Research Report on Detecting Duplicate Question Pairs

Using NLP and Deep Learning Techniques

Prathusha Koouri
Student ID- 013710658
Master’s in Data Analytics
San Jose State University

Contents

Abstract

Quora is a social media platform where people ask questions and connect with others who can provide answers to their questions. With millions of people using this website all around the world in search of answers for their curious questions, there is a high probability that the questions are repeated, and the duplicate questions are scattered in different pages. When those repeated/similar questions are linked together, that can make a major difference on this question answering website. For my research, (Shankar Iyer, n.d.) to differentiate Duplicate Question Pairs from non-duplicate questions.

In this paper, I would like to cite and discuss different approaches used by other people who have resolved the issue of duplicate questions in Quora Website and tried to link them together. As the Quora Questions dataset analysis is based on Natural Language Processing approach, I did a thorough analysis and feature engineering on the dataset and presented a novel approach to the above-stated problem which uses the Attention techniques of the neural network to further enhance and resolve the problem.

Introduction

Natural Language Processing (NLP) is the combination of the following three fields: Computer Science, Linguistics and Machine Learning, which is focused on enabling computers to understand and process natural language. Understanding human language is considered a difficult task due to its complexity and variety of usage. Humans can interpret each sentence in a different way by understanding the context in which it is used but Computers don’t have the same intuition of understanding natural language by the context.

NLP is benefited from the recent advances in Machine Learning, especially from Deep Learning techniques. Deep Learning has enabled us to train machines to perform things like language translation, semantic understanding, and text summarization. These have reduced a lot of manual effort to understand and process large blocks of textual data to extract insights from them.

The field of NLP is divided into three categories as follows (Donges, 2018):

Speech Recognition—The translation of speech to text.

Natural Language Understanding—The computer's ability to understand the human's language by analyzing the context.

Natural Language Generation—The generation of natural language by machine.

Understanding a sentence or a talk with correct context is an unconscious process for humans and for machines it is referred to as Semantic Analysis. Semantic Analysis is the process of understanding the meaning, interpretation, and the context of words and signs by machines. Semantic Analysis is one of the major challenges in Natural language Processing, although speech recognition is well developed, Natural Language understanding is still lacking in proficiency and is considered an AI-Hard problem. Learning semantic text similarity metric and cracking the hard problem has generated a great deal of research interest for me.

With Natural Language Understanding(NLU) machines can deduce what the speaker actually means and not just the words that they say, it enables Voice technology. NLU is all about providing computers with necessary context behind what we say, and the flexibility to understand the many variations in how we might identify things (Amazon Alexa, n.d.). When a machine understands and interprets what we speak or write with context and meaning, then the conversation with bots feel like an actual one instead of just a question-answer format. For my research the topic 'Quora duplicate question pair detection' is based on Natural language understanding as here the machine should understand the intent of the questions and their underlying meaning to club them together by understanding how every human thinks differently for framing similar questions.

Purpose

Quora is a social media platform where people ask questions and connect with others who can provide unique insights and quality answers to their questions, this also enables people to learn from experts in the industry/domain. More than 100 million people visit Quora every month, so it's no surprise that there is a higher probability of repetition of similarly worded questions. Multiple questions with the same intent can cause readers to spend their valuable time finding the correct/best answer for the question and make writers feel exhausted by answering the same question multiple times. Those questions linked together can have vast influence on Quora's user base, as readers can gain more knowledge by learning multiple answers to their question and those answers can also provide various insights to their question. As we all know every human thinks differently and can support a problem by many different aspects which can be very productive for researchers. Even the answer writers will benefit by reaching a larger readership as every answer seeker will read their page.

The ultimate goal of Quora is that there should be a single official question page for all the questions with the same intent which can provide a better experience to active users(both answer seekers and answer contributor) and offers more value to both of these groups in the future (SambitSekhar, 2017). To alleviate the ineffectiveness of having duplicate question pages, we need an automated way of detecting the pair of questions that are semantically alike. Currently, Quora uses Random Forest technique to identify duplicate questions. And my main purpose in selecting this challenging problem is to help Organize Quora website as well as the people looking for multiple insights to their questions in a better way using deep learning techniques.

Literature Review

There are several experiments conducted on this dataset released by Quora to analyze and train a model to perform correct prediction for duplicate questions and link them together. Many experiments are worth mentioning as they provide various insights about the data, and also achieved a test set prediction accuracy of 85% with their NLP logics and machine learning models.

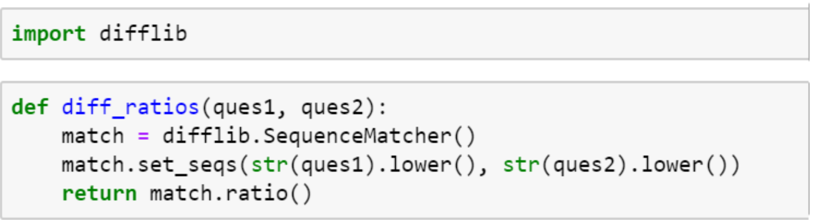
To perform machine learning on data we need to pick features from the data which are influential and helpful in making accurate predictions. For numerical datasets like House Price Prediction , Predicting Diabetes datasets, etc... we have predefined features called predictor variables and for accurate prediction we may use all or some of them depending on the correlation and other factors but for Natural Language Processing problem which deals with human language understanding and processing doesn't have pre-defined features , we need to mine/extract features from the given data.

As per previous studies, they used different features from the data according to the algorithm used to fit the model and make predictions. Here, I would like to discuss and cite some of the features considered and methods used previously for the prediction using this dataset.

Feature Engineering

Feature Engineering process involves the extraction of several character-level, word-level, sentence-level features, as well as the comparison of features in terms of the predictive power in identifying the duplicated questions. The features are categorized into 2 groups where the first group can directly interpret the English words to find the similarity and second group will convert the English words to mathematical representation and then analyze the similarity.

