Exploring Deep Learning in Semantic Question Matching

Ashwin Dhakal [1] Arpan Poudel [2] Sagar Pandey [3] Sagar Gaire [4] Hari Prasad Baral [5]

[1,2,3,4,5] Department of Electronics and Computer Engineering,
Tribhuvan University, Institute of Engineering, Paschimanchal Campus,
Lamachour-16, Pokhara, Nepal

Email: [1] ashwin@wrc.edu.np [2] arpanpoudel@gmail.com [3] 2052sagar@gmail.com [4] gsagarm5@gmail.com [5] haripbaral@wrc.edu.np

Abstract

Question duplication is the major problem encountered by Q&A forums like Quora, Stack-overflow, Reddit, etc. Answers get fragmented across different versions of the same question due to the redundancy of questions in these forums. Eventually, this results in lack of a sensible search, answer fatigue, segregation of information and the paucity of response to the questioners. The duplicate questions can be detected using Machine Learning and Natural Language Processing. Dataset of more than 400,000 questions pairs provided by Quora are pre-processed through tokenization, lemmatization and removal of stop words. This pre-processed dataset is used for the feature extraction. Artificial Neural Network is then designed and the features hence extracted, are fit into the model. This neural network gives accuracy of 86.09%. In a nutshell, this research predicts the semantic coincidence between the question pairs extracting highly dominant features and hence, determine the probability of question being duplicate.

Keywords - Semantic matching, Question duplication, natural language processing, deep learning, Google News Vector

I. INTRODUCTION

Over 100 million people visit Quora every month [1] and more than 14 million questions have been asked so far. Therefore, there is a high chance that many people ask similar questions that may be in different forms. This is a severe issue and hence Quora published its dataset for the first time in Feb 2017 [2]. These dataset consists of 404,290 question pairs along with the is_duplicate parameter. Collection and visualization of the dataset is done before further processing of the dataset. Preprocessing of dataset consists of tokenization, stemming and removal of stop words. All the question pairs are then converted into vectors. Normalized features, fuzzywuzzy parameters, TFIDF ratio, word-share ratio, skew factors and vector distances between the pairs of question are calculated. Feature engineering involves the working on 300 dimensions; using the Google News vector, which is trained on roughly 100 billion words from Google News dataset [3]. Features are extracted from the question pairs and then neural network consisting five hidden layers is designed. Highest accuracy of 86.09% is achieved in 16 epochs.

II. RELATED WORKS

Semantic matching of sentences previously has focused on logical inference based on the Stanford Natural Language Inference Corpus. The paper by Rocktaschel et al. focused on word-by-word attention methods using LSTMs [15]. The SemEval challenge, devoted to semantic similarity, was the foremost to include a task specifically on question-question similarity [11]. Duplicate Question Detection falls under the broader task of semantic text similarity (STS), which has been the topic of the SemEval challenges since 2012 [12]. Early work to detect the similarity between sentences used manually engineered features like word overlap [13] along with traditional machine learning algorithms like Support Vector Machines [14].

Neural Network approaches have been the state-of-the-art in a wide range of NLP tasks. Siamese neural network consisting of two sub-networks joined at their outputs was proposed. [16]. Although the Siamese architecture is lightweight and easy to train, there is no effect of correlation between the parameters, which might cause information loss [17]. So, to solve the limitations of the Siamese framework, Compare-Aggregate model was proposed [18], which captures the interaction between two sentences.

Data science engineers at Quora recently released a public dataset of duplicate questions that is used to train duplicate question detection models. Research at Department of

978-1-5386-6227-4/18/$31.00 ©2018 IEEE

Computer Science, Stanford University [19] and New York University [20] was a great source of knowledge for us to dive into. Evaluation of model for extracting various features [9] was carried out which includes the concept of fuzzywuzzy and vector distances of the texts as well.

III. METHODOLOGY

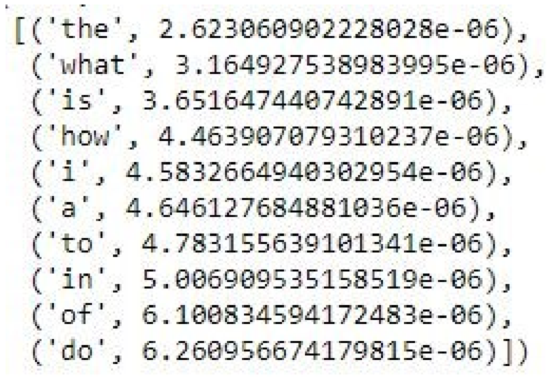
Our research is carried out in the following sections. Each step is carried out precisely with the visualization of its state.

