Detecting Duplicate Quora Questions

Abhishek Patria
Georgia Institute of Technology
abhishek.patria@gatech.edu

Kunaal Ahuja
Georgia Institute of Technology
kunaal@gatech.edu

Arunachalam Muthu Valliappan
Georgia Institute of Technology
avalliappan3@gatech.edu

Abstract

Our goal is to identify Quora questions having similar meaning or intent. We have built an ensemble model by combining the results from our Siamese, Attention based LSTM and BERT models. Siamese LSTMs is our simplest model and gave us close to 79.6% accuracy in the test set. The Attention based LSTM was built to have a interpretable model, which we can examine for leakage features to make it more generalizable. This model gave us a testing accuracy of 80.5%. Our BERT model gave us close to 90% accuracy in the validation set. The ensemble boosting model provides a testing accuracy of 90.8% and an F1 score of 0.878. The codes are hosted in the following repositories.

- Siamese LSTM - https://github.gatech.edu/apatria3/NLP_Quora_Questions
- Attention based LSTM - https://github.com/Arunachalam-M/AttLSTM
- BERT - https://github.gatech.edu/kahuja8/NLP_Quora_Questions
- Ensemble -https://github.com/Arunachalam-M/AttLSTM/blob/master/Ensemble.ipynb

1 Introduction and Related Work

Our goal is to identify the duplicate pairs of questions from the Quora dataset and classify them correctly. Our original plan to build 3 different models - Siamese LSTMs, Attention based LSTMs and BERT to approach the problem from different angles and generate insights, which we later compounded together with a boosting model. We have achieved our original goals without any deviations. The Siamese LSTM is simplistic and gives us a baseline, the Attention based LSTM provides interpretable models and extracts key information from the questions. The BERT model provides the best performance in the similarity identification task. We're going to use BERT as one of the models to classify the pair of questions. BERT stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

Previous work on semantic relatedness of sentences has focused on logical inference and entailment based on the Stanford Natural Language Inference Corpus 2 . The first of these papers which focused on attention methods using LSTMs was Rocktaschel et al. which introduced word-by-word attention methods with the hypothesis attending on the premise.

Classifying questions as duplicate can be quite subjective because the true meaning of a sentence is very difficult to be known with certainty. The paper on Selection Bias Explorations and Debias Methods by Zhang et al. showed that content-independent naive features called 'leakage features' are unreasonably predictive in datasets like the quora QP dataset we use here. We observed the same with some minor differences in output probabilities if the questions are input in a different order. This was our inspiration to build an attention based model where every output decision can be attributed to a simple dot product similarity between certain key words extracted in each question, which can be visualized to examine and correct.

Classifying short texts as duplicate is similar to the problem of record linkage, deduplication etc. Databases often have the same records and field

values which are not syntactically identical but refer to the same entity. This is known as record linkage and it doesn't let data mining algorithms work efficiently. (Torsten Zesch et al., 2012).

2 Methods

Data: To focus the majority of our efforts in model development and analysis, we are using a Supervised Kaggle Dataset. The data is instantly available from the given source.
Data Source: https://www.kaggle.com/c/quora-question-pairs/
For this model we split the data into train-set (363849 question pairs), validation-set (40430 pairs) and test-set(390965 question pairs). We preprocessed the data by converting it in a format of 'question1, question 2, label and a unique ID'.

Training Data(61 MB) - 404279 Question Pairs (363849 - Train, 40430 - Validation)
Test Data(49 MB) - 390965 Question Pairs
Possible Class labels -
0 - Not Duplicate (63%)
1 - Duplicate (37%)

Models:

Siamese LSTM:
We are using the Siamese LSTMs as the baseline model. Traditional Siamese LSTMs use a single sequence vector with a separator token in between. For this architecture, we had to provide 2 instances of the same pair q1 and q2 as $q1 < sep > q2$ and $q2 < sep > q1$ to ensure the output probabilities are symmetrical. We modified this architecture to have two independent vectors for the two questions and having a bilinear model which is inherently symmetric and more usable from a similarity matching perspective. This has reduced the training time and increased the accuracy of our siamese LSTM model.

