NLP Project Report

Shyam Nandan Rai, Ashutosh Mishra, Elizabeth Jasmi George

Problem Statement: Reading Comprehension

Why Reading Comprehension ?

For human beings, reading comprehension is a basic task, performed daily. As early as in elementary school, we can read an article, and answer questions about its key ideas and details.

But for AI, full reading comprehension is still an elusive goal–but a necessary one if we're going to measure and achieve general intelligence AI. In practice, reading comprehension is necessary for many real-world scenarios, including customer support, recommendations, question answering, dialog and customer relationship management. It has incredible potential for situations such as helping a doctor quickly find important information amid thousands of documents, saving their time for higher-value and potentially life-saving work.

Therefore, building machines that are able to perform machine reading comprehension (MRC) is of great interest. In search applications, machine comprehension will give a precise answer rather than a URL that contains the answer somewhere within a lengthy web page. Moreover, machine comprehension models can understand specific knowledge embedded in articles that usually cover narrow and specific domains, where the search data that algorithms depend upon is sparse.

Dataset Used:

We have posed the reading comprehension problem as sentence classification task where we are using SNLI dataset [1] as mentioned in the project statement. SNLI dataset 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). This dataset is a precursor to the formal reading comprehension problem whose aim also was to focus on the textual similarity and see if one of the sentence gets one of the label after it follows the former sentence.

Our Methodology:

We have taken 1,00,000 pairs of sentences only since we have used Convolutional Neural Network and Siamese Network without GPU access. For training, we have trained both the network many times for tuning the parameters for 7 epochs which took around 5.5 hours on i7 processor. Since we wanted to compare the results with other classifiers, we took the same training corpora for Logistic Regression and also for SVM.

We have used the following techniques to achieve the classification results:

1) Logistic Regression: For logistic regression, we have used three non lexical features as described in [1].

(i) BLEU Score for entailment (ii) Absolute difference of word length (iii) Overlap between words

The accuracy with 'l1' distance metric and 'l2' distance metric is listed in the section of results.

2) Support Vector Machine: For this classifier also, we have used same set of features as described above. The different kernels were used as follows: RBF, Poly, Linear. The accuracy with different kernels obtained is listed in the results section.

3) Co-occurrence Matrix: In this method we count the number of times each word appears inside a window of a particular size around the word of interest. We calculate this count for all the words in the training data. After this we apply SVD to obtain the feature vector of words which are then passed to **SVM** and **Logistic Regression** to classify. The actual test corpus was not used for results. Instead of that, 10,000 samples were taken from the training corpora as a test set apart for the training corpus of 1,00,000 samples. The results are shown in the results section:

4) Convolutional Neural Network: We have referenced [2] for the base model of CNN which uses n-grams of different sizes, and each filter is actually an **ngram*dimension** matrix where dimension = 200 taken from the GLoVE vector embedding. We have applied dropout, and have a fully connected layer that feeds into a final classification layer, to give the final label. We have taken the activation function as ReLU as it give a higher accuracy as

compared to tanh. It is also a good choice to have ReLU as it prevents the vanishing gradient problem. The results of the CNN are mentioned in the results section.

5) Siamese Network: We have also implemented Siamese network in which we concatenated the dot product of two convolution features of filter sizes and their difference. The whole pipeline of the network is depicted in the figure mentioned in the results section.

RESULTS:

1) Logistic Regression using non lexical features:

Distance Metric	Accuracy
L1	61.15%
L2	60.75%

As can be seen, LR with L1 is better compared to L2 distance metric

2) Support Vector Machine using non lexical features:

Kernel	Accuracy
Linear	61.05%
Radial Basis	61.89%
Poly	54.33%

From the above table, SVM with RBF performed better as compared to others.

3) (a) Logistic Regression using co-occurrence matrix features:

Distance Metric	Accuracy
L1	62.53%

L2	62.43%

As can be seen, LR with L1 is better compared to L2 distance metric for this set of features as well.