The **First two features** focus on direct similarly of two questions. First, the difference in the sentence length can be calculated using python SequenceMatcher and the second proportion of the words in common (Zihan Chen, Quora Question Pairs, n.d.). Both features will give us output as probabilities which can be directly used as features or can be converted to integers 0 and 1 for predicting output.



```
import difflib
```

```
def diff_ratios(ques1, ques2):
    match = difflib.SequenceMatcher()
    match.set_seqs(str(ques1).lower(), str(ques2).lower())
    return match.ratio()
```

Fig: Python code snippet for using SequenceMatcher

The **Third feature** is also derived from the direct similarity called TF-IDF(Term Frequency-Inverse Document Frequency) metric (Tuvtran, n.d.). It is used in text mining as a weighting factor for features and mainly used to reduce the effect of most frequently occurring words such as stop words. **Term frequency** is defined as the number of times the word found in the document. **Inverse document frequency** is the logarithm of the total number of documents in the corpus divided by the number of documents that contains that specific term, this division helps eliminate most frequently occurring words from documents. We multiply TF with IDF to get the importance of the word. As the stop words usually appear very frequently in all documents, so that can mitigate their importance.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Fig: Formula for calculating TF-IDF

The **Fourth feature** derived is a part of the second group and is called Word2Vec representation of words (Cohen, 2017). We use numerical representation for the normal English words. Word2Vec preserves the relationship among words by generating Word Embeddings(which are also called fixed length feature vectors) which maps the words into vector space where we can perform operations(addition, subtraction, calculate distance) on those vectors to preserve the relationship among words. It creates similar embeddings for the words which have same intent or relate to the same context, by iterating through the large corpus of text where we can see that the words taken can be interchangeable will have similar embeddings. There are 2 techniques used to generate word vectors(Embeddings) :

CBOW(Continuous Bag of Words) – We will predict the target word from the context.

Skip-gram – We will predict the context words from the target word.

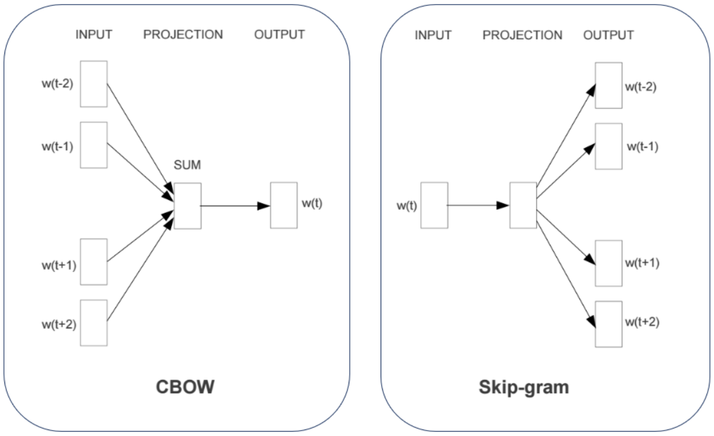


Fig: Showing CBOW and Skip-gram Language Models

We consider a simple 2-layer neural network to predict accordingly for CBOW or Skip-gram. The input and output layer will have the one-hot encoding of each word based on the corpus we selected, and the hidden layer will contain the number of neurons as the size of the final word embedding vector. After updating the weight matrix continuously by backpropagating the errors, we get a final weight matrix from input layer and then it is multiplied with the respective word’s one-hot encoded vector to generate a word embedding which is considered as word feature vector and is given as input for many Machine learning algorithms to extract the semantic relationship among different words. Instead of generating our own Word Embeddings, we can even consider pre-trained word embedding such as google Word2Vec model as features for the problem we will be addressing.

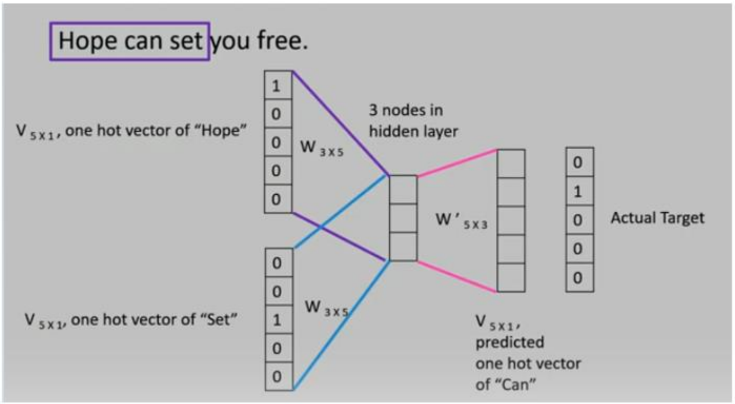
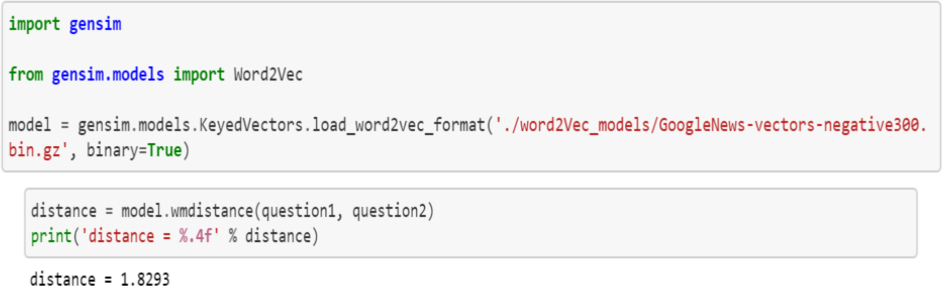

Hope can set you free.
V 5x1, one hot vector of “Hope”
1
0
0
0
0
W 3x5
3 nodes in hidden layer
W ' 5x3
0
1
0
0
0
Actual Target
0
0
1
0
0
V 5x1, one hot vector of “Set”
W 3x5
V 5x1, predicted one hot vector of “Can”

Fig: Example Neural network for CBOW

The **Fifth feature** is Word Mover's distance which is mostly used for semantic matching (Zihan Chen, Quora Question Pairs, n.d.). It can also detect the similarity of questions with totally different words but has the same meaning. The Word-to-Mover measures the difference between two questions by calculating the distance that the Word2Vec embedding of one question have to travel to reach the embedded similar word in other question.



```
import gensim

from gensim.models import Word2Vec

model = gensim.models.KeyedVectors.load_word2vec_format('./word2Vec_models/GoogleNews-vectors-negative300.
bin.gz', binary=True)
```

```
distance = model.wmdistance(question1, question2)
print('distance = %.4f' % distance)
```

```
distance = 1.8293
```

Fig: Shows the code for calculating Word mover distance using pretrained Google News Word2Vec model

The **Sixth feature** is the character n-gram encoding is a very effective method to understand the underlying meaning, although it has higher dimension compared to word embedding works better (Gaurav Singh Tomar, n.d.). Here, we generate character embeddings(feature vector) for each character similar to Word2Vec and sum them to represent words and their vector. It involves some additional computation cost.

Combination of Primary Features

Four other features which are combination of two or more considered above are explained here.

