Identifying Duplicate Questions on Quora

Harsh Patel aka Chaudhari
Computer Engineering Department
San Jose State University
San Jose, CA, USA
harsh.patelakachaudhari@sjsu.edu

Thejaswi Kampalli
Computer Engineering Department
San Jose State University
San Jose, CA, USA
thejaswi.kampalli@sjsu.edu

Manogna Sindhusha Mujje
Computer Engineering Department
San Jose State University
San Jose, CA, USA
manognasindhusha.mujje@sjsu.edu

Pooja Shivasharanappa
Computer Engineering Department
San Jose State University
San Jose, CA, USA
pooja.shivasharanappa@sjsu.edu

***Abstract*— The aim of this paper is to detect duplicate questions on question and answer sites such as Quora and Stackoverflow. The detection is carried out in the semantic level based on the intent of the question using Natural Language Processing and Machine Learning techniques. We are using a technique known as Cosine similarity to calculate how similar the questions are to one another. We also tried using Support Vector Machine on vectors created with the help of pre-trained Word2Vec model by Google.**

***Index Terms*— question sets, compressed sparse row-matrix, cosine similarity, duplicate questions.**

I. Introduction

As the number of users increase on question and answer sites, it is more likely that the users ask similar questions. This is a bad user experience both for the writers and seekers as the answers get fragmented across different versions of the same question. Quora decided to address this problem and launched a challenge on Kaggle to solve this Natural Language Processing problem by applying some advanced techniques to classify whether question pairs are duplicates or not. Quora has also released a dataset for those who are interested in solving this problem. The aim is to automate the way of detecting if pairs of question text actually correspond to semantically equivalent queries. Our approach includes computing cosine similarity for the vector representation of each question in the question pair. So, we first compute a compressed sparse row matrix where each row represents a question and then calculate cosine of the angle between the vector representation of two vectors. The cosine of the angle determines if the pair of the questions are duplicate or not. Another approach we tried was treating the problem as Natural Language Understanding rather than Natural Language Processing by using Google's Word2Vec model to generate vectors in a 300 dimensional Vector space such that it becomes easy to calculate the semantic differences.

II. Dataset analysis

The Quora dataset consists of over 400,000 potential question duplicate pairs. Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair. The data contains 255045 negative samples and 149306 positive samples. The positive samples thus amount to 40% of the total dataset.

The format of the data provided by quora is as shown in the Fig 1.

Quora Sample Data Set

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Fig 1.
The data is provided in tsv(tab separated values).

III. Cosine Similarity Approach

A. Preparing CSR Matrix

The data provided by quora is imported using Pandas Library and stored in form of a list. Each question pair represents two lists: list1 and list2. The list 1 contains the first question of the question pair while the list 2 contains the second question. In order to compute a CSR(compressed sparse row matrix) matrix the two lists are appended together so we get one CSR matrix with each row representing one question.

The construction of CSR matrix is as shown in Fig 2 and Fig 3. Suppose we are to prepare a CSR matrix for a set of three questions:

1) What is your Name?
2) It is not raining, is it?
3) Where is your house?

Each new word is assigned a new word Id. If a word is repeated, it is assigned the same word Id that has been allotted to that particular word in its first occurrence. In our example, id 0 is assigned to 'What', id 1 to 'is' and so on. When the word 'is' is repeated it is given the same value 1, the value that is allotted initially. This process is repeated for all the words in the matrix. and then we calculate three different set of values: Index , Value and Pointer.

- Index shows each of the indices present in each question in a non repetitive manner starting from the least to the highest.
- Value determines the frequency of each unique word in the question.
- Pointer shows the starting and ending point of each different question.

These values of Index, Value and Pointer are then passed as input values to the csr_matrix() function from the scipy library, to obtain a CSR matrix as output as shown in Fig 3.

