QUORA QUESTION PAIR SIMILARITIES

Natural Language Processing (NLP)

Submitted to: **Vlado Keselj**

By: **Aman Arya (B00816348)** & **Navneet Prakash Singh (B00810744)**

1 CONTENTS

2 Abstract 2
3 Introduction 3
4 Related Work 4
5 Problem Definition and Methodology 5
5.1 Cosine Similarity 6
5.2 Machine Learning Algorithm (Naïve Bayes) 6
6 Experiment Design 8
6.1 Dataset 8
6.2 Data Preparation for Cosine Similarity 8
6.3 Cosine Similarity 8
6.4 Data Preparation for Naïve Bayes 9
6.5 Naïve Bayes 9
7 Discussion 10
7.1 Accuracy Measures 10
7.2 Threshold Selection for Cosine Similarity 11
7.3 Dataset Splitting 12
8 Conclusion and Future Work 13
9 References 14

List of Tables

Table 1: Confusion Matrix for Cosine Similarity 9
Table 2: Confusion Matrix for Naive Bayes 10
Table 3: Accuracy comparison 11
Table 4: Threshold vs Accuracy Measures 11
Table 5: Training Set vs Accuracy Measures 12

2 ABSTRACT

The main objective is to determine the similarity between two sentences from different aspects. Based on the corpus received from Quora, word similarity between two sentences is determined using four different aspects. We determined that more measures are required to determine the efficiency than just accuracy. Experiments show that using Naïve Bayes to determine the similarity between two sentences is closer to the people's comprehension to the meaning of the sentence and gives a higher accuracy and efficiency as compared to a cosine similarity.

3 INTRODUCTION

A lot of different techniques and methods have been proposed to compare the similarity between two sentences which includes text mining, information extraction [1], automatic answering question [2, 3], text summarization [4], text classification and machine translation [5]. To deliver the similarity between two sentences more precisely, nowadays application require not only comparing the overall meaning of the sentences, but similarity between different parts of the sentences. For example, one of the question pair received was 'what is the status of stock market' and another question was 'what is the status of stock market in China' which accounts for two sentences with similar sentences but different meaning. From both sentences, we get an idea that the question revolves around status of stock market, however, if we read the complete sentence, we can say that they pose different questions, since one asks about the overall stock market status and other asks about the stock market status in China.

Detecting question pairs also poses several challenges. Because questions are shorter in length, some of the common methods used to detect similarity like n-grams are not that useful. The short length also means that there is a limit up to which the text can be analyzed unlike an article or a book. A semantic analysis may also be less useful here. Since, our primary objective is to find out sentences where the structure is different and the meaning is same, we have not used semantic analysis.

To simulate human's comprehension to a sentence and make sentence similarity more meaningful, we aim to find out the best technique for the purpose. For this purpose, we carried out sentence similarity using two techniques: cosine similarity and using a machine learning algorithm called Naïve Bayes.

The report is organized as follows. Within the next section we review at some similar works like this and in section 5, we will have a look at different techniques we have applied to carry out the experiment. This experiment has been carried out in python[1]. Within section 6 and 7 we will have a look at the results obtained.

[1] GitLab repository link: https://git.cs.dal.ca/courses/nlp-winter-2019/p-16

4 RELATED WORK

Several techniques have been proposed to find the similarity between two sentences, and in some of them the characteristics are obtained from text. For example, counting the frequency of words. Other techniques include getting a deeper understanding of the text and establishing relation between the text. [7, 8]

The approach on DLSITE-2 [9] applies a syntactic analysis to calculate the similarity between a phase and a real sentence that might be inside a sentence. The system builds a syntactic tree from sentences, trying to match them and evaluating the similarity between sentences. Using this approach, the less important words are eliminated, and time of execution is reduced.

