SEMANTIC TEXTUAL SIMILARITY

TEAM DH

Project Report

Tanishq Goel (2019114015)
Veeral Agarwal (2019114009)

PROBLEM STATEMENT:

In this project work we implement a scale down version of the paper Bilateral multi-perspective matching with one matching layer to solve the problem of semantic textual similarity. We also fine tuned on pre-trained transformer models of RoBERTa, XLNet and BERT and observed an improvement in accuracy of close to 10%. We also did a qualitative analysis of the errors reported by each model and compared the performances with one another.

INTRODUCTION:

Semantic textual similarity is the task of comparing two sentences and identifying the semantic relationship between them. It is also about determining how similar two pieces of texts are. For example a paraphrase identification task is to identify whether two sentences are paraphrases or not. Another task is the natural language inference task where the model has to check if the hypothesis sentence can be inferred from a premise sentence. This also can take the form of assigning a score from 1 to 5. The intuition is that sentences are semantically similar if they have a similar distribution of responses. It helps in improving the natural language understanding of the systems.

RELATED WORKS:

There has been much work in the past on finding the relationship between the sentences. The first framework is based on the “Siamese” architecture. In this framework, the same neural network encoder (e.g., a CNN or a RNN) is applied to two input sentences individually, so that both of the two sentences are encoded into sentence vectors in the same embedding space. Then, a matching decision is made solely based on the two sentence vectors. The

with adding more tasks, jointly training them with a skip-thought-like model that predicts sentences surrounding a given selection of text has been used for this task.

ARCHITECTURE AND MODEL OVERVIEW:

Bilateral multi-perspective matching (BIMPM) model(Baseline):

Given a pair of sentences P and Q, the BiMPM model estimates the probability distribution Pr(y|P, Q) through the following five layers.

1. **Word Representation Layer :** The goal of this layer is to represent each word of the sentences P,Q with a d-dimensional vector. The d-dimensional vector had 2 components, the word embedding and character embedding concatenated together. The word embeddings were taken from pre-trained Glove vectors and the character embeddings is calculated by feeding each character within a word into a LSTM. The character embeddings are learned jointly with other network parameters during training.
   The output of this layer are two sequences of word vectors P : [$\mathbf{p}_1$, ..., $\mathbf{p}_M$] and Q : [$\mathbf{q}_1$, ..., $\mathbf{q}_N$]. Word embedding dimension is kept as 300 and character embedding dimensions are taken as 20.

2. **Context Representation Layer :** The purpose of this layer is to incorporate contextual information into the representation of each time step of P and Q using 2 BiLSTMs, one for each sentence P,Q.

$$\overrightarrow{\boldsymbol{h}}_i^p = \overrightarrow{\text{LSTM}}(\overrightarrow{\boldsymbol{h}}_{i-1}^p, \boldsymbol{p}_i) \qquad i = 1, ..., M$$
$$\overleftarrow{\boldsymbol{h}}_i^p = \overleftarrow{\text{LSTM}}(\overleftarrow{\boldsymbol{h}}_{i+1}^p, \boldsymbol{p}_i) \qquad i = M, ..., 1$$

$$\overrightarrow{\boldsymbol{h}}_j^q = \overrightarrow{\text{LSTM}}(\overrightarrow{\boldsymbol{h}}_{j-1}^q, \boldsymbol{q}_j) \qquad j = 1, ..., N$$
$$\overleftarrow{\boldsymbol{h}}_j^q = \overleftarrow{\text{LSTM}}(\overleftarrow{\boldsymbol{h}}_{j+1}^q, \boldsymbol{q}_j) \qquad j = N, ..., 1$$

3. **Matching Layer :** The goal of this layer is to compare each contextual embedding of one sentence against all contextual embeddings of the other sentence. We utilize a full matching layer in which each forward (or backward) contextual embedding is compared with the last time step of the forward (or backward) representation of the other sentence helps in comparing two sentences.