First, to represent a sentence we use word embeddings and then take the mean of all the word embeddings in the sentence to represent sentence embedding, here we did not consider the effect of word sequences.

Second, using the TF-IDF for each word and multiply it with its word vector and then take the mean of words embeddings to represent words. If we are not considering TF-IDF score for words, there is a probability that two completely different sentences/Questions could have similar representation which can incur the loss of information leading to incorrect results. We compute the similarity using these metrics by calculating the inner product of two Question vectors.

The third metric is calculated by using the similarity scores calculated previously and common word percentage(the number of common words in the questions divided by the total length of two questions).

Fourth metric considered is called "length difference percentage" which calculates the difference in length divided by the total length of two questions.

There are many other handcrafted features considered in research papers(not defined in this paper) and each advance feature generates better results and help machines understand the context better. The features detailed here are few among them which are relevant to my research.

Machine learning Algorithms

There are many supervised learning algorithms implemented on this dataset. As new state of the art methods being available in deep learning areas, we can improve the accuracy further by using them on Quora Dataset. Nowadays, even the machines are able to understand the context and emotions of the situation by using deep learning algorithms. A better understanding can result in accurate classification of questions. Here, I will be discussing some of the machine learning and deep learning algorithms used previously for Quora dataset and these algorithms use the above-mentioned features as input for the model.

Traditional and Ensemble ML algorithms

Firstly, I want to discuss the **Random Forest algorithm** which is currently used by official Quora website for classification (SambitSekhar, 2017). It is a very popular machine learning algorithm which is widely used for classification tasks. Random Forest is an ensemble method which is a combination of multiple decision trees(weak learners). A decision tree is a supervised learning method and mostly used for classification problem. Decision trees perform feature selection and use the Gini index or information gain to decide on the split. We can have many leaf nodes based on the number of conditions and level of the tree; the tree can go very deep as the split conditions increase. In random forest, we combine many randomly selected decision trees and using consensus/voting decide on final output for classification. There are many hyperparameters which can be tuned using GridSearchCV to get the best tree which can have better accuracy on predictions.

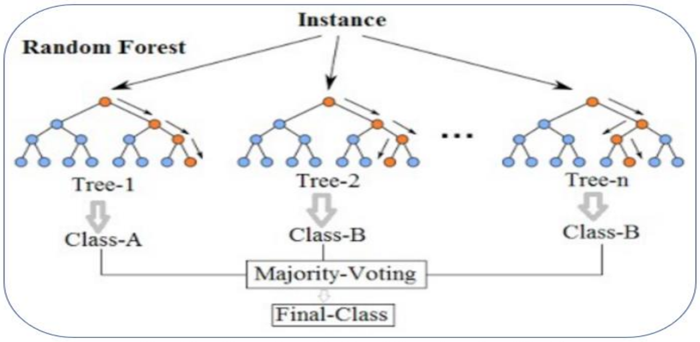


Fig: Random Forest algorithm for classification

For the Quora dataset, RF algorithm uses handcrafted features which are mentioned above and also the cosine similarity of the average of word embeddings vectors for two questions and input these to the machine learning algorithm for classification results. A cosine similarity metric is used to determine how similar the Questions are irrespective of their size. Mathematically, it is defined as the measure of the cosine angle between two vectors which are projected in a multi-dimensional space.

Second, the **Support Vector Machine** classifier algorithm is used for classification of Quora dataset. Support Vector Machine(SVM) classifier looks at the extremes of the data set and draws a decision boundary also called **HyperPlane**. SVM implies that only support vectors are important for classification, Support vectors are the points that are closest to the opposing class. This algorithm uses a margin of fixed width for both classes which is dependent on support vectors. To classify the data which is not linearly separable, SVM uses a **Kernel Trick**(it takes input vectors in the original space and returns the dot product of the vectors in the feature space) which projects the data from lower dimensions to higher dimension to separate the data linearly, and this trick also helps in reducing the computation expenses in conversion to higher dimension.

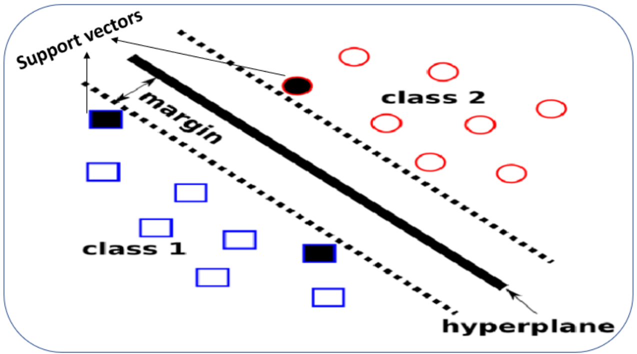


Fig: Support Vector Machine Classifier Approach

By training the dataset using SVM classifier and using the four features(combination of primary features) which are defined and elaborated in the last part of the feature engineering section, the dataset acquired the test set accuracy of 67% (Lei Guo, n.d.). In one other paper, which also used SVM classifier for prediction gave an accuracy of 85%, as the features used in this model are the preliminary features which are specified above and some other Fuzzy string-matching features (Shashi Shankar, n.d.). We can say that the feature engineering and extraction of proper features for text data can make a vast difference in prediction accuracy. As even a traditional machine learning algorithm performs equivalent to Deep learning models when the features selected are optimal.

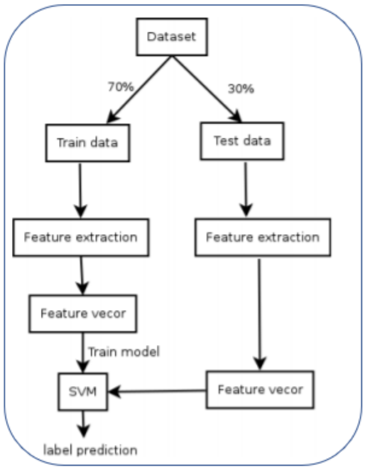


Fig: Model workflow for SVM classifier

The third traditional ML algorithm is **KNN(K-nearest neighbors)**, though it a very simple algorithm gave good results for Quora dataset (Lei Guo, n.d.). It considers the K (K can be any integer value greater than or equal to 2) number of nearest neighbors to a new point in the dataset and uses voting to classify that point. The KNN model gave the test set accuracy of 82% which used the same features as the SVM classifier.