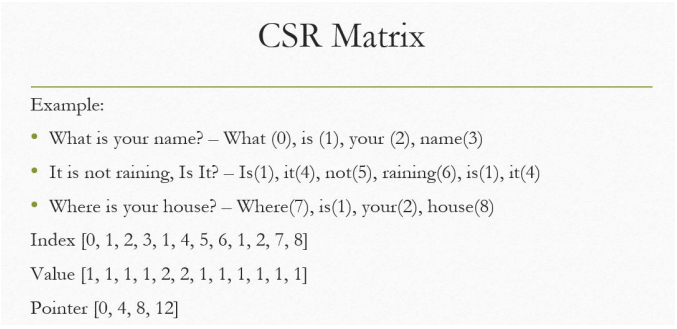

Fig 2

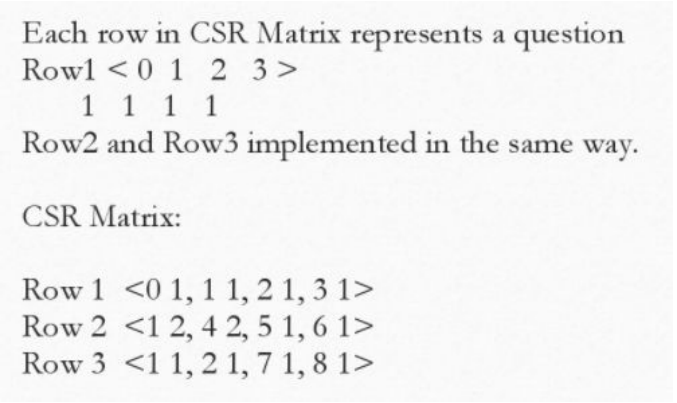

Fig 3

Each row of CSR matrix represents all the words and their frequency present in a question.

B. TF-IDF Vectorizer

TF-IDF often referred to as Term Frequency - Inverse Document Frequency is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus. Calculating Inverse Document Frequency, often optimizes the computational power and helps in deriving faster and accurate results. It is usually a tedious task to check for a word in each document, rather checking the documents that contain the specific word we intend to search for makes one's job easier. Hence, the transpose of the CSR matrix is calculated to find the Inverse Document Frequency.

The ideal formula to calculate Inverse Document Frequency is:

$$idf = \log (N/n_t)$$

where
N = total number of documents in the corpus
n_t= frequency of the nth term

C. Cosine Similarity

In order to compute cosine similarity, the dot product between the vector representation of the two questions in the question pair is calculated. To further simplify the calculations, the two vectors are normalized i.e the vectors representing the questions in question pair. Then the cosine of the angle between normalized vectors is calculated. Calculations are as performed in the Fig 4.

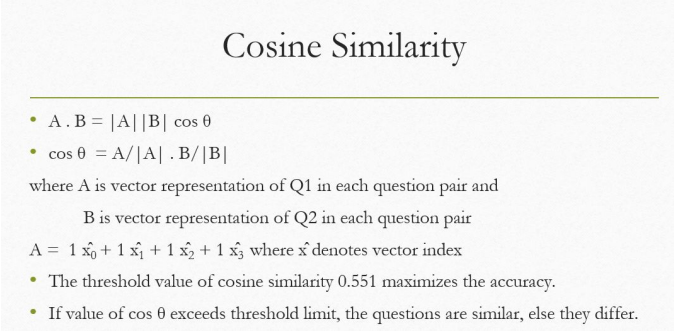

Fig 4.

The cosine of the angle between the two normalized vectors determine if the two questions are similar or not. It is more likely that similar questions have cosine of the angle value closer to 1. For purpose of solving the problem, we have taken the threshold value of cos θ to be 0.551 as it maximizes the accuracy. If the cosine of the angle between two vectors comes out to be more than the threshold value, we conclude that the questions are similar(duplicate), otherwise they are different.

We have compared the results obtained by running cosine similarity against the results provided by the Quora and it gives us accuracy of about 65% approximately.

IV. Supervised Learning Approach

A. Word2Vec model

In order to produce word embeddings on vector space we used Google's Word2Vec model. With this approach we can capture different degrees of similarities between words. Vector arithmetics can be used to reproduce semantic patterns. Patterns like "Man is to women like brother is to sister" can be represented using algebraic operations on word vectors like "Brother" - "man" + "women" which produces a resulting vector close to the vector of "sister".

B. Semantic distance between Sentences

The model we used produces a vector of 300 dimensions for each word. In order to represent a sentence as a vector, we compute the average of all word vectors resulting in a single vector of 300 dimensions representing a sentence. In order to calculate semantic distance between two sentences we compute the absolute difference between those two sentence vectors such that if the sentences speaks about different contexts the absolute difference will be high and if the contexts they are speaking about are same then the absolute difference will be low.