$$\overrightarrow{\boldsymbol{m}}_i^{full} = f_m(\overrightarrow{\boldsymbol{h}}_i^p, \overrightarrow{\boldsymbol{h}}_N^q; \boldsymbol{W}^1)$$
$$\overleftarrow{\boldsymbol{m}}_i^{full} = f_m(\overleftarrow{\boldsymbol{h}}_i^p, \overleftarrow{\boldsymbol{h}}_1^q; \boldsymbol{W}^2)$$

4. **Aggregation Layer :** This layer is employed to aggregate the two sequences of matching vectors into a fixed-length matching vector. Another BiLSTM model is applied to the two sequences of matching vectors individually. Then, a fixed-length matching vector is constructed by concatenating vectors from the last time-step of the BiLSTM models (shown in green in Fig. 1).
5. **Prediction Layer :** The purpose of this layer is to evaluate the probability distribution Pr(y|P, Q). This layer consisted of a two layer feed-forward neural network to consume the fixed-length matching vector and softmax was applied in the output layer to give the final output which is a vector of size equal to the number of labels consisting of probabilities for each label.

We ran the model for 15 epochs observing a decrease in training loss and the best accuracy was achieved at the point when the model started to overfit(By observing validation loss) with a batch size of 128 and learning rate of 1e-3 with ADAMW optimizer.