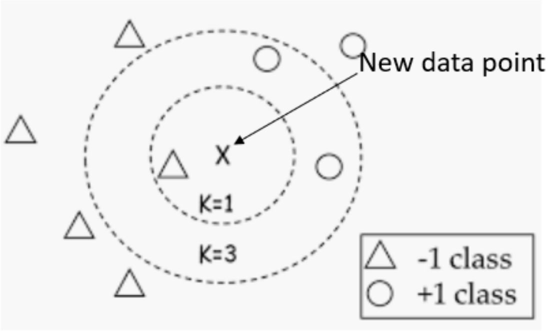


Fig: Example of KNN classifier

There are few other traditional ML algorithms which are used for Quora dataset classification are **Logistic Regression** which gave an accuracy of 75% with hyperparameter tuning, **Adaboost** gave an accuracy of 69.38%, Decision Tree classifier gave an accuracy of 69% (Zihan Chen, Quora Question Pairs, n.d.) and **Gradient boosting machine(GBM)** which gave an accuracy of 84% (surabathula, n.d.).

Deep Learning methods

Inspired by the recent advances in deep learning, I explored many papers based on deep learning techniques for Quora dataset. The basic and primary Deep learning method is Simple Neural network also called a **Feedforward neural network**. Feedforward Neural network is the supervised learning algorithm which is inspired by the functioning of the human brain simulates it artificially by triggering neurons for every action. Artificial neural network methods learn and improve by back-propagating errors and improve the accuracy on each iteration(epoch) through the network. It uses a famous method called Gradient descent to backpropagate errors. These networks are good at analyzing and predicting for numerical data but for sequential text data we use recurrent neural network architectures.

Recurrent Neural Network

The recurrent neural networks are very effective at handling sequential data(like word sequences), it captures the previous input in memory and the computations used, so the information can cycle inside for a longer period. Usually, it takes 2 inputs the current input and previous state output(hidden state output) to generate a new computation. Though it captures long term data, it has limitations in learning long-term dependencies which is called the vanishing gradient problem and the performance gets worse as the input sequence length increases because the network has to remember a large amount of data.

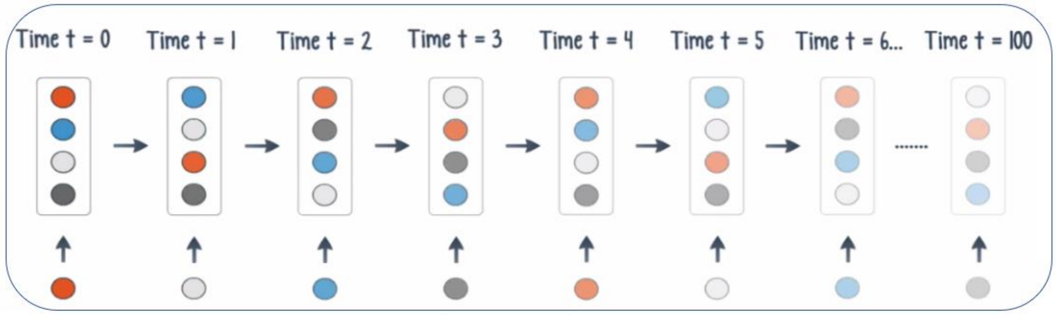


Fig: Showing RNN Architecture with vanishing gradient

Given the limitation of Recurrent neural networks which are unable to capture long term dependencies, we use **Long short-term Memory(LSTM)** units in replacement to simple neurons in Recurrent neural network architecture. LSTM are used for sequence modeling as they can learn long term dependencies. LSTM cell architecture consists of 3 different gates(input gate, forget gate and output gate) and a cell state to maintain the smooth flow of required information through network. Though being a complex network, they are very good at capturing long term dependencies.

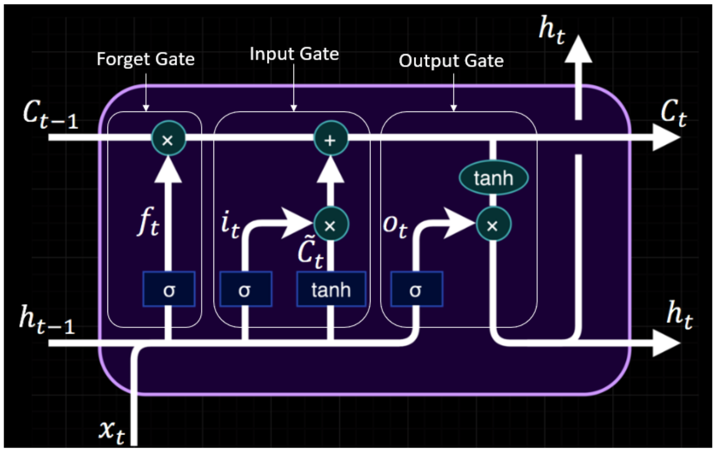


Fig: LSTM cell architecture showing cell state and gates

Siamese Manhattan LSTM

The Siamese Manhattan LSTM architecture used for prediction on Quora dataset has achieved greater accuracy (Cohen, 2017). The Siamese architecture consists of two identical LSTM sub-networks each processing a question which has been tied with weights. The weights are calculated by using Word2Vec embeddings of each word in the question and finding the mean of Vectors to get Sentence Embedding for the question. As both questions have different input size(different size of embedding), the network first converts the variable length input sequences to fixed length vector by concatenating the input vector.

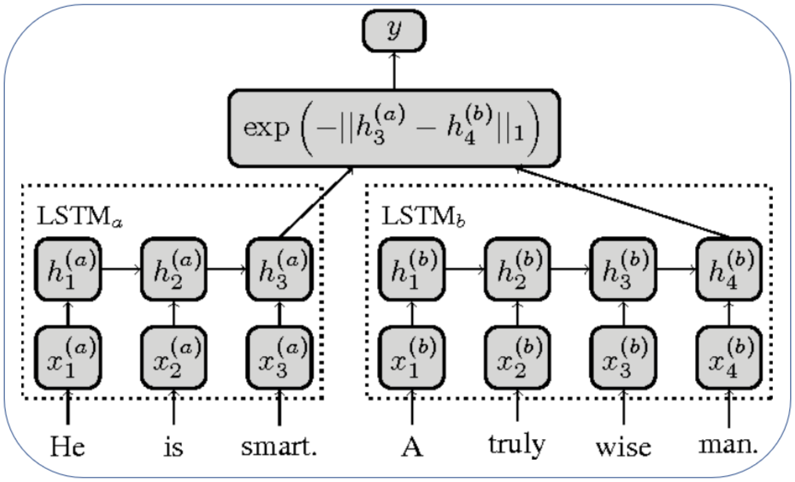


Fig: Siamese Manhattan Architecture

Then, the Manhattan similarity metric(L1 Norm) is used to find the similarity score between two questions by capturing the similarities and differences in the output of the model. Though this model has achieved better accuracy than traditional ML models but it has certain limitations such as it converts the variable size input vector to a fixed size input vector, this incurs loss of information and lead to improper results and the other drawback is that this model uses LSTM cell which captures the information about the language only in one direction which will lead to an incomplete understanding of the language.