C. Supervised Learning using Support vector Machines

In order to classify the question pairs as duplicates and non duplicates we have a need to train a model on the data set provided by Quora. We use Support Vector machines as Supervised Learning model which classifies the duplicates from non duplicates by using a hyperplane in the 300 - dimensional vector space

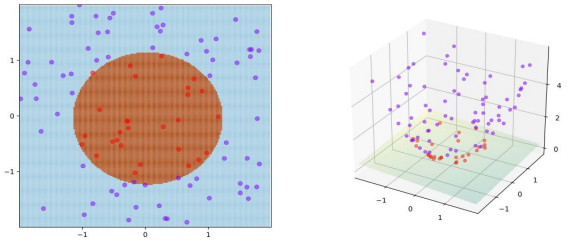

Fig 5

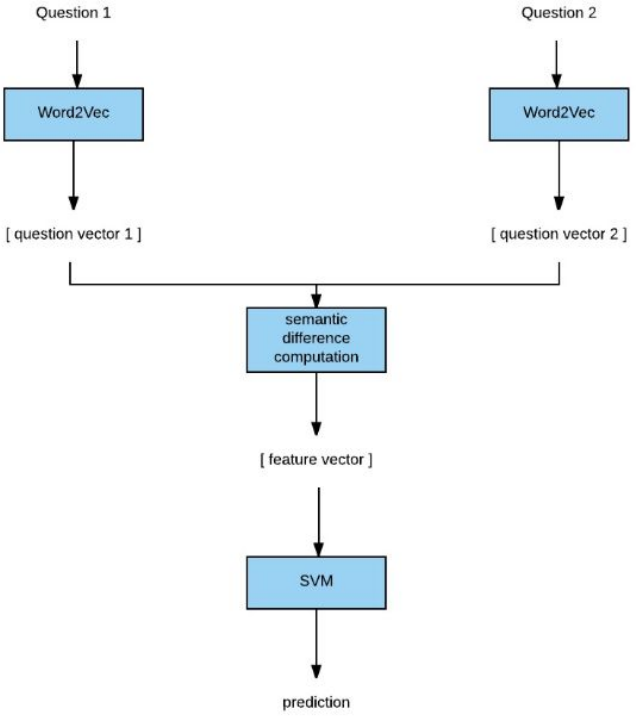

Fig 6

A SVM model is a portrayal of the examples as points in space, mapped with the goal that the examples of the different categories are partitioned by a reasonable gap that is as wide as it could be allowed. New examples are then mapped into that same space and anticipated to have a place with a category in light of which side of the gap they fall. Using this approach we were able to test 10000 question pairs resulting in an increased accuracy.

V. Conclusion

The approach which used cosine similarity resulted in a low accuracy. This approach has some difficulties that is hard to overcome. It is often that two questions with very similar words have very different meaning. For example, “What is the step by step guide to invest in share market in india?” and “What is the step by step guide to invest in share market?” have different meanings with additional two words. But this is predicted as duplicates by cosine similarity. Cosine Similarity only deals with length of the words and frequency of them in a sentence. Thus, this approach is not proven solution as it has no concern about the semantic and syntactic relations.

Secondly, we tried using word2vec model to build vectors for each word. This enables our model to look for semantic differences between sentences. Which overcomes the difficulties faced in the previous approach. This aids in increasing the accuracy as previously mentioned cases are dealt with semantics. Then these vectors are subjected to SVM for learning and then tested on a different dataset leading to an increased accuracy.

Acknowledgment

We are extremely thankful to our professor Mr. Rakesh Ranjan for providing us this opportunity and encouraging us to select the project topic from Data Science and Machine Learning concepts.

References

[1] Elkhan Dadashov, Sukolsak Sakshuwong, Katherine Yu, Quora Question Duplication”

[2] Lili Jiang, Shuo Chang, and Nikhil Dandekar. (2017, Feb 13). Semantic Question Matching with Deep Learning [Blog post]. https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning

[3] Matthew Honnibal. (2017, Feb 13). Deep text-pair classification with Quora’s 2017 question dataset [Blog post]. Retrieved from https://explosion.ai/blog/quora-deep-text-pair-classification

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

[5] Wikipedia: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation

[7] Dmitriy Fradkin and Ilya Muchnik. Support Vector Machines for Classification.

[8] Issam El-Naqa, Yongyi Yang, Miles N. Wernick, Nikolas P. Galatsanos, Robert M. Nishikawa. A Support Vector Machine Approach for Detection of Microcalcifications.

[9] Word2Vec-GoogleNews-Vector: https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[10] Goldberg, Yoav; Levy, Omer. "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method".