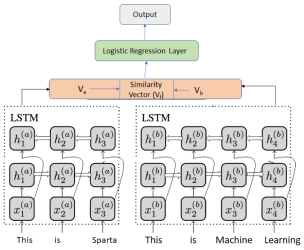

Figure 1: Architecture of the Siamese Model

The two input embeddings are first processed by separate Bi-LSTMs and the respective hidden embeddings are then fed to a Bilinear layer to get a similarity vector. This similarity vector is then trained through a logistic regression node. We use of pre-trained embeddings from FastText corpus for our word embeddings. Because our methodology depends on pre-trained embeddings for the LSTM inputs, the model will benefit from improvements and research in word embedding methods as these comprehensively capture synonymy and entity-relationships.

Attention based LSTM:
We built an architecture for Interpretable LSTMs with inspirations from the Transformer architecture and Attention Mechanism. Unlike a Transformer or our BERT model, this attention architecture is deliberately designed to identify the important parts of a sentence without looking at the other sentence to improve the generality of the summarization and Information Extraction. This architecture is not optimized for accuracy, but is more generalizable across many applications by avoiding leakage features.

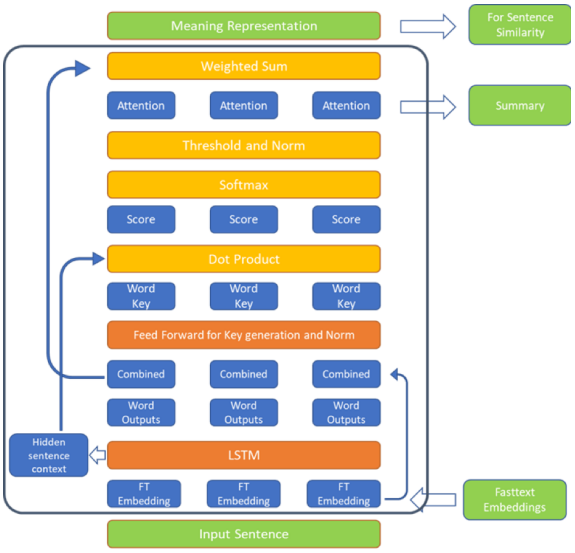

Figure 2: Custom Architecture of the Interpretable Attention Block

During the initial phases of the project, we had observed that the model was fitting to some of the leakage features leading to asymmetric predictions for the same question pair depending on which question was first and second. This led us to build a more generalized and interpretable model. This method extracts information without explicit

supervision tags. This architecture can be used to understand the functioning of the model and ensure that the accuracy obtained is because of the extraction of meaningful features and not leakage features.

The following is the complete architecture which uses the attention blocks and extracts additional features using its output to detect if a given pair of questions is duplicate or not.

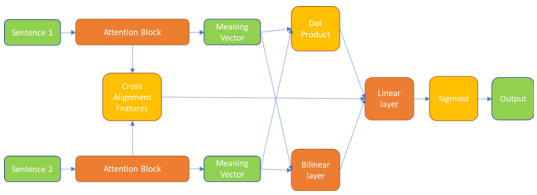

Figure 3: Architecture of the Attention based LSTM Model for Prediction

BERT:

BERT stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

On top of the standard BERT model, we added extra layers with the Softmax as the last layer. We also created a dropout layer with the hyper-parameter set to 0.1. This was done to reduce the overfitting of the model on the training data. We used Adam optimiser with L2 regularization/ weight decay to be consistent with the pre-trained BERT model. We divided the data into mini-batches and loaded them to TPU for this mode. We ran the model for 2,3 and 5 epochs using TPUs in the Google Colab environment.

We preprocessed the data by converting it in a format of ‘question1, question 2, label and a unique ID‘ and padding the data to have the max length to be 200 characters.