There are many deep learning models implemented on this dataset. Few examples are GRU(Gated recurrent unit) + SVM, GRU + dropout + Adaboost (Ameya Godbole, n.d.), Siamese CNN(Convolutional neural network), Feedforward Neural network with character encodings (Gaurav Singh Tomar, n.d.) and many more.

Research Design and Methods

Data Acquisition and Dataset Description

The dataset of interest for the study is the dataset published by the QA website Quora.com containing 400K annotated question pairs with binary labels (SambitSekhar, 2017). The training dataset contains 404,290 valid Quora question pairs. Each data entry consists of the ID of the question pair, the unique IDs and full text of each question, as well as the binary target variable indicating whether the two questions are duplicates (i.e. have the same meaning) or not. As a comparison, the test dataset contains 6 times the number of question pairs in the training dataset which has the value of 2,345,796(question pairs) but without any predefined labels. The test set is retrieved from the Kaggle competition hosted in 2017. Some questions also include special characters such as mathematical symbols, foreign language characters, etc.

The duplicate labels in the training set are provided by human experts and are inherently subjective and prone to errors (Shankar Iyer, n.d.). As a result, the labels on the training data set to signify a reasonable consent and are considered true but are not 100% accurate.

Data Analysis

The Data Analysis is a very critical step of machine learning as in this step we can uncover many unknown facts about data. Through my analysis, I have found that there are only 37% positive samples in the data and 63% negative samples, we can say that the data is highly imbalanced.

By doing further analysis many facts have been disclosed. Firstly, there are about 54,000 unique questions in the data and 20% of them are repeated more than once. There are few questions which are repeated even more than 100 times, can be considered as outliers.

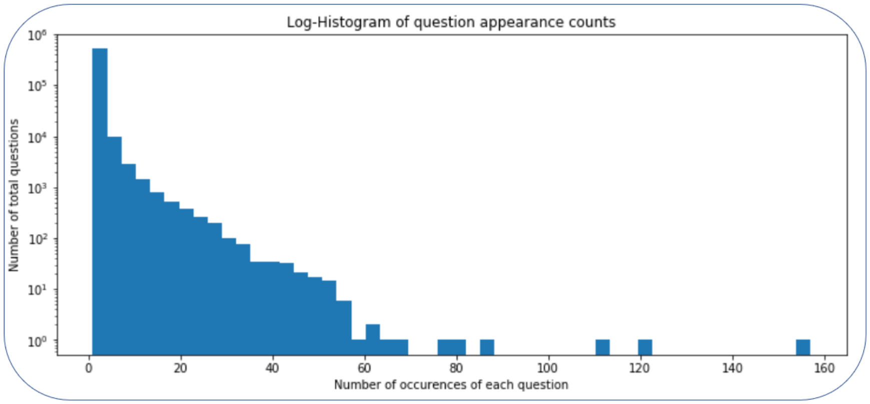
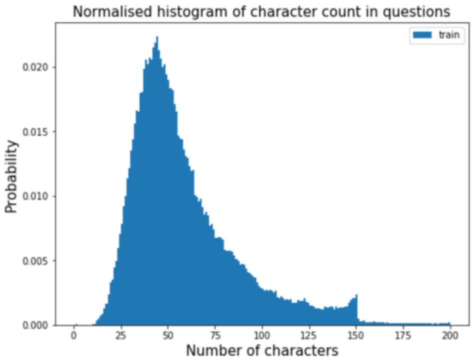
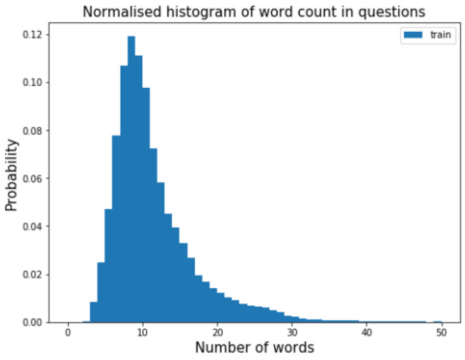


Fig: Showing histograms for number of questions with count

There is a sharp dip at 150 characters this can be because the official Quora website allowed the character limit of 150. But still, there are questions with more than 150 characters. To analyze the questions with more than 150 characters we need more data such as the date of the question upload, when did the 150-character limit is applied by Quora, etc.





And the word count for questions is ranging from 1-30 with few outliers containing greater than 30 words and the mean of the word count is approximately 10.

Secondly, I analyzed the frequency of occurrence of words using a word cloud and found a few words which have a very high frequency of occurrence. Some examples are Hillary Clinton, best way, difference, Donald trump, India, etc. There is lot more research I did on data like analyzing the question length(words and characters) for both questions and finding the difference in length, analyzing the number of words that match for both questions, if the words matches, are they considered duplicate or not, etc. Thus, this analysis has helped me a lot to deeply understand the data.

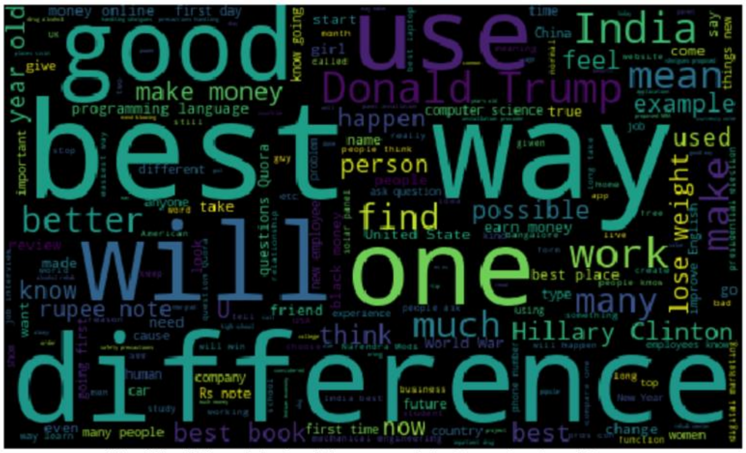
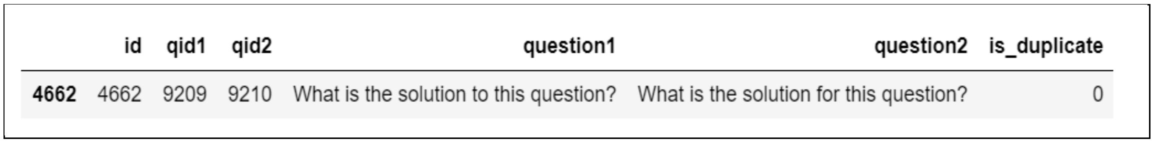


Fig: Word Cloud derived from words in Question1 and 2

Furthermore, there are 3 more important observations from analysis.