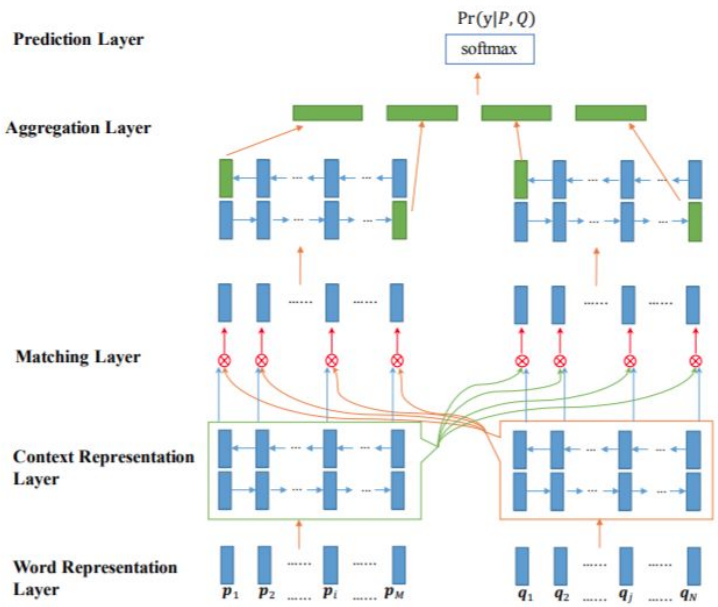


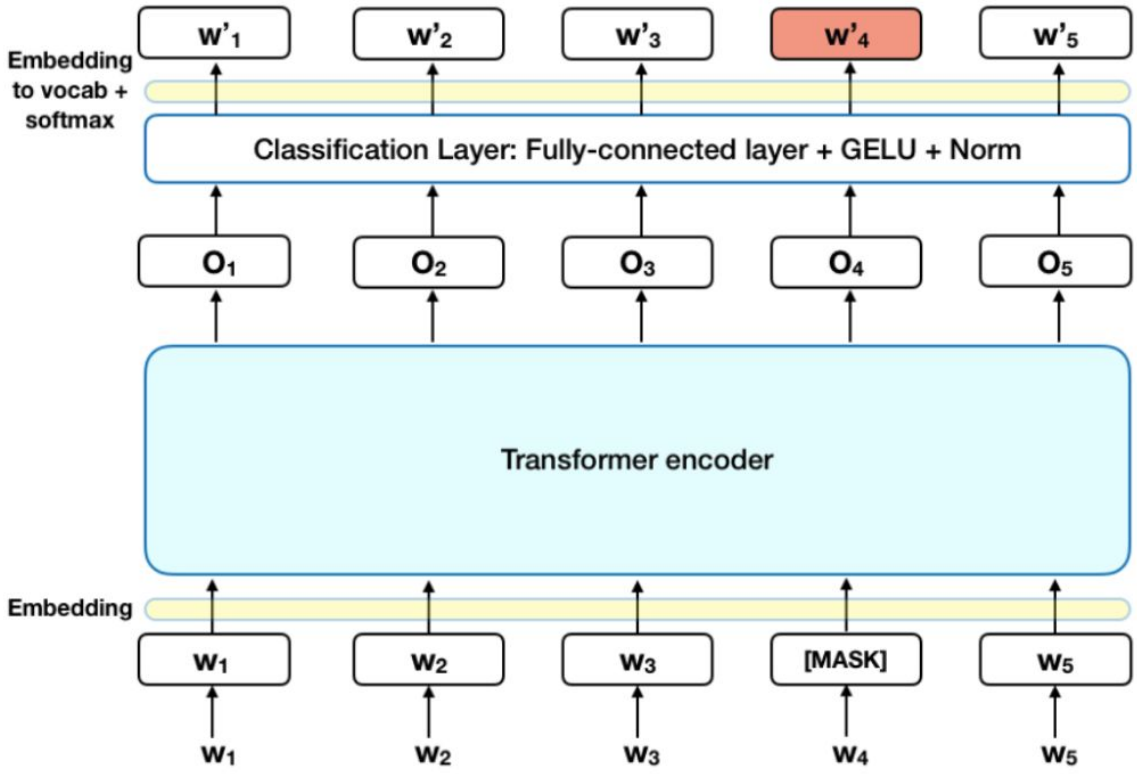
Fig. 1: The architecture of BIMPM

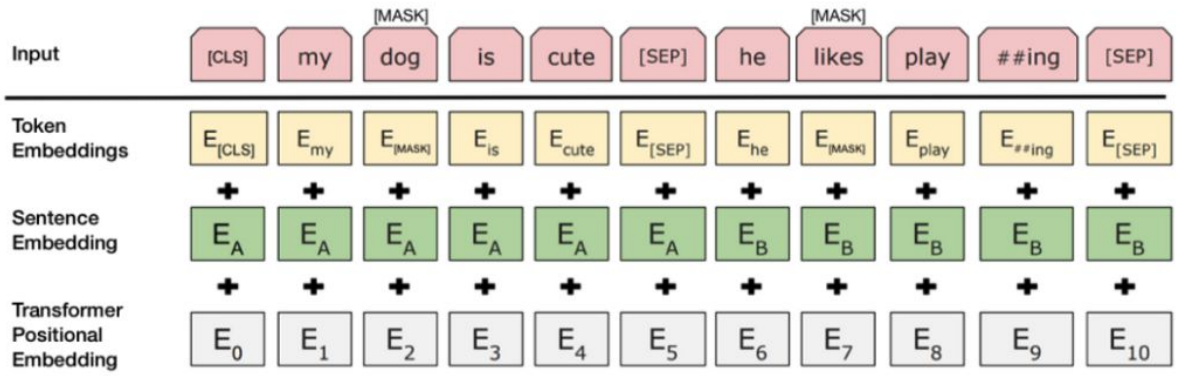
Transformer model(Baseline+):

We used a hugging face transformers library for fine tuning our task on pretrained transformer models. All models accept one input which is given by concatenating the two sentences and adding classification and separation tokens appropriately.

BERT(Bidirectional Encoder Representations from Transformers)

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.
As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word) with the help of positional embeddings. It is a masked Language model with 15% of the words in each sequence replaced with a [MASK].The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence , so it takes more time when compared to bidirectional models. The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its increased context awareness. Also in training next sentence prediction is also learnt by the model.When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies.BERT and similar transformer models was easily fine tuned on our task of Semantic Textual Similarity.





XLNet:

XLNet introduces permutation language modeling, where all tokens are predicted but in random order. This is in contrast to BERT's masked language model where only the masked (15%) tokens are predicted. This helps the model to learn bidirectional relationships and therefore better handles dependencies and relations between words. In addition, Transformer XL was used as the base architecture, which showed good performance even in the absence of permutation-based training. It uses improved training methodology, larger data and more computational power to achieve better than BERT prediction metrics on 20 language tasks.