The following is the complete architecture of BERT which takes in the two sentences to detect if a given pair of questions is duplicate or not.

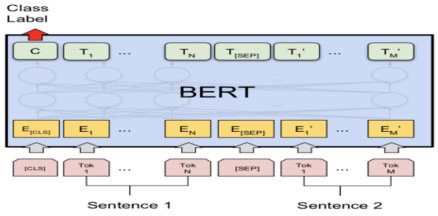

Figure 4: BERT Architecture for Classification tasks

Baseline Models:

Numerous Natural Language Processing models utilize n-gram approach to build up the set of features to be used in a model. For example, machine translation, information retrieval, sentence similarity detection etc. are some of the applications of n-gram model. The reason behind choosing n-gram with logistic regression as a baseline is that both the components have a probabilistic interpretation. Lakshay Sharma et al. [15] tested a linear model with unigram features using logistic regression with L2 regularization, controlled by a learning rate and trained with stochastic gradient descent.This model presented an accuracy of 75.4% and an F-score of 63.8% on the Quora Dataset.

Another very common baseline used for prediction problems is Support Vector Machine. SVMs are geometrically interpretable and their simplicity makes them an ideal candidate for a baseline model. Lakshay Sharma et al. [15] tested an SVM with a linear kernel and a penalty parameter on engineered features such as punctuation count, common last word, common prefix, contains ”not” et cetra. This model gave an accuracy of 75.9% with an F-score of 63.7 on the Quora Dataset.

A third robust baseline is a tree based model. Quora is reported to use a random forest with derived custom features to identify duplicate questions. Reddit uses random forests to classify content. The wide and scalable industry applicaion of tree based model for Natural Language Tasks make it a good fit for a baseline. Random Forest has been reported to give an accuracy of 75.7% and an F-score of 66.9 on the Quora Dataset.

3 Results

Experimental Setup:

LSTMs:

For both Siamese and Attention based LSTMs, we used Fasttext embeddings with the original 300 dimensional vectors. We used the following hyperparameters for the LSTMs

HIDDEN DIMENSIONS = 15
LEARNING RATE = 0.05
LSTM LAYERS = 2
DROPOUT = 0.1
OUT DIM = 30

We used 2 LSTM layers with a 0.1 dropout to model complex relationships between words in a question. For the Attention based LSTMs we made a linear Key layer, which reduces the concatenated dimension of the word embedding and the output embedding down to the Key Dimension which is the same as the hidden state dimension of 60 for each word. This enables us to take a dot product Attention for every word in the question.

The training set consisted of 363849 pairs of questions and the models were trained on Google Colab and a GTX 1660 Ti with 6GB of VRAM. Since batch processing of the entire training set would not be possible with 6GB of VRAM, we used the SGD optimizer and iterated over the questions one at a time. We chose a lower learning rate of 0.05 to ensure stability of the Stochastic Gradient Descent.

BERT:

BERT Parameters	Value
TRAIN BATCH SIZE	32
EVAL BATCH SIZE	8
PREDICT BATCH SIZE	8
LEARNING RATE	2.00E-05
NUM TRAIN EPOCHS	3
WARMUP PROPORTION	0.1
MAX SEQ LENGTH	200

Result Comparison:
Siamese LSTM:
The Siamese LSTM was able to achieve 79% of accuracy after training for 7 epochs. The motivation behind choosing this model was its minimalistic design and its simplicity of setting up. As we are using a logistic layer for the final output, we thought it would be prudent to draw the ROC curve and check the area under the curve to get a fitness check of the model.