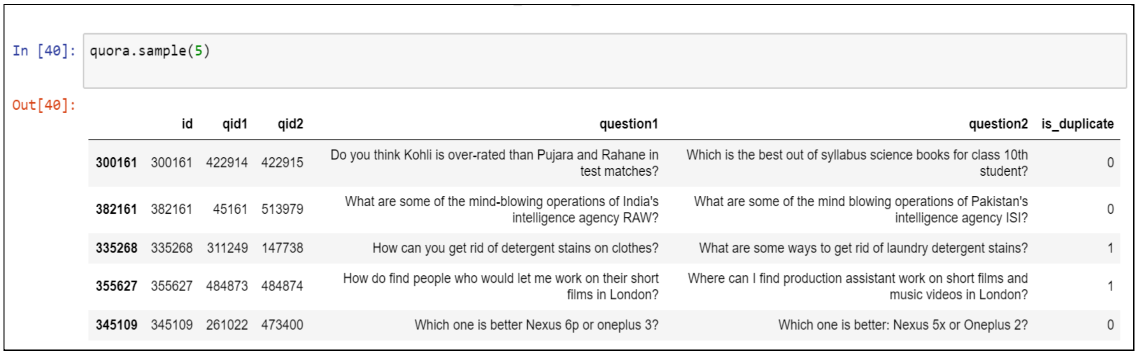
First, training dataset consists of wrongly labeled data as I observed some questions which are duplicates but labeled wrongly. Following is the example of one such wrong labeled question pair:



	id	qid1	qid2	question1	question2	is_duplicate
4662	4662	9209	9210	What is the solution to this question?	What is the solution for this question?	0

Second, there are some questions which share all the words but as the order of the words has changed, they are marked as non-duplicates. This indicates that we have to be careful while doing feature extraction on this dataset as simple word-to-word matching can fail and predict wrongly for test data.

Third, the first words of the questions are mostly interrogative, such as What, Why, Can, Who, How, etc. and they are not significant in determining the question pair as duplicates (Zihan Chen, Quora Question Pairs, n.d.). As for most of the questions, though the first words are the same they are categorized as non-duplicates. Few examples are shown below.



```
In [40]: quora.sample(5)
```

Out[40]:

	id	qid1	qid2	question1	question2	is_duplicate
300161	300161	422914	422915	Do you think Kohli is over-rated than Pujara and Rahane in test matches?	Which is the best out of syllabus science books for class 10th student?	0
382161	382161	45161	513979	What are some of the mind-blowing operations of India's intelligence agency RAW?	What are some of the mind blowing operations of Pakistan's intelligence agency ISI?	0
335268	335268	311249	147738	How can you get rid of detergent stains on clothes?	What are some ways to get rid of laundry detergent stains?	1
355627	355627	484873	484874	How do find people who would let me work on their short films in London?	Where can I find production assistant work on short films and music videos in London?	1
345109	345109	261022	473400	Which one is better Nexus 6p or oneplus 3?	Which one is better: Nexus 5x or Oneplus 2?	0

Research Model

Inspired by the Siamese Manhattan architecture (Jonas Mueller, n.d.), I will use same architecture for my research but with changes in the internal layers. For my novel architecture, I will be using two similar Bidirectional Transformer cells that rely on Attention technique instead of LSTM cells in the Siamese Manhattan architecture.

My structure consists of 2 input layers each consisting of word embedding of the questions, which are passed to the Transformer network and then it generates a sentence embedding or vector representation of each question. Then, we compare the two sentence vectors using Manhattan similarity metrics which applies the Manhattan distance metric or L1 norm, and the output is again passed through the attention layer for capturing similarities entirely in the combination of two questions which is obtained after applying Manhattan distance metric and then predicting the output.

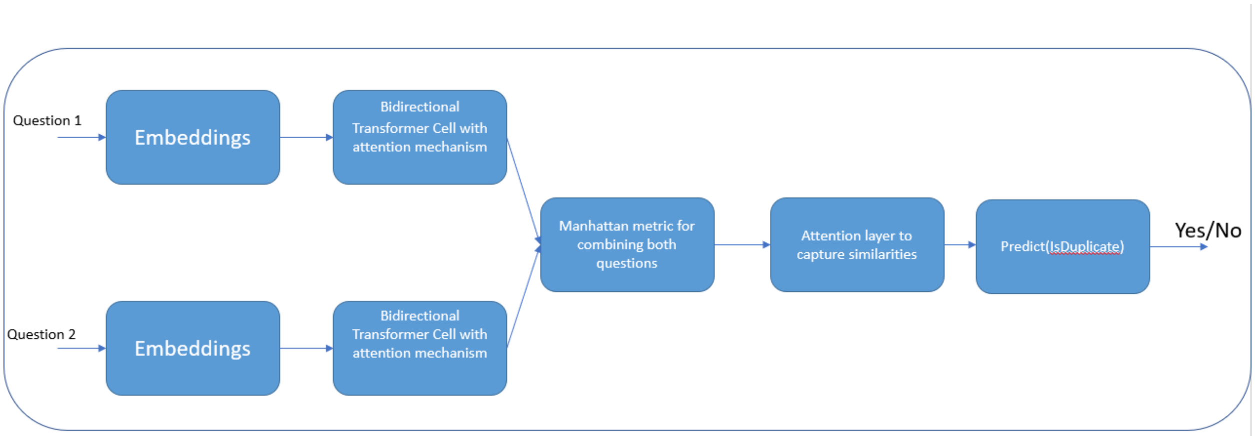

Question 1
Embeddings
Bidirectional Transformer Cell with attention mechanism
Question 2
Embeddings
Bidirectional Transformer Cell with attention mechanism
Manhattan metric for combining both questions
Attention layer to capture similarities
Predict(IsDuplicate)
Yes/No

Fig: My architecture for predicting Quora questions as Duplicate or not

Here, I want to implement the advance version of Encoder-Decoder model called Transformers(Attention is all you need). It is a novel neural network architecture for **Natural Language Understanding**. To understand the Transformers network, I analyzed the Encoder and Decoder model first and then moved to Transformers model.

