLI train dataset is used, and the batch size is set to 128. The input size and batch size are tuned to give the maximum accuracy, limited by system restrictions. A special token [CLS] is added in the beginning of the input: sentence and hypothesis. [SEP] is added in between the sentences. BERT tokeniser is used to convert the input words into tokens that are recognised by BERT model. The final token representation for [CLS] is an aggregation of the whole sentence. Segment Mask is used to identify between sentence 0 and 1 of the input. Attention mask masks the padded part of each sentence. Labels are stored as 0 for entailment, 1 for contradiction and 2 for neutral relation.

BertForSequenceClassification pytorch interface of huggingface is used to implement this task, curated according to MNLI data and the output needed. Pretrained model for Bert-base-uncased dataset is used, which consists of lower-cased letters. Word_embeddings for vector representation of words, position_embeddings for vector representation of a word based on its position, token_type_embeddings for representing the sentence the word belongs to, are used in the BERT model for input representation in this task. There are 12 layers in BERT pretrained model, each layer consists of an attention head, intermediate representation and output.

3 Improvements to Baseline

3.1 RoBERTa

RoBERTa stands for Robustly Optimized BERT approach (Liu et al., 2019), was introduced by facebook months after Google's BERT. It stands as an overall better model than BERT leveraging BERT's potential with proper pre-training. RoBERTa improves upon Google's BERT as even the base model of RoBERTa was trained on a huge 160GB of text, which also included the small dataset BERT was initially trained upon. The additional data was crawled from various news datasets as well as websites. This coupled with the fact that Facebook used a whopping 1024 GPUs alongside, allowed them faster compute time.

To improve upon the training further, the team at Facebook decided to ignore the Next Sentence Prediction(NSP) task from BERT's pretraining and introduced a new masking mechanism which dynamically changes the masked tokens in the input. Large training batch-sizes coupled with 3 times the pretraining compute steps, RoBERTa clearly beats most state of the art and proved to be an improvement on BERT.

3.2 Roberta and Contextualized Embeddings

In our proposed model, we use the pretrained weights of RoBERTa to create contextualized embeddings of our word.

What are Contextualized Embeddings?

Contextualized Embeddings basically are embeddings of each word with respect to the context in which that word was used. In comparison to GloVe or word2vec which generate same embedding for the word irrespective of the meaning or the context in which it was used. *e.g:* Consider the sentence:

> 'He was planning to rob the bank and flee through the secret passageway on the river bank.'

Here the word bank has two different meanings and its vital to get the correct interpretation of the word in order to better understand the meaning. By translating a word to an embedding it becomes possible to model the semantic importance of a word in a numeric form and thus perform mathematical operations on it. For our task, we are basically provided with two sentences which either are close in meaning, diverge from context or are not as such related to the context. This task in itself calls out for understanding the context of statement and hypothesis and basically classify if they point towards the same context.

RoBERTa has 12 Bert Encoders each generating a 768 dimensional vector for each word. Each word follows its own unique path and hence using the attention(Vaswani et al., 2017) mechanism, each word therefore has its own unique embedding with respect to the context it was used in. Although RoBERTa or BERT were not actually designed as embedding generators like ElMo(Peters et al., 2018), it has been studied that they produce similar vectors for particular word used in same context and different vectors for the same word used in another context.

We propose our model leveraging this context based vectors formed by RoBERTa and training them extensively, to get a proper embedding fine-tuned for our dataset. This is then passed on to transformers and/or BiLSTM to learn the attentions as well as the positional data.

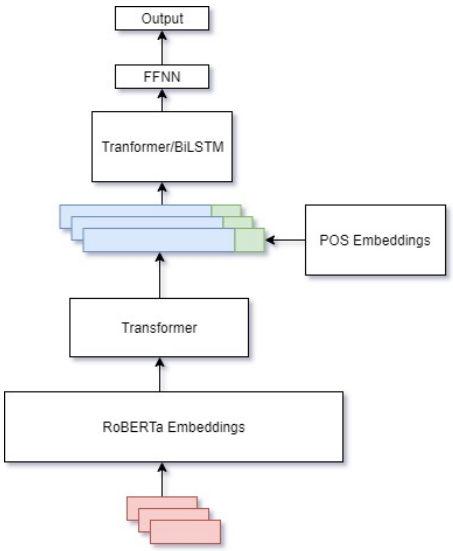

Figure 2: Proposed "Transformed RoBERTa" Model

4 Proposed Model

The intuition behind our proposed model is basically to leverage the contextualized word embeddings, train our own transformer to generate more finetuned embeddings for each word which are then concatenated with the POS embeddings of each word. This is followed by BiLSTM/Transformer to get the generalized sentence embeddings for both the sentences and feed them to the huge feed forward network for prediction of class.