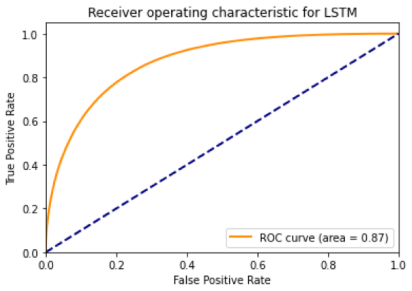

Figure 5: ROC Curve for the Siamese LSTM Model

We can see that the AUC is 0.87 which is a good figure. A threshold of 0.6 was selected because it maximized the TPR and kept the FPR to a minimum. The confusion matrix generated is as follows:

		Predicted	
		0	1
Actual	0	236551	18476
	1	70225	79038

	Precision	Recall	F1 Score
0	0.77	0.93	0.84
1	0.81	0.53	0.64

The rationale behind this selection was that Quora would like to preserve its original question first while removing the duplicates. The performance on the validation set is as follows:

		Predicted	
		0	1
Actual	0	23651	1864
	1	7003	7854

An important point to note is that the performance metrics remain nearly the same for train and validation sets. This phenomena is indicative of a robust model.

Attention Based LSTM:
The primary purpose of this model was to give insights into the interpretability of the model and some amount of information extraction. The architecture is also designed to be more generalizable across many applications by

avoiding leakage features. After 5 epochs of training, we obtained a training accuracy of 84% in the duplicate detection problem. We also obtained a 80.5% accuracy in the test set. The results in Information extraction and summarization also seem promising. The observations and results regarding the same are provided below.

Figure 6: Attention Results 1

In the above sentence ”I made the mistake of searching his social media and now I think he is too happy for me. Any words to make me feel better?”, the words ’mistake, ’searching’, ’social’, ’media’, ’words’, ’feel’, ’better’ are given attention over the threshold. So the above sentence will be summarized as ’mistake searching social media words feel better’ which is a reasonable summarization of the request with little details left out.

Figure 7: Attention Results 2

Similarly in the above example we see the sentence ”What is the most intelligent thing a kid has ever said to you?” being summarized as ”most intelligent thing kid ever said you” which is also a pretty accurate summarization of the sentence.

Aside from summarization, we can also see how the model identifies duplicates by assigning attention in the following example.

Figure 8: Attention Results 3

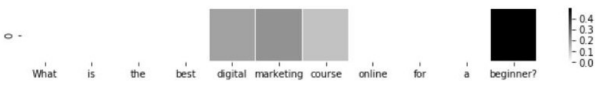

Figure 9: Attention Results 4

This was a duplicate question pair. The model predicted it correctly assigning it a 78% chance of being duplicates. But it is interesting to note that in addition to all the key words in the sentence, it assigned attention to ‘Where’ in the first question but didn’t assign attention to ‘What’ in the second question. We can see that the second question can be summarized easily as ‘Digital Marketing Course Beginners’ without the ‘What’. But ‘where’ is essential to signify the source of the digital marketing course as that was an important part of the intent of the question.

The model makes some mistakes for small questions. For eg. the question pair ”How is Israel fighting ISIS?” and ”Why is Israel Fighting ISIS?” are both summarized as ”Israel fighting ISIS” without distinguishing between How and Why. Here 4 words out of the 5 are essential to convey the complete meaning of the question. We used cross alignment to see which words do not have equivalents in the other sentence as shown below.

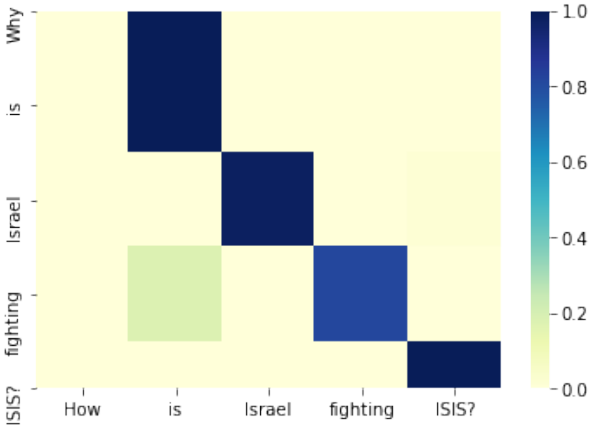

Figure 10: Cross Attention

Ideally, we expected the unaligned words to have distributed attention across other words and this can be used to quantify the cross attention between sentences using cosine similarity. But most words had the highest alignment with prepositions since the underlying fasttext embeddings are based on contextual similarity. This has reduced the importance of cross alignment features and they had a very low weight in the final linear layer determining the probability of duplicates.