(b) Support Vector Machine using using co-occurrence matrix features:

Kernel	Accuracy
Linear	62.13
Radial Basis	65.33
Poly	63.13

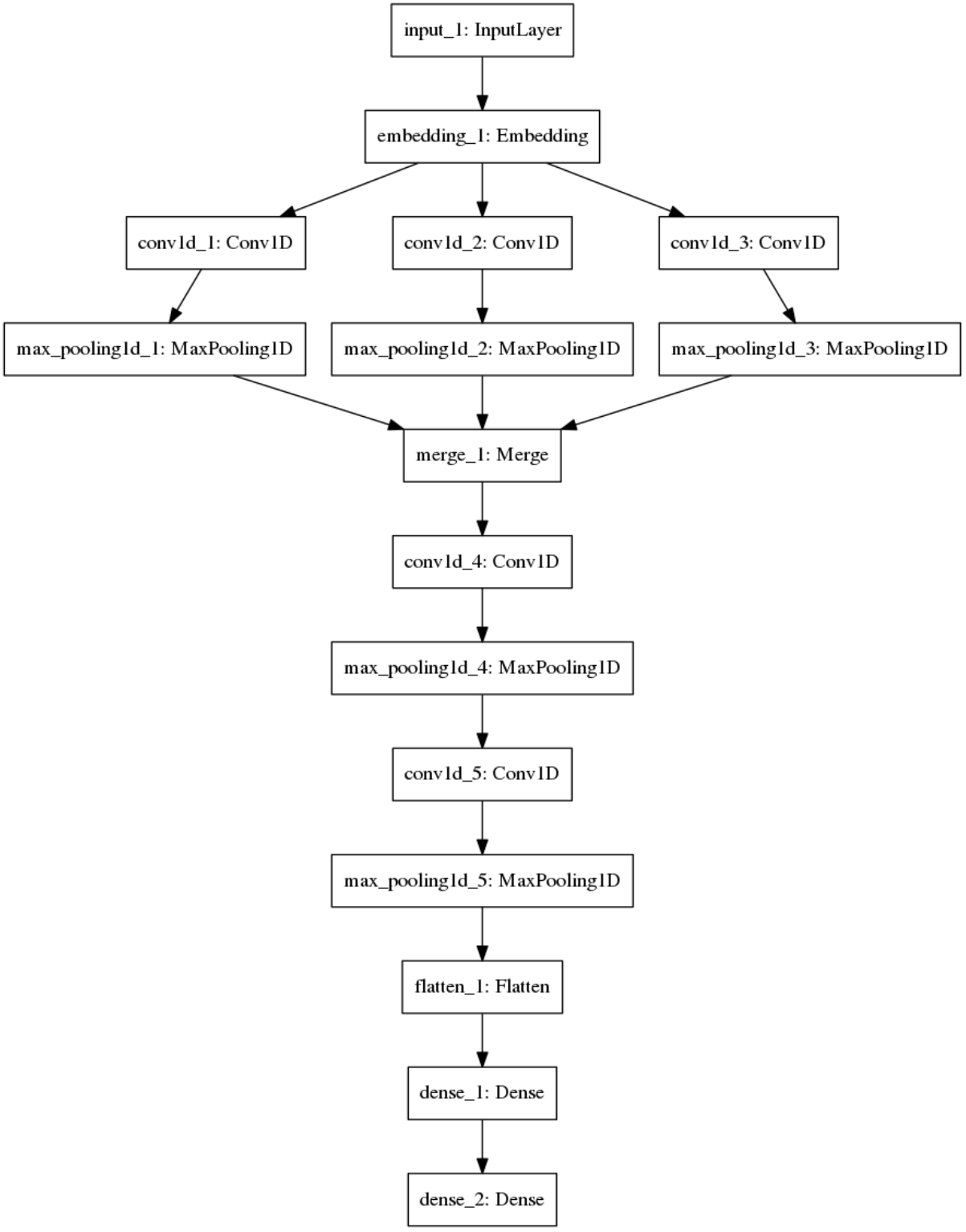
From the above table, SVM with RBF performed better as compared to others.