Encoder-Decoder Sequence to Sequence model is used when we want to map from variable size input to variable size output (Kostadinov, n.d.). This model contains 3 stages: The Encoder, Intermediate Encoded Vector, and The Decoder. The input data will be passed through a stack of several Recurrent Neural Network cells taking one word as new input at each RNN cell and the final hidden layer output is considered as Intermediate Encoded Vector which consists of all the input information in encoded format. For each RNN cell in Encoder, the input is a single word from the question and a hidden state output vector of previous RNN, this is the Encoder part which encodes the input sequence into an Encoded vector(Final hidden State). Then, this vector will be passed through Decoder which is similar to encoder as it consists of a stack of several recurrent units, but the only difference is that it takes only the hidden state vector as input and predicts the output using the SoftMax activation(It calculates the probability for each output) function in the decoder layer. Below is the architecture of the Encoder Decoder Model:

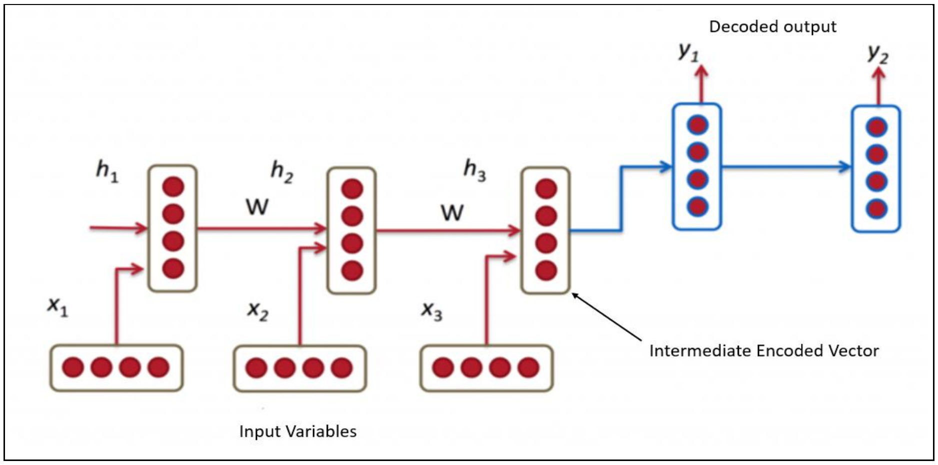


Fig: An Encoder-Decoder Architecture.

The drawback of the above Encoder-Decoder model is that the intermediate encoded vector has to remember all the input data in an encoded format and then decode it to produce output. This can be challenging as the Encoder's final hidden Layer output is a very long encoded vector consists of all the input data in encoded format. This may lead to loss of long-term information same as the drawback of the recurrent neural network. To overcome this drawback, the Transformers are introduced which uses attention technique and parallel computation.

The Transformer's basic architecture is similar to Encoder-Decoder model but there is a change in internal layers of Encoder and Decoder. In Transformer's architecture, each encoder cell takes the input as a weighted sum of all the words in the sentence and the weights for each word are determined by the features used(such as TF-IDF, Word2Vec embeddings, etc.) for input words.

Attention is defined as "The active direction of the mind to an object" (A Beginner's Guide to Attention Mechanisms and Memory Networks, n.d.). In machine terms, Attention is described as a distribution of importance unevenly across a field through focus, which brings certain inputs to the foreground and minimizes the importance of others by moving them to the background. As we all know the human vision works in a similar fashion, this attention mechanism in a neural network is inspired by human vision and focus. But in real time the attention mechanism in the neural network comes with a cost as it has to process all the words multiple times to select the important word i.e., to select the final focus word. This can be computationally very expensive, so the focus has shifted to Transformers with Attention instead of RNN with attention. The transformers can take advantage of parallelization in GPU's which traditional RNN cannot do as they use a sequential approach to process data. Below is the architecture of the Attention model.

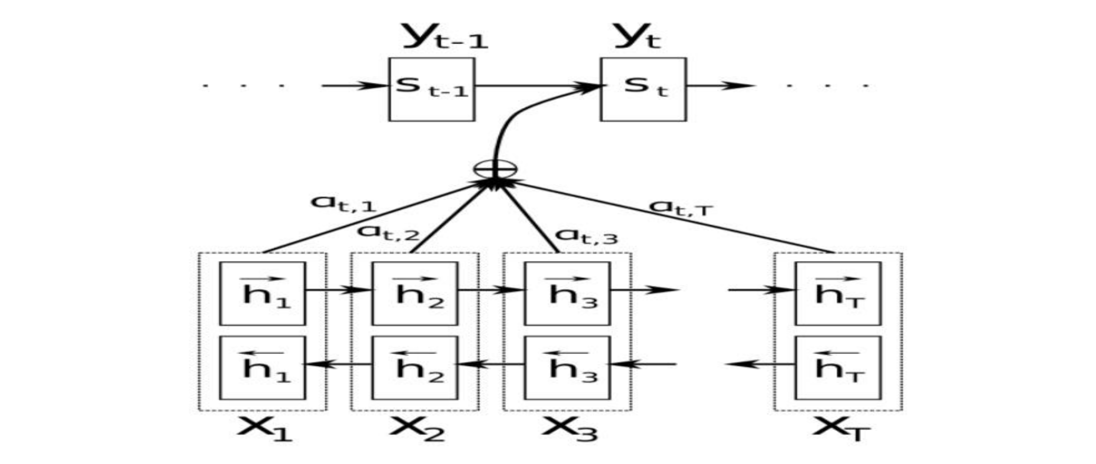


Fig: Attention model, which uses the weighted sum of inputs at each step.

The transformer uses a Multihead Self-attention mechanism in the Encoder and Decoder model, which computes the weighted sum of input parallelly for all the words in the question. It eliminates recurrence completely and uses attention to handle input and output dependencies which also reduces computational complexity as the tasks are executed parallelly. Self-Attention or intra-attention is the attention which words pay to each other is the same sentence/question (A Beginner's Guide to Attention Mechanisms and Memory Networks, n.d.).

Obstacles

The main obstacle for this research is extracting features and selecting the best among them as each has its own disadvantage and second obstacle is implementing transformers on text data which is very challenging.