For the Attention based LSTM, we generated 4 different features as below. The final linear layer used these 4 features to determine the duplicate output probability. The coefficients of these

features are also provided beside them to indicate their relative importance in identifying duplicates.

- Cosine Similarity of Attention weighted embeddings - 9.2
- Bilinear layer output to identify differences in embeddings - (-6.8)
- Cross Alignment Similarity of Question 1 with Question 2 - 0.58
- Cross Alignment Similarity of Question 2 with Question 1 - 0.5

The above weights show that the first 2 features of weighted attention and difference identification were significant in identifying the duplicates compared to the cross alignment features which were of less use.The confusion matrix of the Attention based LSTM in the test test is given below.

		Predicted	
		0	1
Actual	0	21220	4325
	1	3533	11352

This gives us an accuracy of 80.5%, Precision of 0.72, Recall of 0.76 and an F1 Score of 0.74. Accuracy of 80.5% is better compared to the Unigram baseline models of Logistic Regression (75.4%), SVM (75.9%) and Random Forests (75.9%). Our F Score of 0.74 is also better than the Unigram baseline models of Logistic Regression (0.638), SVM (0.637) and Random Forests (0.669). In addition to these benchmarks, the attention based model is also more interpretable, resistant to leakage features and has a better generalization performance. The ROC curve of the same is given below with an AUC of 0.88.

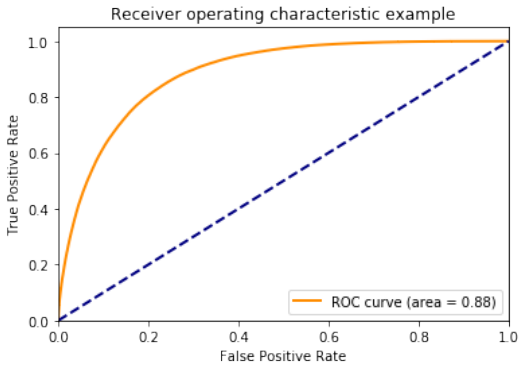

Figure 11: ROC Curve for the Attention based LSTM Model

BERT:

The BERT model was able to achieve 97% of accuracy with 5 epochs. This provides the state of the art results.
We've the following ROC curve for the BERT model.

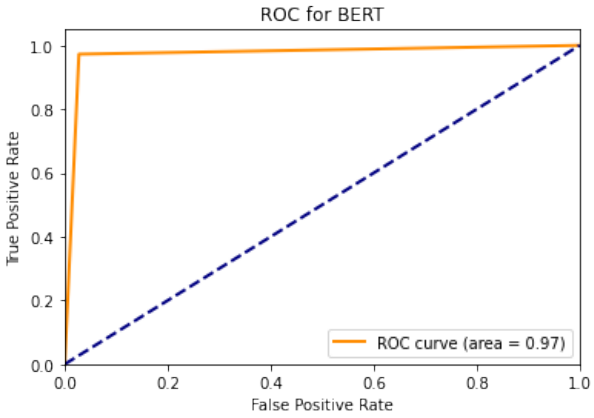

Figure 12: ROC curve for BERT

We experimented with various combinations of hyper-parameters before finalizing the combination below.
The preliminary results of BERT model are displayed below.