4) Convolutional Neural Network

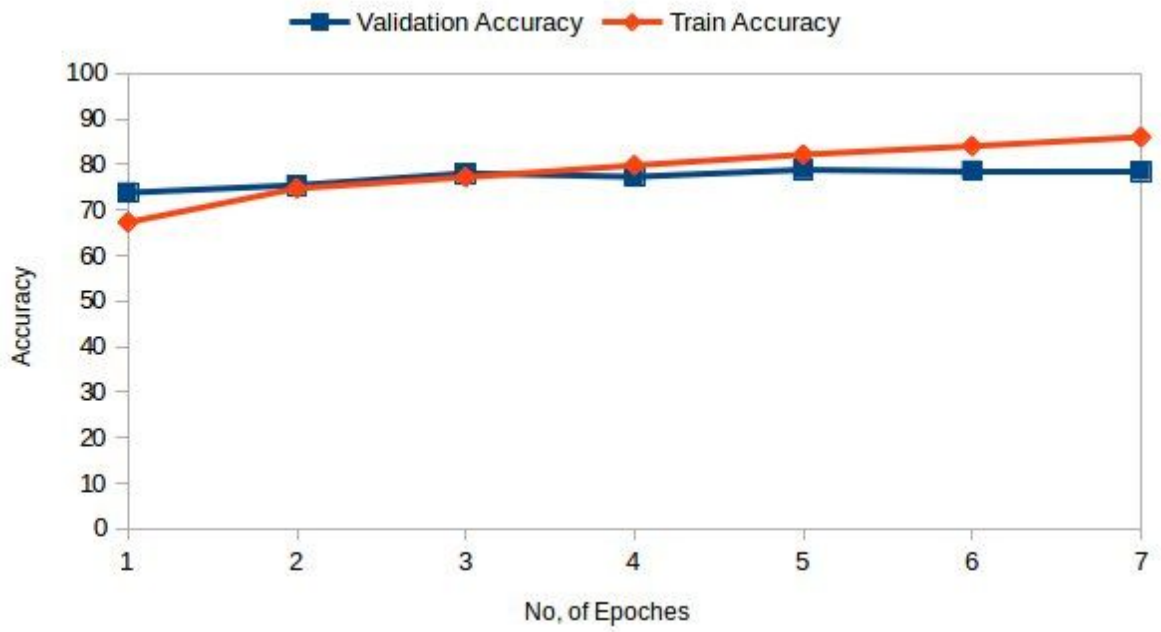
The model used for this is depicted below.

The plots for loss and accuracy are also depicted below.