RoBERTa:
RoBERTa is an optimized BERT approach RoBERTa, is a retraining of BERT with improved training methodology, 1000% more data and compute power.RoBERTa removes the Next Sentence Prediction (NSP) task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs. Larger batch-training sizes were also found to be more useful in the training procedure. Due to this, RoBERTa outperforms both BERT and XLNet on GLUE benchmark results.

The batch size was taken as 128 and max length of sequence was considered as 128. We decided max length as 128 because the maximum sum of two sentences was only 120. We used the base transformer models as it is enough. We used the ADAMW optimizer with learning rate 2e-5. We also used adaptive learning rate with function lr*0.1^(epoch/40). We also add some weight decay as regularization to the main weight matrices.

EXPERIMENT AND METHODOLOGY:

Dataset For Experimentation:

Popular datasets for the task of Semantic Textual Similarities are Stanford Natural Language Inference 1.0(SNLI) dataset, Quora Paraphrases dataset(Quora), Microsoft Research Paraphrase Corpus(MRPC) corpus for paraphrase identification, STS benchmark tasks, SICK relatedness, amongst others.
For our paper we have utilized SNLI and Quora datasets for validating our model.

SNLI
The SNLI corpus (version 1.0) is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels *entailment*, *contradiction*, and *neutral*, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE). After some data preprocessing of removing the sentences in which a label could not be assigned the distribution of the dataset is as follows.

Total pairs	Train	Dev	Test
569033	549367	9824	9842

We also maintained a balanced test dataset with distribution as follows for better hypotheses.

Entailment	**Neutral**	**Contradiction**
3368	3219	3237

Quora

The Quora Question Pairs dataset consists of over 400,000 pairs of questions on Quora. Systems must identify whether one question is a duplicate of the other. The distribution of the dataset is as follows. We can observe that this looks like an imbalanced dataset with a number of non-paraphrases.

Total Number	**Paraphrases**	**Non-Paraphrases**	**Train**	**Dev**	**Test**
404290	149263	255027	384290	10000	10000

But we maintained a balanced testing dataset with 5000 examples of paraphrases and non-paraphrases in the testing dataset.

Note that both the dataset are different with respect to their task they are used for and are important tasks for evaluating Semantic Textual Similarity also known as Natural Language Sentence Matching(NLSM) tasks.

Procedure:

We implemented 4 models for identifying semantic textual similarity and validated it over the two datasets. The first model Bilateral Multi-Perspective Matching for Natural Language Sentences(BIMPM) with only full matching serves as the baseline model. We then fine tuned our dataset on pretrained state-of-the-art transformer models(BERT, XLNet, RoBERTa) to compare it with our baseline model as baseline+. Brief descriptions of the models are given in the Architecture and Model Overview Section.

EVALUATION:

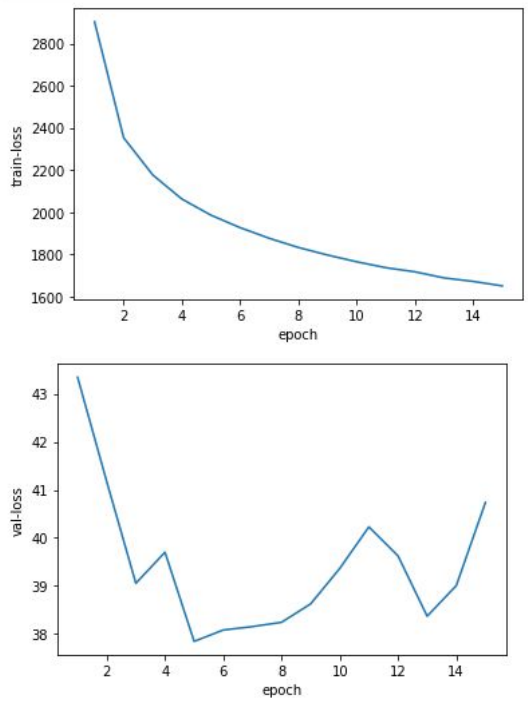
For evaluating the effectiveness of our proposed model, we use the standard recall, precision, and F1 measures(maco averaged) along with accuracy. We use Macro-averaged as given in Equation. The scores are first computed for the binary decisions for each individual category and then are averaged over categories for multi category evaluation(SNLI). Here Px, Rx, F1x refers to precision, recall and F1 score respectively of class x.