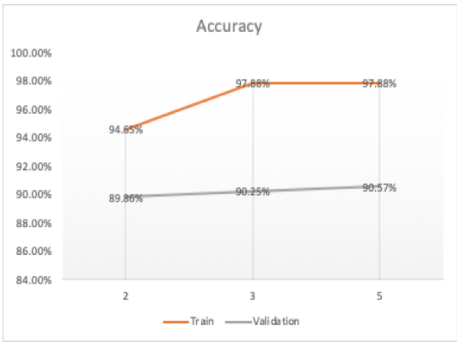

Figure 13: Train and Validation Accuracy of BERT

Training Set:

	Epoch		
Metric	2	3	5
auc	0.946492	0.978767	0.978767
eval_accuracy	0.946766	0.977842	0.977842
eval_loss	0.234789	0.104277	0.104277
f1_score	0.929171	0.970367	0.970367
global_step	22740	56851	56851
loss	0.237231	0.107138	0.107138
precision	0.913448	0.958717	0.958717

We observe that training beyond 3 epochs doesn't improve the results and may lead to overfitting. To overcome overfitting, we used the dropout parameter.
Here is the confusion matrix for the BERT model on the train dataset.

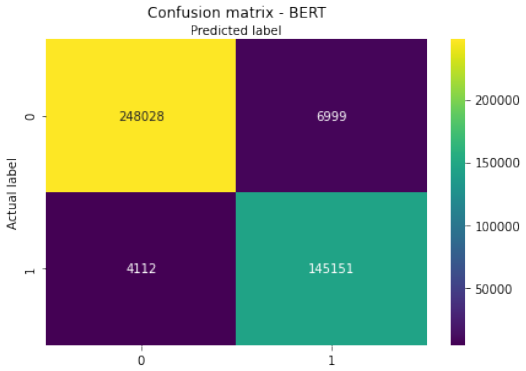

Figure 14: Confusion Matrix

Validation Set:
We then predicted the labels on the validation set. We are able to achieve an accuracy of 90.56% and F1 score of 87.81 with 5 epochs.

	Epoch		
Metric	2	3	5
auc	0.898579	0.902529	0.90569
eval_accuracy	0.902533	0.906071	0.908594
eval_loss	0.454646	0.448919	0.497576
f1_score	0.869683	0.8745	0.878129
global_step	22740	34110	56851
loss	0.47838	0.465976	0.497576
precision	0.856203	0.860367	0.862176

Here is the confusion matrix for the BERT model on the validation dataset.

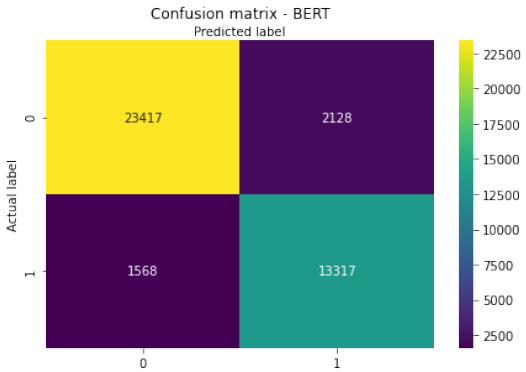

Figure 15: Confusion Matrix

Ensemble:

We used boosting methods like XGBoost and AdaBoost to combine the predictions of our 3 individual models and provide better and more robust predictions. A short summary of the performance of our individual models and ensemble is provided below.

Training Summary:

Model	Accuracy	Precision	Recall	F-Score
Attention	0.805877	0.726591	0.760296	0.743061
BERT	0.972517	0.953999	0.972451	0.963137
Siamese	0.795686	0.757902	0.656218	0.703404
Ensemble_XG	0.971728	0.956041	0.967929	0.961948
Ensemble_Ada	0.972631	0.955162	0.971473	0.963248

Figure 16: Training Statistics

Test Summary:

Model	Accuracy	Precision	Recall	F-Score
Attention	0.805639	0.724118	0.762647	0.742883
BERT	0.908583	0.862221	0.894659	0.87814
Siamese	0.796414	0.757707	0.657172	0.703868
Ensemble_XG	0.908855	0.871649	0.882365	0.876974
Ensemble_Ada	0.908781	0.864415	0.892173	0.878075

Figure 17: Testing Statistics

From the above models it is clear that BERT is much better in the duplicate detection problem compared to both LSTMs. This make the performance of the ensemble model almost identical to our BERT model. Thus we can conclude that the value addition provided by the ensemble model is not much in comparison to using just BERT from an accuracy and performance point of view.