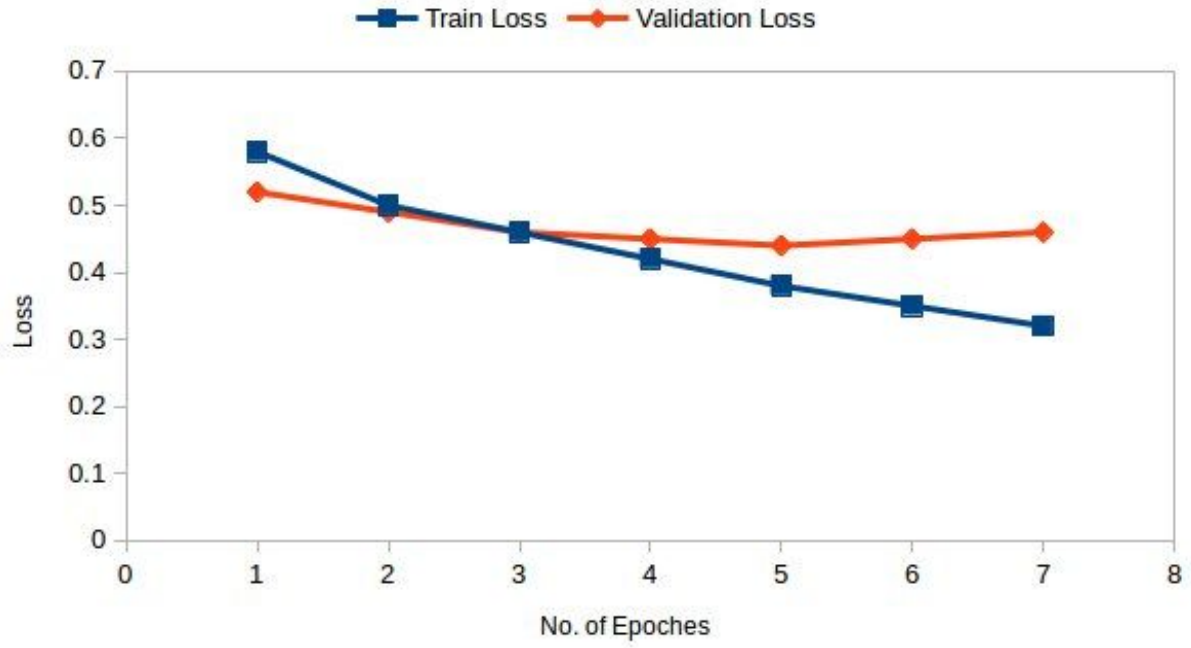
Training Accuracy in CNN: 77.25%



Model description used in normal Convolutional Neural Network



Train/Validation Accuracy vs Epochs



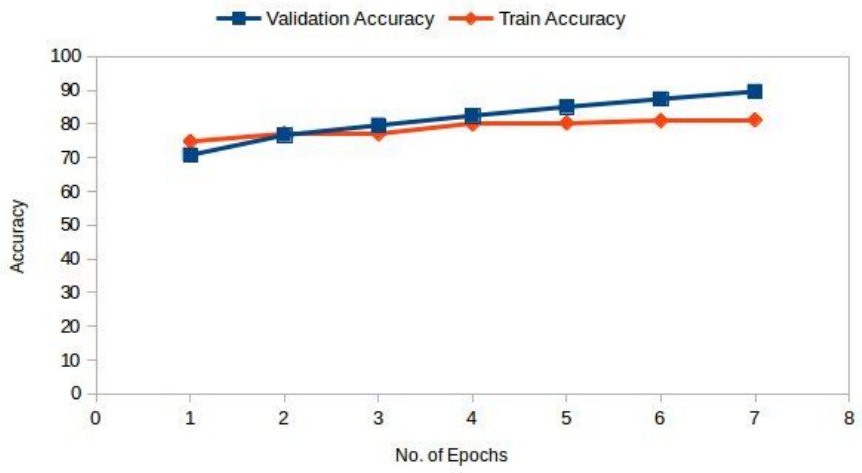
Train/Validation Error vs Epochs

5) Siamese Neural Network

The model used for this is depicted below.