Dataset
The dataset consisted of approximately 400,000 sentences for training. Each sentence consisted of the statement concatenated by the hypothesis, both separated by RoBERTa's sentence separator. The training dataset was then further divided into Training-Validation pairs. Each sentence, consisted of its parse-tree which was leveraged and used as POS tag for that particular word in the sentence.

Each sentence-hypothesis concatenated pair was then truncated till 128 length. This was a major compute power issue, since the amount of VRAM was limited. Smaller sentences were concatenated with zero-padding to match the 128 length of each sentence.

Different Approaches
For each of our approach, we divided our training data into 90/10 Training/ validation split and stopped training after convergence. Then each of the model was tested on dev_matched and dev_mismatched dataset which was treated as unseen test data. The metrics used to evaluate each model were Matthews Correlation Coefficient(MCC), accuracy, individual class accuracy. MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

4.1 Finetuned RoBERTa with BiLSTM

In this approach the intuition was to get the embeddings from RoBERTa, and inject historical positional information into it with the contextualized embeddings, to generate a sentence-level embedding for each sentence. Each sentence here was represented by a 768 dimensional vector, which was then fed to a feedforward neural network which predicted each class. When we trained vanilla RoBERTa on our dataset, it was outperforming BERT within the first epoch itself. So to leverage this fact, and get more positional context, BiLSTM layer worked pretty well, outperforming Vanilla RoBERTa. This method although got the positional context, lacked the context-specific fine-tuned word embedding. RoBERTa embeddings, although trained on huge amounts of data, were not trained to learn our specific domain/training data contexts.

4.2 Finetuned RoBERTa with Tranformed Embeddings into BiLSTM

Due to the domain specific embeddings limitation of BiLSTM, we moved forward with our own Tranformers-based approach. The transformer consisted of multiple headed-self attention layers, followed by dense layer which was then forwarded to BiLSTM. The BiLSTM here was responsible in generating the positional context of the entire training sentence. This embedding was then passed on to the classifier to generate results. We saw that when we applied this trick, there was increase in the Matthews Correlation coefficient, indicating that indeed this method curbed the number of false positives and false negatives.

The way this works is, the transformer again uses the attention mechanism and concentrates on specific parts of our input. This inturn lets us encode bert-like embeddings for each of the input

sentence's word. The input sequence word is then passed on to the BiLSTM to get sentence level representation of each word in the form of the hidden vector. Once the hidden vector is trained, it is passed on the linear classifier which gives 3 outputs.

4.3 Transformed RoBERTa

As we saw, adding in more transformer encoding layers, was helpful, we tried digging deeper into a pure transformer-like embedder which would be passed on to the classifier. The intuition behind this one was, since we were using transformers for contextualized word embeddings, why not use the transformer again for sentence level embedding. Hence our revised model, consisted of dual transformer encoders, with each transformer having 2 layers of self-attention. The second transformer generated sentence-level embeddings which were then passed onto the fully connected layer for prediction.

This model outperformed RoBERTa with transformed embeddings, and clearly showed the way to go forward for future research. We wanted to further experiment having multiple transformers take in contextualized embeddings followed by BiLSTM to get the positional contextualized sentence embeddings, but were faced with limited compute power. The Google Colaboratory only provides GPU with 16GB of Memory, which led to compute limitations.

5 Results

5.1 BERT-Baseline

We considered BERT-base version as our baseline. This was necessary since BERT is state of the art and transfer learning upon bert is the way forward. BERT model started to converge after just 3 epochs over the training data. Over the period of our research, we found that the BERT model was consistently beaten and was giving considerably low accuracies and MCC values.

5.2 RoBERTa Finetuned

To beat our baseline we first checked if going forward with RoBERTa model is the right direction and clearly RoBERTa outperformed BERT in all the accuracies plus MCC. RoBERTa model was getting converged in 4 epochs. After 4 epochs, the model started overfitting with increase in validation loss.