Culotta and Sorensen [10] proposed a method to detect relations between sentences. The relations are classified by establishing connection between words. Each relation is instantiating a tree of dependencies. A dependency represents the grammatical dependency between two words and the algorithm adds more characteristics to each node of the tree. The similarity between two sentences in this work is determined by counting the number of common segments in two sentences.

One of the methods based on trees was proposed by Rui Wang [11], and it is capable to recognize textual links through the similarity between sentences. Textual links infers the meaning of a sentences inferred from another sentence. The method attempts to find connection between sentences through different linguistic layers. It is based on the fact that a hypothetic text is shorted than a common text, and not all the information is relevant to decide what is the connection between a text and a sentence.

Both semantic and syntactic information makes contribution to the meaning of a sentence. When comparing similarity between two sentences, many methods use semantic to compare the similarity. Mandreoli et al. [12] proposed a method adoption the Edit distance as a similarity measure between (parts of) sentences, and the method mainly pays attention to the similarity of syntax structure. Hatzivassiloglou et al. [13] presented a composite similarity metric over short passage which only utilize semantic information Mihalcea et al. [14] developed a method to score the sematic similarity of sentences by exploiting the information that can be drawn from the similarity of the component, but the syntax structure is ignored.

Palakorn et al. [15] tested different combinations of sentence vector similarity, word order similarity, POS similarity and considered question category similarity to measure the question similarities.

5 PROBLEM DEFINITION AND METHODOLOGY

In order to compare two sentences, we wanted to define the best method to carry out the comparison. For this purpose, we used various techniques like cosine similarity and machine learning algorithm I.e. Naïve Bayes to carry out the experiment. We tested them on various factors discussed in the next section.

Initially we stated out by cleaning the data. We used various techniques like removing non-alphabetic characters, removing stop words and stemming the words to their root. Once we had the clean data, we created a TF-IDF from the clean data set. We used this TF-IDF formed from the corpus as in input to the cosine similarity algorithm and Naïve Bayes algorithm. We received the required output and measured the success of an algorithm based on four factors which is discussed in experiment design section.

In figure 1, the block diagram describes the operation of the method proposed. Once we have selected a pair of sentences, the next step is to clean the data.

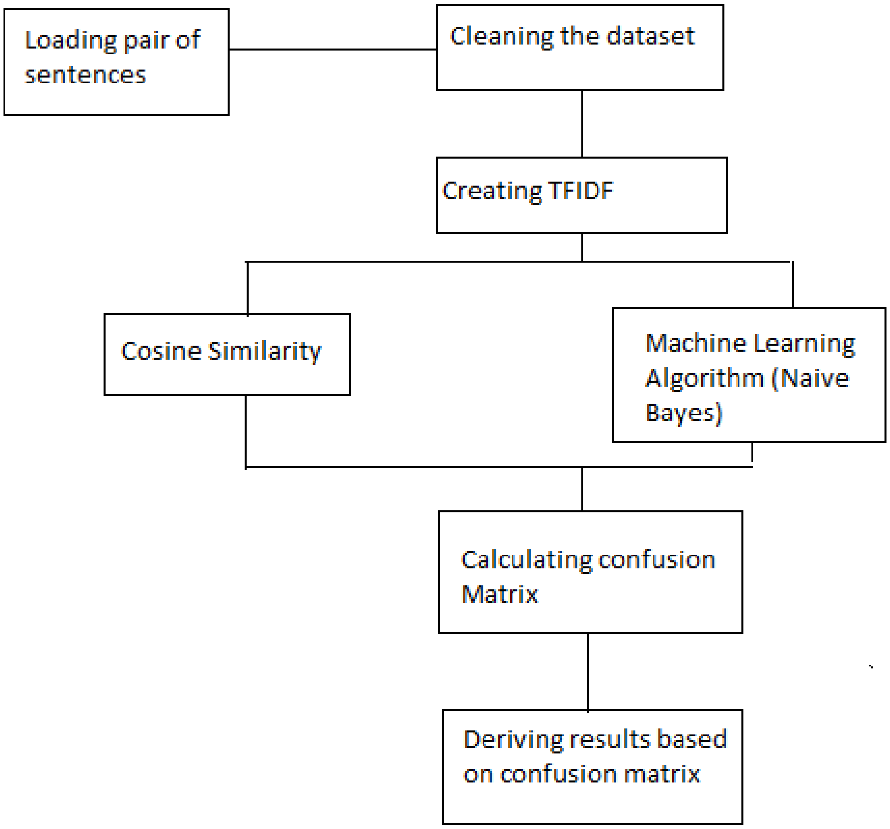
Loading pair of sentences
Cleaning the dataset
Creating TFIDF
Cosine Similarity
Machine Learning Algorithm (Naive Bayes)
Calculating confusion Matrix
Deriving results based on confusion matrix