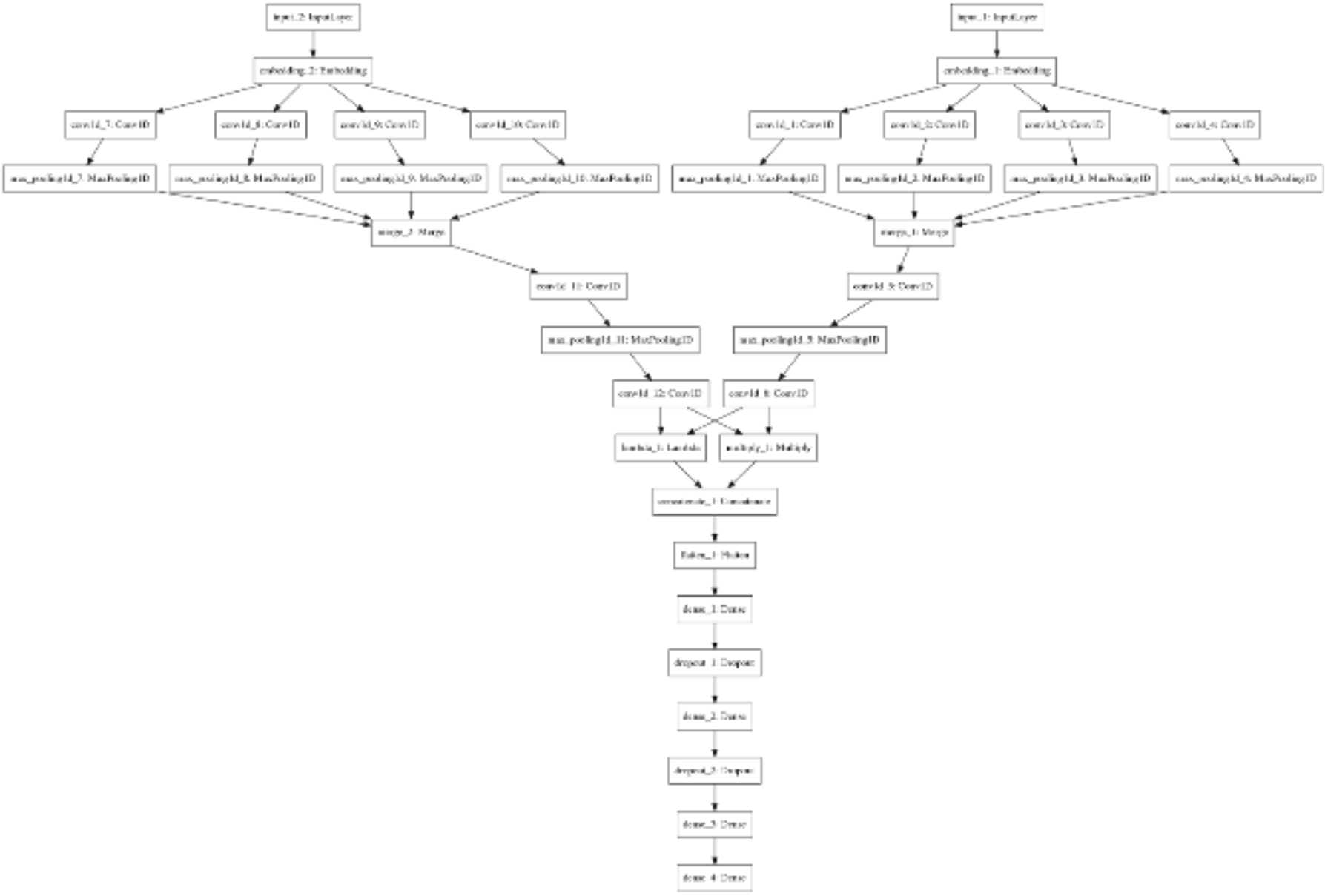
The plots for loss and accuracy are also depicted below.

Training Accuracy in Siamese network: 80.075%



Train/Validation Accuracy vs Epochs

Train/Validation Error vs Epochs


input_2: InputLayer
embedding_2: Embedding
conv1d_7: Conv1D
conv1d_8: Conv1D
conv1d_9: Conv1D
conv1d_10: Conv1D
max_pooling1d_7: MaxPooling1D
max_pooling1d_8: MaxPooling1D
max_pooling1d_9: MaxPooling1D
max_pooling1d_10: MaxPooling1D
merge_2: Merge
conv1d_11: Conv1D
max_pooling1d_11: MaxPooling1D
conv1d_12: Conv1D
lambda_1: Lambda
multiply_1: Multiply
concatenate_1: Concatenate
flatten_1: Flatten
dense_1: Dense
dropout_1: Dropout
dense_2: Dense
dropout_2: Dropout
dense_3: Dense
dense_4: Dense
input_1: InputLayer
embedding_1: Embedding
conv1d_1: Conv1D
conv1d_2: Conv1D
conv1d_3: Conv1D
conv1d_4: Conv1D
max_pooling1d_1: MaxPooling1D
max_pooling1d_2: MaxPooling1D
max_pooling1d_3: MaxPooling1D
max_pooling1d_4: MaxPooling1D
merge_1: Merge
conv1d_5: Conv1D
max_pooling1d_5: MaxPooling1D
conv1d_6: Conv1D

OBSERVATIONS:

1) Siamese network gave highest accuracy
2) We have also applied Batch Normalization in the process of improving the accuracy but actually the accuracy dropped by 1-2%.

3) SVM's were only trained on 20000 train pairs as SVM is much more suited for low amount of data.

4) We have also showed the tSNE visualization of the features of the co-occurrence matrix in the slides.

REFERENCES

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In EMNLP, 2015.

[2] Kim, Yoon. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 .