A. Data Extraction

Dataset is provided by Quora, which contains a total number of 404,290 valid question pairs. The dataset is structured as following column labels: “id”, “qid1”, “qid2”, “question1”, “question2” and “is_duplicate”. Similarly, the test dataset contains totally 2,345,796 question pairs but without any “is_duplicate” label.

B. Dataset Preprocessing

Removing of stop words, tokenization, normalization and stemming are performed at first. Column “id” is dropped since it has no use in the prediction of duplicate question. Similarly, question mark (?) and all the stops words that act as outliers in the dataset are removed. Here is the list of words having high TFIDF score.



```
[('the', 2.623060902228028e-06),
 ('what', 3.164927538983995e-06),
 ('is', 3.651647440742891e-06),
 ('how', 4.463907079310237e-06),
 ('i', 4.583266494030295 4e-06),
 ('a', 4.646127684881036e-06),
 ('to', 4.783155639101341e-06),
 ('in', 5.006909535158519e-06),
 ('of', 6.100834594172483e-06),
 ('do', 6.260956674179815e-06)])
```

Fig 1: words having the highest TFIDF score

C. Dataset Description

Statistics of the datasets are analyzed plotting histogram, which gives the ratio of duplicate question pairs.

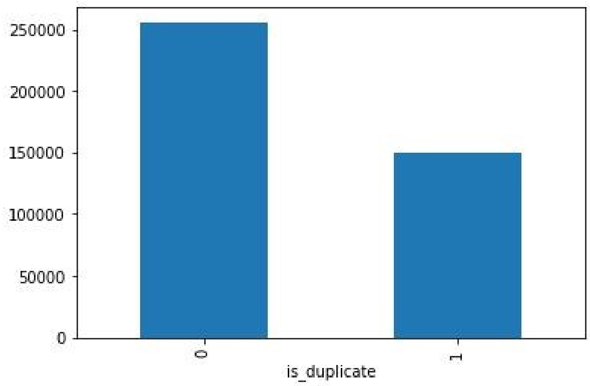


Fig 2: ratio of duplicate and non-duplicate questions in dataset

D. Relation between two questions

Relationship on word share ratio and TFIDF word share is plotted as shown below which shows that an increase in the word share and TFIDF share increases the probability of the question being duplicate.

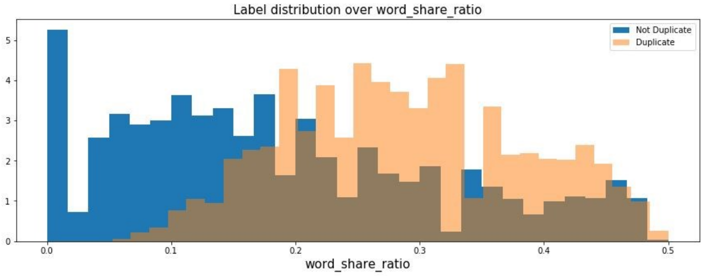


Fig 3: Visual representation of word share ratio of questions

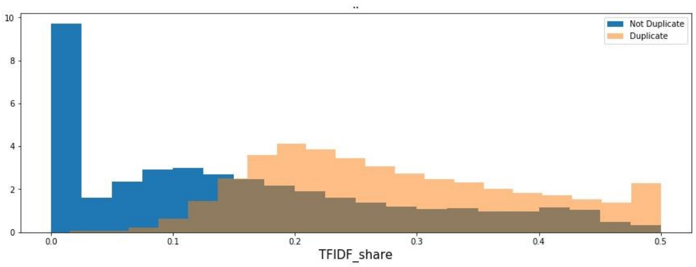


Fig 4: Visual representation of TFIDF share ratio of questions

E. Feature Extraction

Since Machine Learning models do not work directly with text, [4] we convert questions into numbers or number of vectors. This is done through Vector Space Modelling (VSM). VSM can largely be implemented via following techniques:

1) Traditional Bag of Words

A bag-of-words model is a way of extracting features from the text for the use in modeling. It is called a “bag” of words, because information about the order or structure of words in the document is discarded [5]. This is only concerned with whether the known words occur in the document, not where in the document. Here, the length of document vector is equal to number of known words.