Figure 1: Experiment Design Approach

Once we obtained the cleaned data set, we applied TFIDF to the pair of sentences. Further we used this as an input for the cosine similarity and machine learning algorithm to obtain the confusion matrix. We further used this confusion matrix to obtain various results described in next section.

5.1 Cosine Similarity

Similarity between two documents is derived by calculating cosine value between two documents' term vectors [16]. Implementation of this type can be applied to any two texts (sentences, paragraph, or whole document). Let x and y be two vectors for comparison. Using the cosine measure as a similarity function, we have

$$Sim(x, y) = \frac{x * y}{|x| * |y|}$$

where |x| is the Euclidean norm of vector x = (x1, x2, x3… xp). Conceptually, it is the length of the vector. Similarly, ||y|| is the Euclidean norm of vector y. The value computes the angle between vector x and vector y. A cosine value of 0 means that the two vectors are at 90 degrees to each other and closer the cosine value to 1, the smaller the angle and higher is the chance that both vectors match with each other.

Using the above technique for the question pairs from Quora data set, we were able to determine how similar the two questions were.

5.2 Machine Learning Algorithm (Naïve Bayes)

Naïve Bayes uses the Bayesian classification algorithm, which is based on the Bayesian theory, which is one of the classical statistical algorithms. Bayesian classifier is constructed from a training data set with class labels. Assuming n attributes A1, A2, … An instance E is represented by a vector <a1, a2, … , an> where ai is the value of Ai. C is used to represent the class variable; c is the value for C and c(E) denotes which class label E belongs to. So, Bayesian classifier is defined as [17]:

$$C(E) \ = \ arg\ max\ P(c)\ P(a1, a2, a3, \dots ., an)$$

Assuming all attributes are independent, we can deduce:

$$P(E|c) \ = \ P\ (a1, a2, a3, \dots, an \mid c)$$

$$P(E|c) \ = \ \Pi P(ai \mid c)$$

Using the above values, we get Naïve Bayes (NB) as follow:

$$C(E) \ = \ arg\ max\ P(c)\, \Pi P(ai \mid c)$$

Naïve Bayes has many advantages, such as efficient computation and very good performance on a dataset which is changing frequently. Within the Bayesian network, its node represents attribute and the arcs represent dependency. Within our experiment, Naïve Bayes classifies whether a question pair is similar or not using the Bayesian network and returns 1 for similar and 0 if the sentence pair is not similar.

In a probability inference task, our goal is to calculate the probability of a hypothesis C holds given condition that that the data has been observed A1, A2, A3,... An have been trained, which is

$$P(C \mid Ai, A2, A3, \ldots An).$$

We need to test our Naïve Bayes on a test data, for that Naïve Bayes calculates the joint probability using the following formulae.

$$P\ (A1 \mid C, Aj)\ =\ P\ (Ai \mid C)$$

Therefore, the joint probability can be rewritten as:

$$P(C, A1, \ldots\ , An)\ =\ P(C)\ \Pi P(Ai \mid C)$$

Using formula, we have:

$$P\ (\ C \mid A1, .., An)\ = 1/Z\ P(C)\ \Pi P(Ai \mid C),$$

where Z = P(A1, ..., An) is a constant during one given probability inference task.