Anticipated Results

As I am using Transformers with attention techniques in my approach to solve the problem of detecting duplicate Quora questions, this can improve the accuracy and also reduces a lot of computation as the transformers use parallel processing and take advantage of GPU's power. The Attention mechanism assigns the higher weight to more important words which can lead to vast difference while calculating similarity value. The novel approach can also compare variable length inputs as it uses encoder decoder approach is an advantage.

To test a classification problem, the metrics used for my model are log loss and confusion matrix. To achieve better accuracy for the model we have to reduce the logarithmic loss which is a value between 0 and 1. The perfect model which has 100% accuracy has the log loss score of 0. The confusion matrix shows the 4 values for a binary classification which are true positives, false positives, true negatives, and false negatives. These metrics give us a clear understanding of our output.

The anticipated accuracy for my proposed model will be better than the existing models for this dataset and while implementing this model tuning hyperparameters and using regularization can help to optimize the model.

Visualization Tools used

Used Jupyter notebook and python libraries for analysis on data and visualization tasks.

Discussion and conclusion

Significance of the Study

The semantic understanding of a text has vast applications other than just in detecting duplicate Quora Questions, so by solving this challenging problem we can also address many other problems which can be solved using NLU and semantic understanding. Few of the other applications of Natural language understanding are as follows:

i) **Relation extraction**: Extracting useful information from large volumes of data which is generated on daily basis, the information such as identifying entities and their relationships in a big data is called Relation extraction, for e.g., the occurrence of person and organization in a sentence will be linked together using relation extraction. This can be achieved by Natural language understanding and have many applications such as Question Answering forums, Information Retrieval which can benefit from Relation extraction.

ii) **Sentiment analysis**: Sentiment analysis is to categorize the writer’s opinion towards a product or service into positive, negative or neutral classes by analyzing the text computationally. This is another application of natural language understanding by machines.

iii) **Chatbots**: These interacts through instant messaging as we see in all websites nowadays, these try to artificially replicate the patterns of human interactions. To achieve this, natural language understanding of machines is a must. They offer companies an opportunity to improve their customer engagement process and operational efficiency by reducing the cost of customer service.

Insights of the Study

The are many insights from dataset and domain analysis. Two most important of them are :

First, analyzing data carefully before approaching for a solution to make prediction can make a vast difference while building model. As speaking for this data, the labels provided were not right for some questions specified earlier in data analysis section.

Second, by knowing the domain of the problem and analyzing it properly from previous papers and other resources can also help while building ML models. And also knowing the domain can address many questions raised while analyzing data and doing research for prediction.

Tools and Skills required

Proper data analysis is the key tool. I used python libraries and Jupyter notebook for coding and analysis. For Machine learning algorithms, SKLearn and Keras with TensorFlow backend packages are used.

Recommendations and Future Study

For future study, I want to work on mainly two areas. First, I will try different neural network architecture based on further analyzing the methods used in the area and try different embeddings for questions which can give better understanding of context. Further data analysis can also be done for obtaining new features. Will work on large training data and on Bigdata tools to handle data. Second, want to extend my work to other applications which are similar to the Quora questions such as Automatic Question Answer grading system, Semantic analysis and Text paraphrase detection problem.

References

A Beginner's Guide to Attention Mechanisms and Memory Networks. (n.d.). Retrieved from skymind: https://skymind.ai/wiki/attention-mechanism-memory-network

Amazon Alexa. (n.d.). Retrieved from https://developer.amazon.com/alexa-skills-kit/nlu?tag=askcomdelta-20

Ameya Godbole, A. D. (n.d.). *Siamese Neural Networks with Random Forest for detecting duplicate question pairs*. Retrieved from [18] https://www.researchgate.net/publication/322675027_Siamese_Neural_Networks_with_Random_Forest_for_detecting_duplicate_question_pairs

Cohen, E. (2017, June 7). *How to predict Quora Question Pairs using Siamese Manhattan LSTM*. Retrieved from MLReview: https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07

Collier, A. B. (2015, 12 14). *Making Sense of Logarithmic Loss*. Retrieved from datawookie: https://datawookie.netlify.com/blog/2015/12/making-sense-of-logarithmic-loss/

Donges, N. (2018, August 7). *Introduction to NLP*. Retrieved from https://towardsdatascience.com/introduction-to-nlp-5bff2b2a7170

Gaurav Singh Tomar, T. D. (n.d.). *Neural Paraphrase Identification of Questions with Noisy Pretraining*. Retrieved from https://aclweb.org/anthology/W17-4121

Jonas Mueller, A. T. (n.d.). *Siamese Recurrent Architectures for Learning Sentence Similarity*. Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12195/12023

Kostadinov, S. (n.d.). *Understanding Encoder-Decoder Sequence to Sequence Model*. Retrieved from Towards Datascience: https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

Lei Guo, C. L. (n.d.). *Duplicate Quora Questions Detection*. Retrieved from https://pdfs.semanticscholar.org/4c19/2b8f45b1e913ee7da32624cd7559eccb0890.pdf

SambitSekhar. (2017). *First Quora Dataset Release: Question Pairs*. Retrieved from https://www.kaggle.com/sambit7/first-quora-dataset

Shankar Iyer, N. D. (n.d.). *First Quora Dataset Release: Question Pairs*. Retrieved from DATA @ QUORA: https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

Shashi Shankar, A. S. (n.d.). *Identifying Quora question pairs having the same intent*. Retrieved from https://pdfs.semanticscholar.org/ff89/c3925581c9551323e9fd7deff699fa149dcb.pdf

surabathula, s. (n.d.). *Identifying Duplicate Questions: A Machine Learning Case Study*. Retrieved from https://medium.springboard.com/identifying-duplicate-questions-a-machine-learning-case-study-37117723844

Tuvtran. (n.d.). *project-based-learning*. Retrieved from https://github.com/tuvtran/project-based-learning/commit/a9aba7599f20fbe0206c5457ab5bedd998c5f283

Yanting. (2017, July 16). *Identifying Duplicate Quora Question Pairs (Kaggle Competition Bronze Medal Winner)*. Retrieved from https://emmating.github.io/identifying-duplicate-quora-question-pairs-kaggle-competition-bronze-medal-winner.html

Zihan Chen, H. Z. (n.d.). *Quora Question Pairs*. Retrieved from http://xiaojizhang.com/files/quora-question-pairs.pdf

Zihan Chen, H. Z. (n.d.). *Quora Question Pairs*. Retrieved from http://static.hongbozhang.me/doc/STAT_441_Report.pdf