2) *Term Frequency Inverse-Document Frequency*

TFIDF is a reflect how important a word is to a document in a collection or corpus [6]. One approach is to rescale the frequency of words by how often they appear in all the documents. Frequent words like is, a, the, etc are penalized. This approach to scoring is called Term Frequency – Inverse Document Frequency, where:

- *Term Frequency*: It is the scoring of the frequency of the word in the document.
- *Inverse Document Frequency*: It is the scoring of how rare the words are across the current documents.

$$\boldsymbol{W_{i,j}} = \boldsymbol{tf_{i,j}} * \mathbf{log}\left(\frac{\boldsymbol{N}}{\boldsymbol{df_i}}\right), \qquad (1)$$

Where, $\boldsymbol{tf_{i,j}}$ = number of occurrences of i in j
$\boldsymbol{df_i}$ = number of documents containing i
N = total number of documents

3) *Word Embedding: word2vec*

Word embedding is a collective name for a set of feature learning techniques and language modeling in Natural Language Processing. Here words and phrases are mapped into vectors of real numbers. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words [7]. Word2vec takes input a very large corpus of text and produces a vector space, which consist of several hundred dimensions, with each unique word in the corpus assigned equivalent vector in the space. Word vectors are positioned in the corresponding vector space such that words that share common contexts in the body are located in near to one another in the space.

4) *Google News vectors*

The repository provided by the Google consist of word2vec pre-trained Google News corpus word vector model which consists of 3 million 300 dimension English word vectors [8]. Google News Vector includes word vectors for a vocabulary of 3 million words and phrases that are trained on around 100 billion words from a Google News dataset. We require a high amount of memory to process this because gensim allocates a big matrix to hold all of the word vectors, and if we do the math, this is a huge matrix. Mathematically,

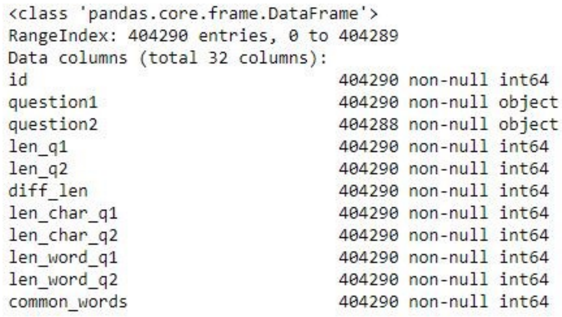
3 million words * 300 features * 4bytes/feature = ~3.35GB

F. *Feature Engineering*

Feature engineering involves extracting of information from the given dataset. Features are divided into four categories. This involves the working of basic NLTK mathematics, fuzywuzzy parameters, Word Mover Distance and vector distance.