Work Division:
Siamese LSTMs - Abhishek Patria
Attention based LSTMs - Arunachalam MV
BERT - Kunaal Ahuja
Ensemble - Team

4 Conclusion

For the attention based model, we can conclude that the attention mechanism in LSTMs is interpretable and can be used for summarization purposes. Since any question can be used

individually to compute the attention and the attention weighted embeddings, these vectors can be stored in a database and be used for simultaneous search for duplicates instead of combining the questions in pairs of 2 and identifying duplicates as used in other methods. The single head attention worked well and had a high predictive power but the cosine based cross alignment features did not work as intended. To further improve the model, we can try using a multi-head attention instead of cosine similarities to compute cross attention features although it would increase the time complexity of the attention mechanism to $O(n^2)$.

To conclude, we used an ensemble method to classify the pair of Quora questions as duplicate or not. Using the ensemble method we could achieve an accuracy of 97.2% and F1 score of 96.32 on the training set and accuracy of 90.87% and F1 score of 87.69 on the test set. The ensemble method was built on top of BERT, Attention LSTM and Siamese LSTM which had accuracy of 90%

To compare it with the baseline model used, the model used has outperformed the previous methods in the same domain. The baseline model had used three methods - SVM method with Unigrams which achieved an accuracy of 75.9 and F1 score of 63.7, second method - Random Forest which achieved an accuracy of 75.7% and F1 score of 66.9 and the final model being Logistic Regression with Unigrams which achieved an accuracy of 75.4% and F1 score of 63.8. The better results can be attributed to tuning of the hyperparameters and using state of the art models like BERT for classification.

5 Code Repository

Siamese LSTMs - https://github.gatech.edu/apatria3/NLP_Quora_Questions

Attention based LSTMs - https://github.com/Arunachalam-M/AttLSTM

BERT - https://github.gatech.edu/kahuja8/NLP_Quora_Questions

Ensemble -https://github.com/Arunachalam-M/AttLSTM/blob/master/Ensemble.ipynb

6 References

[1] Ankur Parikh, Oscar Tackstr om, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In Proceedings of EMNLP

[2] Viswanathan S., Damodaran N., Simon A., George A., Anand Kumar M., Soman K.P. (2019) Detection of Duplicates in Quora and Twitter Corpus.

[3] In: Peter J., Alavi A., Javadi B. (eds) Advances in Big Data and Cloud Computing. Advances in Intelligent Systems and Computing, vol 750. Springer, Singapore

[4] Mueller, Jonas, and Aditya Thyagarajan. ”Siamese Recurrent Architectures for Learning Sentence Similarity.” AAAI. 2016.

[5] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, Diana Inkpen. ”Enhanced LSTM for Natural Language Inference” - ACL 2017

[6] Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, Manaal Faruqui. ”Attention Interpretability Across NLP Tasks” - Computation and Language

[7] Khalil Mrini, Franck Dernoncourt, Trung Bui, Walter Chang, Ndapa Nakashole. Rethinking Self-Attention: An Interpretable Self-Attentive Encoder-Decoder Parser. - Computation and Language

[8] Guanhua Zhang, Bing Bai, Jian Liang, Kun Bai, Shiyu Chang, Mo Yu, Conghui Zhu, Tiejun Zhao.Selection Bias Explorations and Debias Methods for Natural Language Sentence Matching Datasets. ACL 2019

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need .

[10] Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. UKP-TUDA

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee,

Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[12] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, Iryna Gurevych. Classification and Clustering of Arguments with Contextualized Word Embeddings. ACL

[13] Sofia Serrano, Noah A. Smith. ”Is Attention Interpretable?”. ACL 2019

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach

[15] Lakshay Sharma, Laura Graesser, Nikita Nangia, Utku Evci. Natural Language Understanding with the Quora Question Pairs Dataset