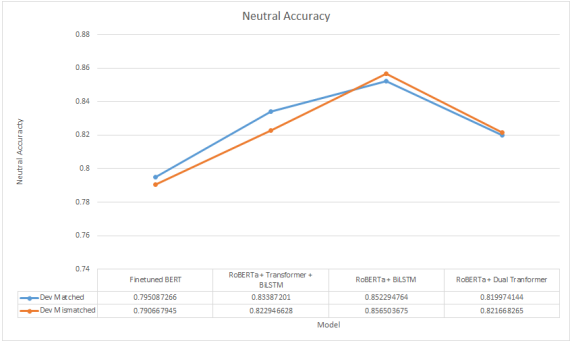

Figure 3: Neutral Class Accuracy vs Model

Some important findings from RoBERTa were that firstly, compared to BERT, RoBERTa is a better model and classifier. The MCC score of RoBERTa 0.792 was significant improvement over BERT's 0.75. It showed that not only RoBERTa was displaying higher number of True Positives and True Negatives, it did so while not increasing False Positives and False Negatives. This model was performing way better than baseline and hence we decided to build upon this model as our embedding generator.

5.3 RoBERTa + BiLSTM

The intuition behind adding a BiLSTM layer over RoBERTa was to encapsulate positional embeddings and create a overall sentence embedding from the word embeddings of RoBERTa. As we expected, RoBERTa + BiLSTM did give us better accuracy but the number of FP/FN increased, hence although the MCC score of this model was 0.788, a bit lower than vanilla RoBERTa 0.792, it did increase the amount of TP and TNs(overall accuracy) which is a lot beneficial in NLI tasks.

Another interesting finding is that adding BiLSTM layer, increased the accuracy for *Neutral* class. It is obvious that the semantic context of Neutral relation between statement and hypothesis would be harder to capture, but the BiLSTM layer succeeds in increasing the Neutral class accuracy from 0.847 to 0.858 for dev_mismatched dataset. The results show us that introducing BiLSTM layer on top was indeed beneficial, it was able to produce good sentence-level embeddings used for classification.

5.4 RoBERTa + Transformer + BiLSTM

The main goal behind introducing a transformer encoder was to get data-specific contextualized embeddings. These embeddings were then fed to BiLSTM with the notion that transformer's new embed-

Model	MCC	Dev(matched/mis_matched)
BERT-Baseline	0.757/0.755	0.838/0.836
RoBERTa-Finetuned	0.792/**0.797**	0.861/0.862
RoBERTa + BiLSTM	0.788/0.786	0.858/0.856
RoBERTa + Transformer + BiLSTM	0.797/0.787	0.864/0.858
Transformed RoBERTa	**0.798**/0.796	**0.865/0.864**

Table 1: Accuracy Comparison of Models.

dings will capture better context which would then be remembered by BiLSTM more, inturn giving us better sentence-level embeddings.

As expected, this model outperformed RoBERTa + BiLSTM, in the sense it clearly had better MCC score (0.796) and Test data set accuracy(Matched/Mismatched). But one important fact to know was, the addition of transformer's new attention layers led to it focusing more on parts which were prominent in each sentence (e.g : leaning towards same meaning/ leaning towards contradictory meaning). But this inadvertently led it to

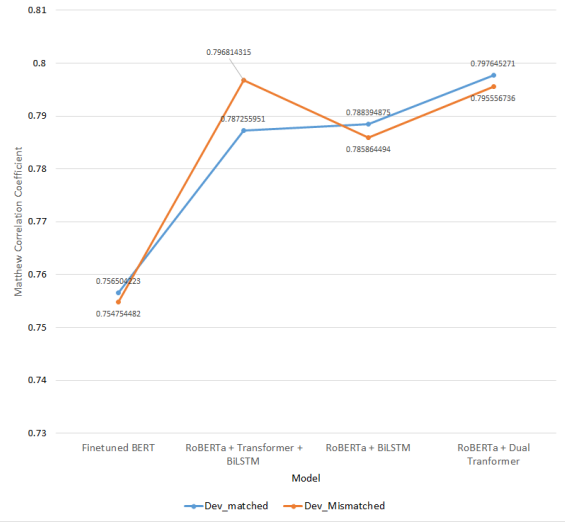

Figure 4: MCC vs Model

focus less on unimportant contexts like those in Neutral hypothesis. The accuracy for Neutral on dev_mismatched went down from 0.858 to 0.822. But the accuracy for Entailment increased from 0.844 to 0.86. The results show that addition of transformer's focussed encoding leads to more attention to distinct features.