1) *NLTK Library Groundworks*

The features: *len_q1, len_q2, diff_len, len_char_q1, len_char_q2, len_word_q1, len_word_q2* and *common_words* are derived from the training dataset for the input to the machine learning model.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 404290 entries, 0 to 404289
Data columns (total 32 columns):
id                              404290 non-null int64
question1                       404290 non-null object
question2                       404288 non-null object
len_q1                          404290 non-null int64
len_q2                          404290 non-null int64
diff_len                        404290 non-null int64
len_char_q1                     404290 non-null int64
len_char_q2                     404290 non-null int64
len_word_q1                     404290 non-null int64
len_word_q2                     404290 non-null int64
common_words                    404290 non-null int64
```

Fig 5: Basic features extracted for the training model

2) *Fuzzy wuzzy Features*

Fuzzy String Matching is sometimes known as approximate string matching. It is the process of finding strings that approximately match a given pattern. The closeness of a match is often measured in terms of edit distance, which is the number of primitive operations (insertion, deletion and substitution) necessary to convert the string into an exact match. *Fuzzy QRatio, Fuzzy WRatio, Fuzzy partial ratio, Fuzzy partial token set ratio, Fuzzy partial token sort ratio, Fuzzy token set ratio* and *Fuzzy token sort ratio* are extracted from the questions pairs.

The characteristic plot shown by fuzzy features with common words are shown below:

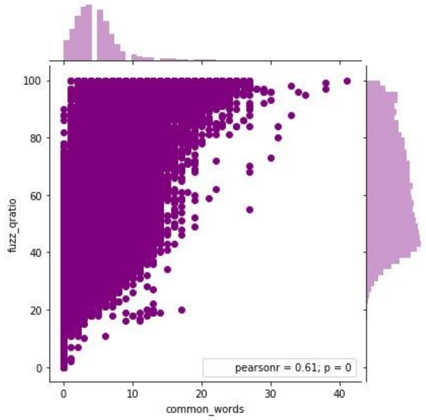


Fig 6: Characteristic plot of fuzzy quick ratio vs common words

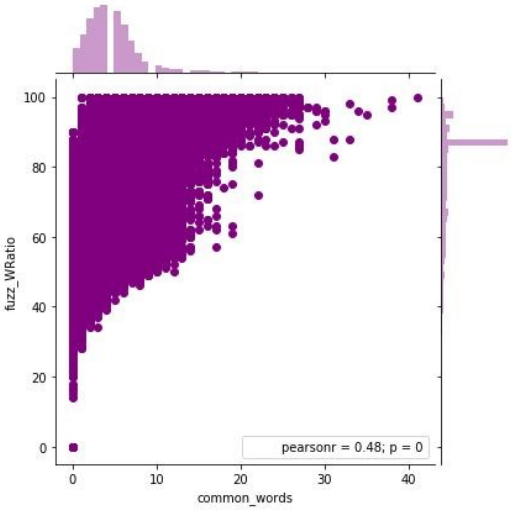


Fig 7: Characteristic plot of fuzzy weighted ratio vs common words

3) *Word Mover Distance*

WMD is a method that allows us to assess the *distance* between two texts, even when they have no words in common. It uses word2vec vector embeddings of words [10]. Normalized WMD, on the other hand, uses the Euclidean distance for calculation. To use WMD, we need some word embeddings. Gensim was very helpful for this purpose.

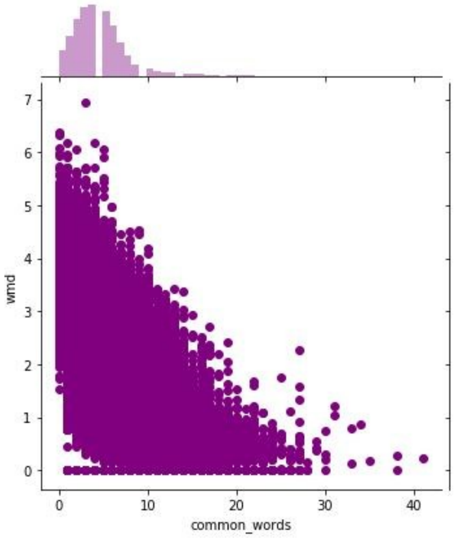


Fig 8: Characteristic plot of word mover distance vs common words

4) *Vector distances*

Various vector distances: Cosine, Cityblock, Canberra, Euclidean and Minkowski distances are measured for all the question pairs. These distances are plotted against common words which gives the similar plot.

Here is a plot of cosine distance vs common words.

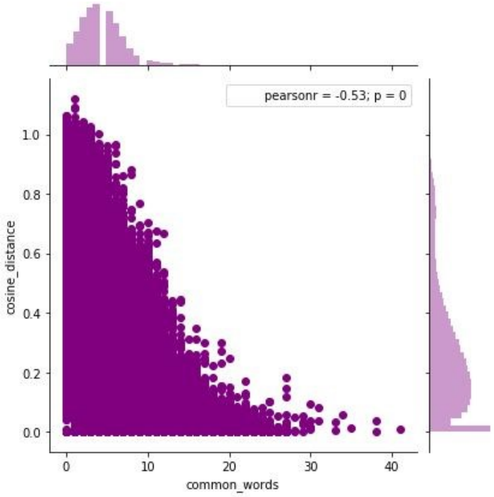


Fig 9: Characteristic plot of cosine distance vs common words

G. *Supervised Machine Learning Models*

For the validation of feature-engineered parameters, we applied different supervised machine learning algorithms and studied their accuracy and results for few parameters. The comparison can be shown as:

TABLE I: COMPARISON OF SUPERVISED MODELS FOR DUPLICATION ANALYSIS

S.N.	**Supervised Machine Learning Algorithm**	**Accuracy**
1	Random Forest	0.741
2	Logistic Regression	0.677
3	Decision Tree	0.683
4	Support Vector Machine	0.542
5	K Nearest Neighbors	0.719