$$F1_x = 2\frac{P_x R_x}{P_x + R_x}; \qquad \mathcal{F}_1 = \frac{1}{n}\sum_x F1_x$$

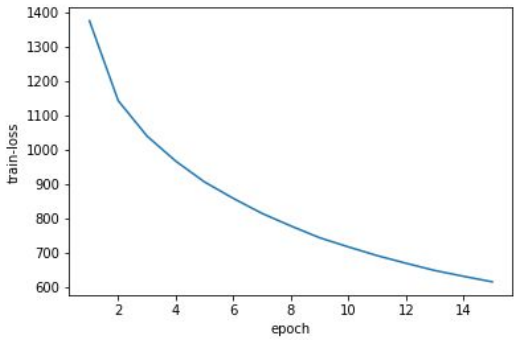
This metric is significantly more robust towards the error type distribution as compared to the other variants of the macro-averaged
We also used a confusion matrix to analyse the performance of the classification model. It helped us identify some patterns in the errors generated by the model.

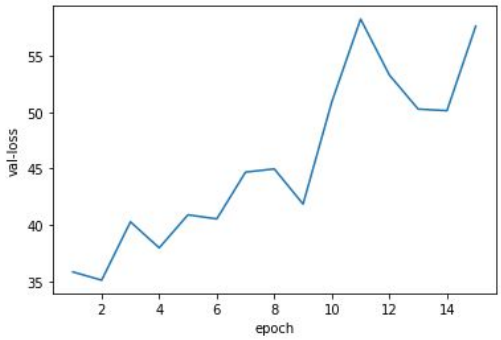
We also observed a decreasing trend in train loss over each epoch of the BIMPM model for both the datasets which validates the model is correctly implemented and the trend of validation loss tells us the point where the model begins to overfit.

BIMPM SNLI Train-Loss Curves



BIMPM Quora Train-Loss Curves





RESULTS:

Model	Dataset	Accuracy	F1_score (macro)	Recall	Precision
BIMPM	SNLI	81.1	81.6	80.9	81.0
BIMPM	Quora	80.7	80.5	80.7	82.1
XLNet	SNLI	90.869	90.869	90.857	90.888
XLNet	Quora	90.680	90.680	90.680	90.684
BERT	SNLI	90.147	90.133	90.129	90.138
BERT	Quora	89.550	89.547	89.550	89.594
RoBertA	SNLI	91.358	91.292	91.311	91.360
RoBertA	Quora	91.620	91.620	91.620	91.622

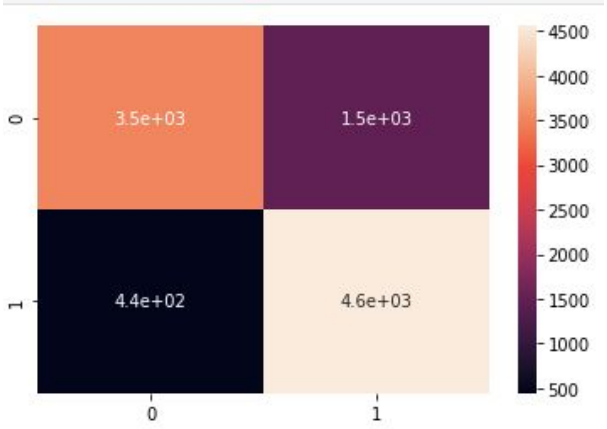
We achieved consistent results on both the datasets. Our baseline model was able to reach accuracy of ~81% on both datasets when compared to ~88% on the original implementation because we only implemented one(as proposed) matching layer(full) compared to their matching layers(full, maxpool, attentive and max attentive). The fine tuned contextual models as expected performed way better than our baseline model with a difference of ~10%

across all evaluation metrics. Comparing the transformer models we observe that RoBERTa is the best among the three followed by XLNet and last is BERT which is consistent with other studies.

Transformers(used in Baseline+) show better results(improves accuracy by 10%) than Bi-LSTMs(used in BIMPM) as they are non-sequential and uses self-attention with positional embeddings taking care of recurrence improving the longer dependencies and enabling parallel computing.

