CSI5386: Natural Language Processing
Assignment 2 : Text Entailment and Semantic Relatedness

Submitted By:
Reshma Sri Challa 300071545
Raj Kumar Endla 300058021

Work Split-up:

Task	Reshma Sri	Raj Kumar
Task 1	LSTM, BiLSTM	Siamese BERT-Networks
Task 2	Approach 1	Approach 2
Report	Task 1	Task 2

Loading Data:
We loaded the data from the following and converted into pandas:
train_data = "http://www.site.uottawa.ca/~diana/csi5386/A2_2020/SICK_train.txt"
test_data ="http://www.site.uottawa.ca/~diana/csi5386/A2_2020/SICK_test_annotated.txt"
validation_data ="http://www.site.uottawa.ca/~diana/csi5386/A2_2020/SICK_trial.txt"

Data Analysis:

```
Train Data Shape: (4500, 5)
Validation Data Shape: (500, 5)
Test Data Shape : (4927, 5)
```

1. **Task 1: Text entailment [30 points]**
   For this task, the target variable was entailment_judgment which is Neutral, Contradiction, Entailment. This is a Categorial variable so we used label encoder to convert into numerical values.

Data Analysis:

```
Train Data :
 NEUTRAL          2536
ENTAILMENT       1299
CONTRADICTION     665
Name: entailment_judgment, dtype: int64

Validation Data :
```

```
 NEUTRAL           282
ENTAILMENT        144
CONTRADICTION      74
Name: entailment_judgment, dtype: int64

Test Data :
 NEUTRAL           2793
ENTAILMENT        1414
CONTRADICTION      720
Name: entailment_judgment, dtype: int64
```

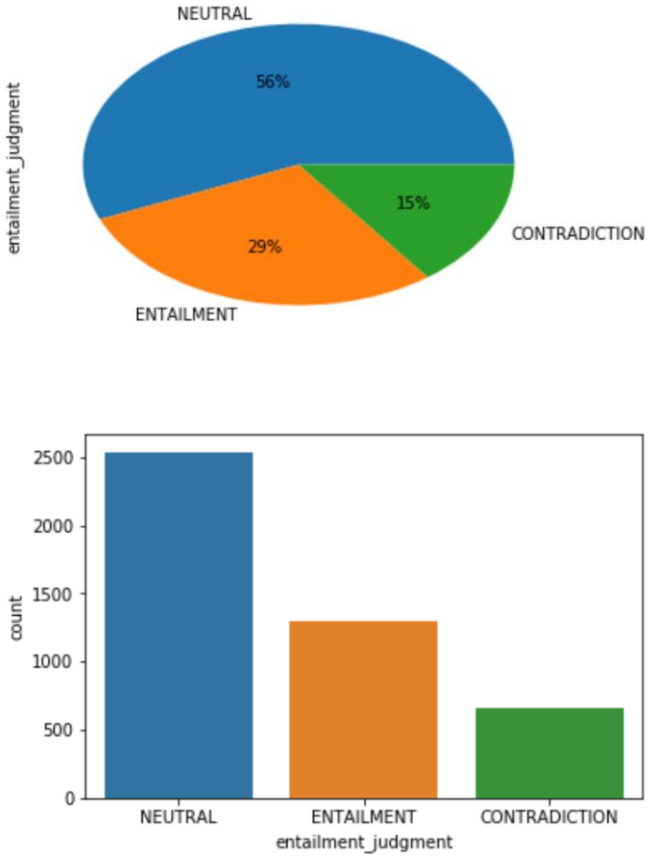

Figure 1: Text Entailment Class Distribution

The data provided is very small and highly imbalanced. Therefore we used Over - Sampling method SMOTE but the test accuracy was more without using sampling. So, we proceeded without sampling. We mapped the words in the sentences to vectors. We tried word embeddings fast text , glove and BERT but BERT had best test accuracy of 77% when compared to others.

Sentence Embeddings using Siamese BERT-Networks

Word embeddings Word2vec and Glove had poorer performance because their word embeddings didn't dynamically change based on the context of the surrounding vector. BERT is trained using

a denoising objective,where it aims to reconstruct a noisy version of a sentence back into its original version. The concept is similar to autoencoders.