6	Multinomial Naive Bayes	0.673

H. *Neural Network Design*

After completion of feature engineering, the parameters extracted are now tested to evaluate how these parameters affect the performance of our neural network. 15 best parameters are chosen through the analysis of heat-map.

Artificial Neural Network is designed consisting of 5 hidden layers. Input layer consists of 15 nodes that send information to the hidden layer.

Here is the exact artificial neural network of our model, which consist of an input layer with 15 input nodes, 5 hidden layers with 8 nodes each for precise result calculation and a single output node that results 1 or 0 i.e. duplicate or not_duplicate.

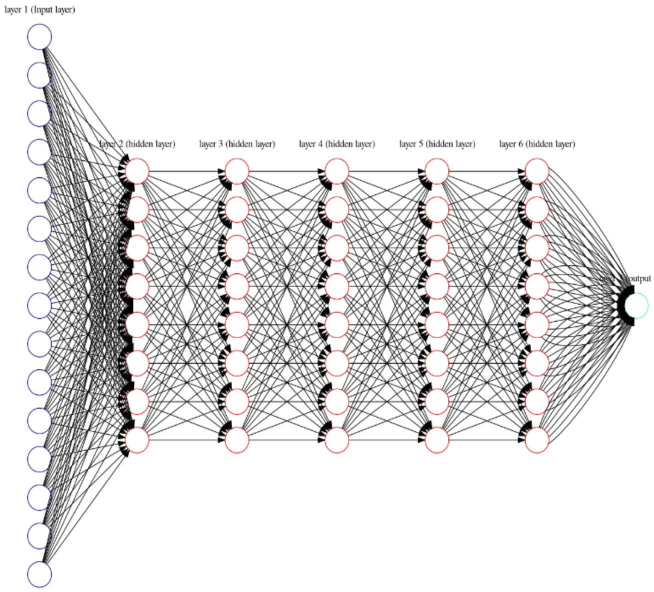


Fig 10: Our artificial neural network consisting of 15 input nodes

Here, we have used 6 ReLU activation function i.e. ***A(x) = max (0, x).*** The first one is used in the output of the input layer. Likewise, 6th ReLU is used in the output of the 4th hidden layer and the value is fed to last hidden layer. Sigmoid function is in of the form ***f(x) = 1/(1+e^-x).*** The main reason why we use sigmoid function is that our output exists between (0 to 1) i.e. question is duplicate or not_duplicate.

I. Training

We use 10% of dataset as testing dataset and remaining 90% as training dataset. This results in 40,429 testing data. Input parameter are: *diff_len, common_words, fuzz_qratio ,fuzz_WRatio,fuzz_partial_ratio,fuzz_partial_token_set_ratio, fuzz_partial_token_sort_ratio, fuzz_token_set_ratio, fuzz_token_sort_ratio,cosine_distance, cityblock_distance, canberra_distance,euclidean_distance,minkowski_distance* and *braycurtis_distance*. The batch size is 32 and 16 epochs are used for training of our model. Batch size defines the number of samples that are going to be propagated each time through the network. The training and testing accuracy is plotted as:

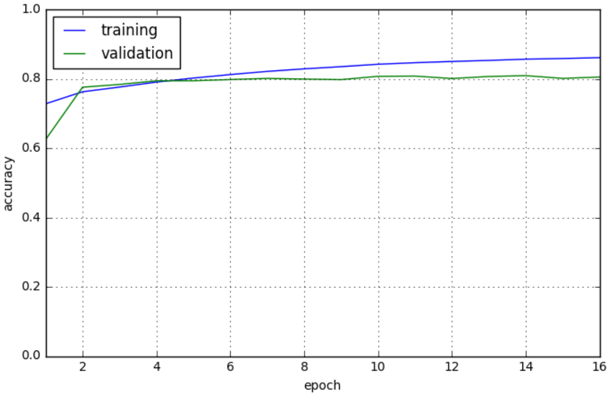


Fig 11: training and testing accuracy

The corresponding table of confusion, for the duplicate and non-duplicate, would be as shown:

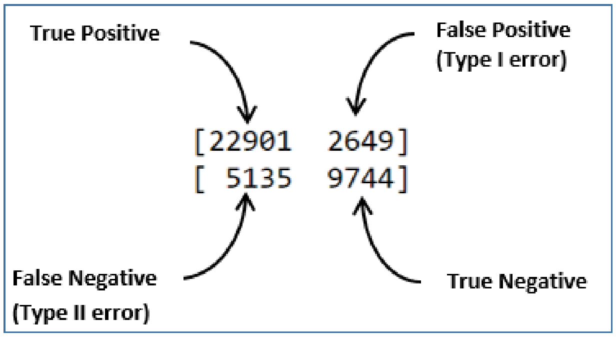


Fig 12: corresponding table of confusion

Accuracy (ACC) = (Σ True positive + Σ True negative)/ Σ Total population

We get testing accuracy as,

Testing accuracy = [True Positive + True Negative] / [Total Population]

= [22901 + 9744] / [22901+2649+5135+9744]

= 80.74649 %

IV. RESULTS

Random Forest was the best supervised machine learning algorithm among the 6 algorithms tested for this scenario. Moreover, Hyper-parameter optimization of the artificial neural network (as shown in Fig.10) resulted in a gradual increase in accuracy of the model. Finally, training accuracy of our model increased to 86.09% and validation accuracy is noted to be 80.74%.

V. DISCUSSIONS AND CONCLUSION

Hence, this research work adopts Natural Language Processing to address the problem of question duplication in Q&A forums by applying Deep Learning to classify whether question pairs are duplicates or not. Selected highly dominant features from the questions and implementation of minimal cost architecture of ANN makes it effective archetype to detect duplicate questions and eventually find high-quality answers to questions in Q&A forums.

ACKNOWLEDGMENT