5.5 Transformed RoBERTa

This model was the best of the lot in terms of accuracy score. The model consisted of two transformer encoders each having 2 layers respectively. The intution behind this approach was to get more accuracy by generating even better word-level embeddings. With each word having better context, the overall sentence embedding would also be decent leading to better accuracy results.

General observation is that the transformer fails to identify Neutral hypothesis compared to base RoBERTa or even the BiLSTM model. The Neutral class-based accuracy for Transformed RoBERTa(0.822) was considerably low when compared to vanilla RoBERTa(0.848) or RoBERTa with BiLSTM(0.856). But on the otherhand, Transformer was performing better in handling and understanding semantic differences between entailment and contradiction. Addition of the BiLSTM layer seems to increase the Neutral accuracy as seen in the RoBERTa + Transformer + BiLSTM model which showed Neutral accuracy of 0.8338. The contradiction and entailment class accuracy of this model was highest when compared to other models.

Overall the future scope for this project would be to purchase better compute power and have a LSTM network follow Transformed RoBERTa to give a better sequential embeddings and increase the neutral class-based accuracy and thus improving the overall accuracy and MCC for Entailment Reasoning.

6 Comparisons

During training of model, approximately each model took 100 mins per epoch. Most of the models converged within 3-4 epochs. The validation accuracy for all the models are shown in the figure 5 below.

All the models outperformed the baseline with Transformed RoBERTa giving the best validation set accuracy. Moreover, the convergence rate for Transformed RoBERTa was faster, as can be seen from the huge jump and convergence.

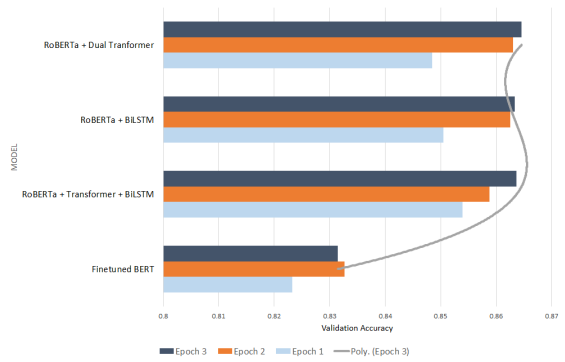

Figure 5: Validation Accuracy per epoch

We used dev_matched and dev_mismatched as our test dataset. We compared accuracies of each of our models against these. Our results show that the best model to go forward with is the Transformed RoBERTa i.e dual transformers following RoBERTa. It beats all the other models in accuracies.

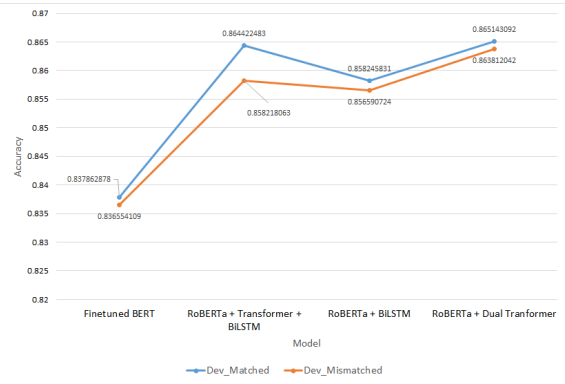

Figure 6: Dev_set Accuracy vs Model

Not only accuracy, as can be seen from the above diagram, the Transformed RoBERTa model outperforms every other model in terms of their Matthew Correlation Coefficient, thereby letting us know that not only the overall accuracies but the individual classes are also classified equally well. By this we mean the number of false positives and false negatives are considerably low.

7 Conclusions

Hence going forward, correct approach would be to get much better contextualized embeddings of each word by using multiple transformers with higher dimensions to obtain better features and then passing them through BiLSTM layers which would compress the individual contexts with sequential information, necessary to identify neutral hypothesis and give out proper predictions. The advantages and disadvantages of each method are laid out by the research undertaken, each boosting some property while sacrificing other.

The model that we have finalized for this project is the Transformed RoBERTa model and going forward we aim to increase its accuracy by increasing the batch_size from 128 to 512 and increasing the Transformers encoding dimensions in addition to increase in the hidden dimension of BiLSTM.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Project DriveLink. Includes all the project notebooks, datasets and model weights. https://drive.google.com/drive/folders/1XBd9KNz3u7vvkjch4nqQ0PMczy5fYApb. Accessed: 2019-12-06.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.