SentenceBERT: The idea here is to Fine-tuning BERT to give good Sentence Embeddings. This is orders of magnitude better than having to pass in each pair of sentences through BERT. This is the current state of the art. The general idea introduced is to pass 2 sentences through BERT, in a Siamese fashion. A good diagrammatic representation is below:

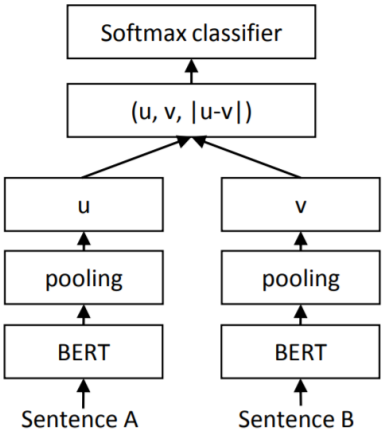

Figure 2: SBERT architecture for Classification -Task 1

The idea is simple. SBERT adds a pooling operation to the output of pretrained BERT word embeddings to derive a fixed sized sentence embedding. The pooling strategy is Mean strategy. We then *concatenate* the embeddings as follows: (u, v, ‖u-v‖), multiply by a trainable weight matrix $W \in \mathbb{R}^{3N \times K}$, where N is the sentence embedding dimension, and K is the number of labels.

Model	Epoch	Batch Size	Embedding	Sampling	Learning Rate	Optimizer	Accuracy
LSTM	10	32	Glove	No	N/A	Adam	54.2
LSTM	10	32	Glove	Yes	N/A	Adam	52.1
LSTM	10	32	FastText	No	N/A	Adam	44.6
BiLSTM	10	32	FastText	Yes	N/A	Adam	56.2
BiLSTM	10	32	FastText	No	N/A	Adam	57.46
BiLSTM	15	32	Glove	Yes	N/A	Adam	59.4
BiLSTM	15	32	Glove	No	N/A	Adam	60.0
BiLSTM	10	32	Glove	Yes	N/A	Adam	61.2
BiLSTM	10	32	Glove	No	N/A	Adam	65
BiLSTM	10	64	Glove	No	N/A	Adam	47.1
BiLSTM	10	64	Glove	No	N/A	Adam	47.1
BiLSTM	10	32	Glove	No	N/A	rmsprop	44.8
BiLSTM	10	32	Glove	No	N/A	Adagrad	37.4
BiLSTM	30	32	Glove	No	0.0001	SGD	47.3

Siamese BERT-Networks	20	450	BERT	No	0.01	SGD	60
Siamese BERT-Networks	15	300	BERT	No	0.01	Adam	72
Siamese BERT-Networks	10	450	BERT	No	0.01	Adam	73
Siamese BERT-Networks	10	500	BERT	No	0.01	Adam	66
Siamese BERT-Networks	**20**	**300**	**BERT**	**No**	**0.001**	Adam	**77**

Table 1: Results for Task 1

We can see that Siamese BERT-Networks with the hyperparameters highlighted in blue in table 1 has the best text accuracy of 77%.This is our best model.

	precision	recall	f1-score	support
CONTRADICTION	0.85	0.76	0.80	720
NEUTRAL	0.81	0.80	0.81	2793
ENTAILMENT	0.66	0.70	0.68	1414
accuracy			0.77	4927
macro avg	0.77	0.76	0.76	4927
weighted avg	0.77	0.77	0.77	4927