This research is supported by the Departmental Research Unit (DRU), Department of Electronics and Computer Engineering, Paschimanchal Campus, Pokhara.

REFERENCES

[1] YEUNG, K. (2016, March 17). Quora now has 100 million monthly visitors, up from 80 million in January. Retrieved from venturebeat.com: https://venturebeat.com/2016/03/17/quora-now-has-100-million-monthly-visitors-up-from-80-million-in-january

[2] Lili Jiang, S. C. (n.d.). Quora. Retrieved from engineering.quora: https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning

[3] mccormickml. (2016, April 12). Retrieved from mccormickml: http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python

[4] Machine Learning Mastery. (2017, June 15). Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn

[5] Brownlee, J. (2017, October 9). A Gentle Introduction to the Bag-of-Words Model. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/gentle-introduction-bag-words-model

[6] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining". Mining of Massive Datasets (PDF). pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[7] Gilyadov, J. (2017, March 23). Word2Vec Explained. Retrieved from github.io: https://israelg99.github.io/2017-03-23-Word2Vec-Explained

[8] McCormick, C. (2016, April 12). Google's trained Word2Vec model in Python. Retrieved from mccormickml.com: http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python

[9] Thakur, A. (2017, Feb 27). *Is That a Duplicate Quora Question?* Retrieved from Linkedin: https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur

[10] Github. (2017, April 16). Retrieved from github: https://markroxor.github.io/gensim/static/notebooks/WMD_tutorial.html

[11] E. Agirre, C. Banea, D. Cer, M. Diab, A. GonzalezAgirre, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 497–511.

[12] E. Agirre, A. Gonzalez-Agirre, D. Cer, and M. Diab, "Semeval-2012 task 6: A pilot on semantic textual similarity," in Proceedings of 1st Joint Conference on Lexical and Computational Semantics, 2012, pp. 385–393.

[13] Andrei Z Broder. 1997. On the resemblance and containment of documents. In Compression and Complexity of Sequences 1997. Proceedings. IEEE, pages 21–29.

[14] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2016. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In COLING. pages 2880–2890.

[15] Tim Rocktaschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, Phil Blunsom. Reasoning about entailment with neural attention. In ICLR 2016

[16] Bromley, Jane, et al. "Signature Verification Using A "Siamese" Time Delay Neural Network." IJPRAI 7.4 (1993): 669-688.

[17] Wang, Zhiguo, Wael Hamza, and Radu Florian. "Bilateral Multi-Perspective Matching for Natural Language Sentences." arXiv preprint arXiv:1702.03814 (2017).

[18] Wang, Shuohang, and Jing Jiang. "A Compare-Aggregate Model for Matching Text Sequences." arXiv preprint arXiv:1611.01747 (2016).

[19] Addair, T. (2016, Feb 20). *Duplicate Question Pair Detection*. Retrieved from stanford.edu: https://web.stanford.edu/class/cs224n/reports/2759336.pdf

[20] Lei Guo, C. L. (2017, Jan 16). *Duplicate Quora Questions Detection*. Retrieved from semanticscholar.org: https://pdfs.semanticscholar.org/4c19/2b8f45b1e913ee7da32624cd7559eccb0890.pdf