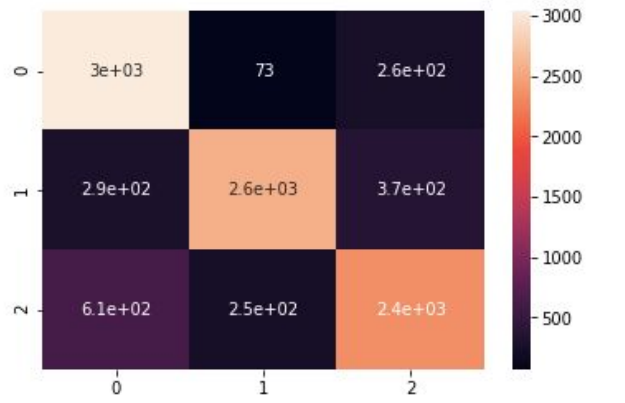
BIMPM:

Quora



0 - non-paraphrase 1 - paraphrase

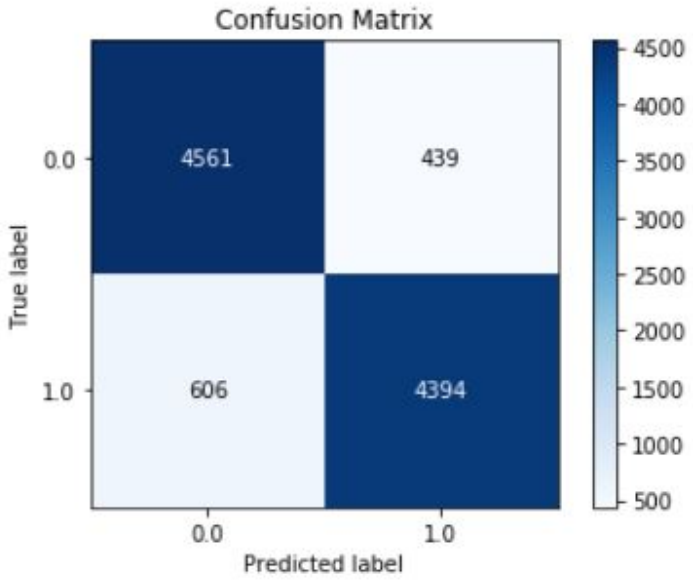
SNLI Dataset



0 - entailment 1 - contradiction 2 - neutral

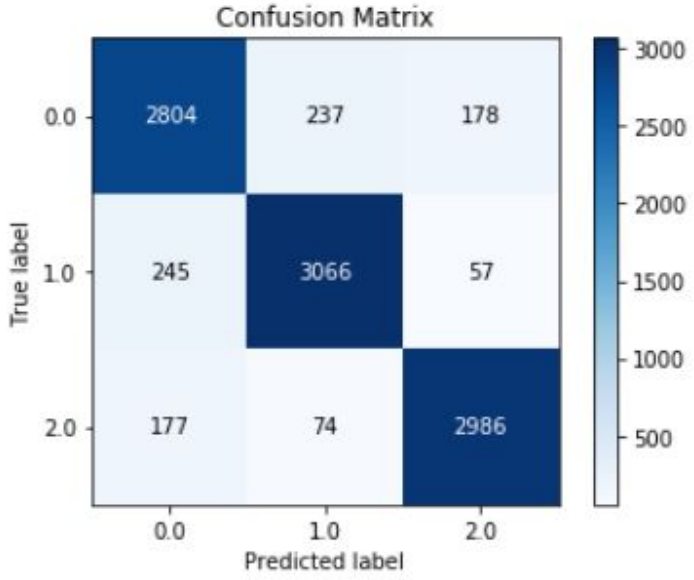
BERT

Quora



0 - non-paraphrase 1 - paraphrase

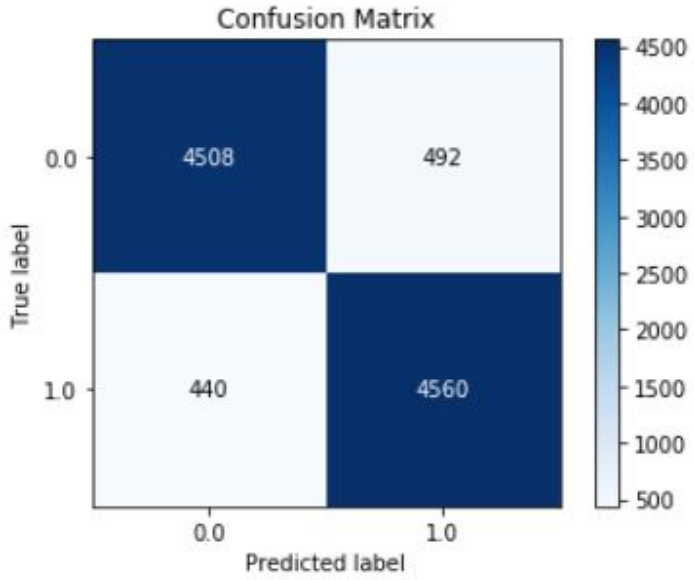
SNLI



0 - neutral 1 - entailment 2 - contradiction

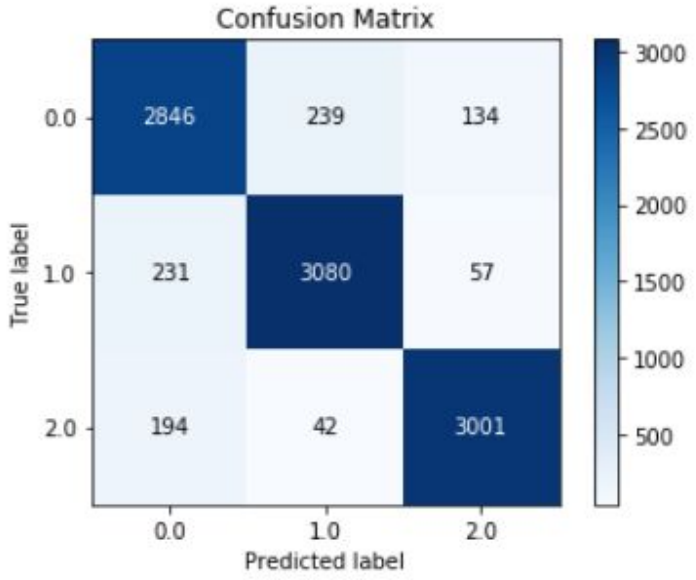
XLNET

Quora



0 - non-paraphrase 1 - paraphrase

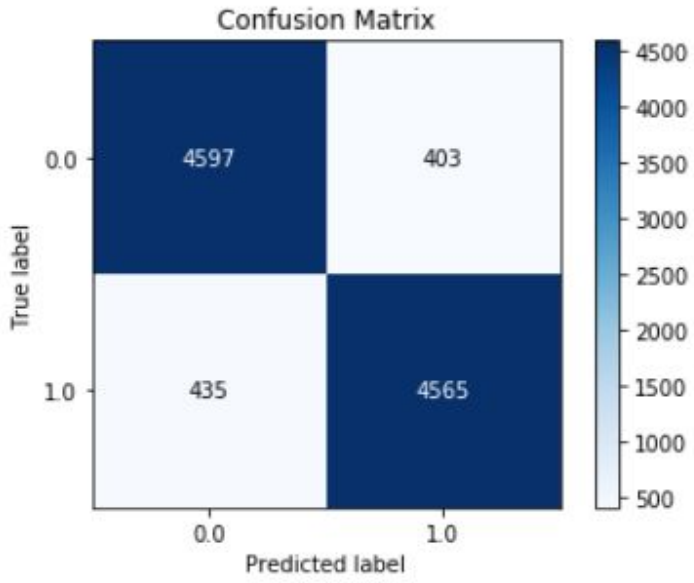
SNLI



0 - neutral 1 - entailment 2 - contradiction

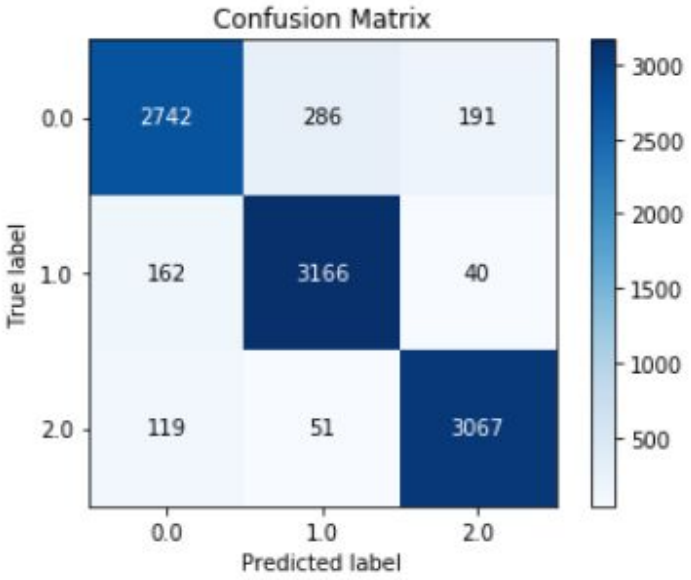
RoBERTa

Quora



0 - non-paraphrase 1 - paraphrase

SNLI



0 - neutral 1 - entailment 2 - contradiction

Qualitative Analysis:

- **Quora dataset**
  - The task for this dataset was to predict whether the pair of questions were paraphrases of each other or not.

- Out of 5000 non-paraphrases pairs about 30% (1479) of the pairs were predicted as paraphrases in the case of the BIMPM model but were predicted correctly for baseline+ models. Few of the examples are shown here:
  - How much revenue does it make ? q2: how much revenue does freeciv.net make ?
  - q1: what is the war on drugs ? q2: should we end the war on drugs ?