Figure 3: Classification report for Task 1

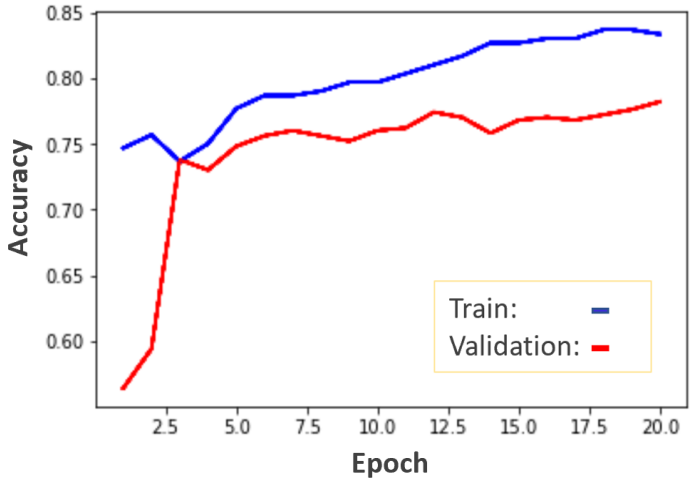

Figure 4: Accuracy for Task 1

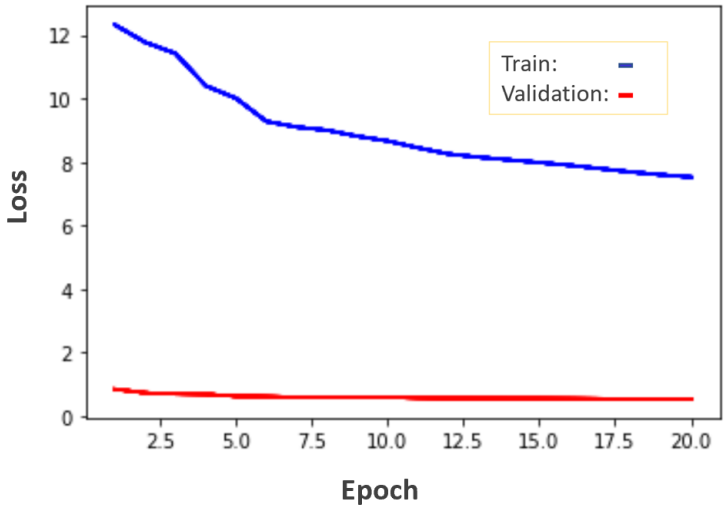

Figure 5: Loss for Task 1

Task 2: Semantic Relatedness [30 points]

We performed this task with two approaches using pooling (best results – approach 2) on pretrained word embeddings and without using pooling (approach 1).

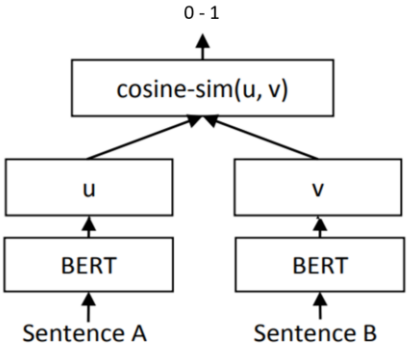

Figure 6: Approach 1 architecture to compute similarity scores

Approach 1: The architecture in Figure 6 is used for this approach Using pretrained BERT word embeddings on the sentences ,we obtain the sentence embedding by averaging the word embeddings. The cosine-similarity between the two sentence embeddings u and v is computed. We convert our relatedness_score column is scaled with MinMaxScaler in range of 0 to 1 as the cosine output is between 0 to 1. We inverse transform the output from our cosine-similarity architecture of SBERT using the scaler to get it into a range of 0 to 5.The Spearman correlation for this approch is 0.5913 , Pearson correlation is 0.6183 and MSE is 0.1074.

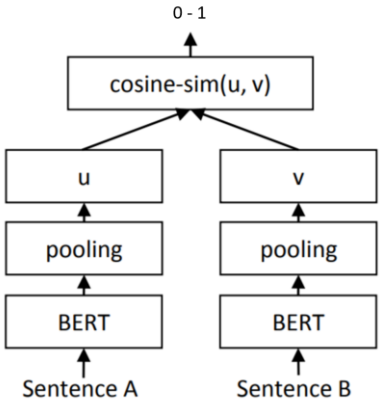

Figure 7: SBERT -Approach 2 architecture to compute similarity scores

Approach 2: The architecture in Figure 7 is also used with the regression objective function. Using pretrained BERT word embeddings on the sentences and pooling with mean-squared-error loss as the objective function. The cosine-similarity between the two sentence embeddings u and v is computed. We convert our relatedness_score column is scaled with MinMaxScaler in range of 0 to 1 as the cosine output is between 0 to 1. We inverse transform the output from our cosine-

similarity architecture of SBERT using the scaler to get it into a range of 0 to 5.**The Spearman correlation for approach task is 0.729 , Pearson correlation is 0.729 and MSE is 0.0369. Therefore we conclude that architecture in Figure 7 is the best approach for task 2 as it has higher correlation.**

Note: We have included the code for the best model for task 1 and both approaches for task 2 as mentioned above. We used Pytorch, Pytorch-pretrained-bert and sentence-transformers packages for our assignment.

References

1. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks by Nils Reimers and Iryna Gurevych ,Ubiquitous Knowledge Processing Lab (UKP-TUDA) Department of Computer Science, Technische Universitat Darmstadt- Aug 2019