  - q1: difference between sociology and psychology ? q2: what is the difference between sociology and science
- The above examples are the one where the question pairs become non-paraphrases of each other because of a single word inclusion.
  - Transformer models were able to correctly identify because they are better aware of the context and multi-head attention and positional embeddings both provide information about the relationship between different words. Non-sequential helps it to not to forget the information
- Few examples of question pairs that were actually paraphrases of each other but were actually predicted as non-phrases are:
  - q1: how does one get uk-citizenship ? q2: how do i get uk citizenship ?
  - q1: what do you mean by science ? q2: what does science mean to you ?
  - q1: which is best iit in india ? q2: which is the best among iits ?
- The above examples were identified correctly by the transformer architecture?
- XLNet was able to capture longer dependencies when compared to BERT.
  - Label: 1 Pred: 0.0 Q1: Will I have an arranged or love marriage ? Q2:Will I have an arranged or love marriage ? I am Male , born 6th December , 1989 at 2:26 AM in Bicholim , Goa , India .
  - Adding a continuity in a question confuses BERT in determining it as a paraphrase.

- **SNLI dataset**
  - The BIMPM model found difficulty in identifying contradictions whereas Baseline+ improved it significantly.

  - Baseline+ models along with BIMPM model found difficulty in differentiating entailment and neutral sentences.
    - Label: neutral Pred: entailment
      - Q1: The boy in pajama pants jumps off the sofa. Q2:The boy is bouncing on the sofa
    - Label: entailment Pred: neutral

- Q1: Man jumping over a rusty fence on a blue bicycle.Q2:Man doing a trick riding his bicycle.

NOTE: link for accessing the codes, models checkpoints and data is posted below.

https://iiitaphyd-my.sharepoint.com/:f:/g/personal/jashn_arora_research_iiit_ac_in/ElzXoU3kniZJnNkm7q0IB7QB4FRnizSw7nL8a3qfYwlMWQ?e=73A2Ql

REFERENCES:

[1] https://arxiv.org/pdf/1702.03814.pdf
[2]https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8
[3]https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
[4]https://scikit-plot.readthedocs.io/en/stable/Quickstart.html
[5]https://pytorch.org/docs/stable/optim.html
[6]https://ai.stackexchange.com/questions/20075/why-does-the-transformer-do-better-than-rnn-and-lstm-in-long-range-context-depen
[7]https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9