Using the above technique, we were able to train our data set and test our model on question pairs to check similarity between sentences.

6 EXPERIMENT DESIGN

6.1 DATASET[2]

In our experiment, we use the dataset provided by Quora[3]. It contains five columns: id, question one id, question two id, question one, question two, and question duplicate indicator. The dataset is somewhat biased towards questions about India. It contains questions that either directly mention India or are set in that context. Due to this bias we suspect that the machine learning model we produce as a part of this experiment will contain these biases and will be more proficient in predicting similarities in India than the rest of the world. For the purposes of this experiment, the model will provide accurate results because we are creating the test data set from the training dataset.

6.2 DATA PREPARATION FOR COSINE SIMILARITY

The dataset needs to be prepared for both the methods that we use for this experiment: cosine similarity and Naïve Bayes. For preparing the data we use the methods described previously. First, we clean the dataset which includes removing stop words, stemming, and removing non-alphabetic characters. The next step is creating term frequency-inverse document frequency (tf-idf) to create document vectors for the data.

6.3 COSINE SIMILARITY

Cosine similarity is calculated after the data has been prepared for it. The cosine similarity algorithm described in the previous section is a bag of words model and gives some interesting results. Table 1 shows the confusion matrix achieved for thousand records of data for a similarity threshold value of 0.3. From this confusion matrix we derive and accuracy of 61.8%. We can also derive other evaluation measures which we will discuss in further sections. We will also describe the choice of threshold value in further sections.

[2] The datasets can be found at: https://dalu-my.sharepoint.com/:x:/g/personal/am768517_dal_ca/EUmtV9muI5ZHqnlkWAwaiBoBGmZb-Sv_bxsBXCD6G6rCnw?e=rLoEwp and https://dalu-my.sharepoint.com/:x:/g/personal/am768517_dal_ca/EdMKog1IxJlLgcRogyKdG5ABGIoBvhIH1MwDL7HI5gv-7w?e=ud1dkK

[3] Kaggle challenge link: https://www.kaggle.com/c/quora-question-pairs/overview

Table 1: Confusion Matrix for Cosine Similarity

		Actual	
		No	Yes
Predicted	No	615	377
	Yes	4	3

6.4 Data Preparation for Naïve Bayes

Data preparation for Naïve Bayes method requires some additional steps as compared to the cosine similarity method. The cosine similarity method required two arguments whereas the Naïve Bayes method requires only one argument. For this reason, the inputs have been combined for it. Another, area where data preparation is different is, we use a standard scalar after creating tf-idf. The standard scalar scales the data according to the machine learning method used so that there are no invalid values. For example, if there are negative values, all the values are scaled so that those negative values become positive. On applying these additional data cleaning methods, we are ready for applying Naïve Bayes method.

6.5 Naïve Bayes

Naïve Bayes method, described in the previous sections, is a bag of words model. Once we have the clean data, we need to spit the dataset into training dataset and test dataset. We use the training dataset to train the Naïve Bayes classifier. We use the test dataset to make predictions and get the confusion matrix shown in table 2. We found the accuracy of the classifier to be 67%. We do however, use more measures here as discussed in the further sections.

Table 2: Confusion Matrix for Naive Bayes

		Actual	
		No	Yes
Predicted	No	55	7
	Yes	26	12

7 Discussion

In this section, we discuss the results of our experiments and delve a little deeper into what they mean. We first start with our choice of the accuracy measures for this experiment.

7.1 Accuracy Measures

For the purpose of this experiment we have selected accuracy, F-measure, recall and precision as our accuracy measures. Table 3 shows a comparison between these measures between the two approaches used. This table also makes it apparent why just the measure of accuracy is not enough to measure the accuracy. As shown in the table, the accuracy of the algorithms is closer, but the other measures have a much greater difference. This clearly displays that the Naïve Bayes approach is much better than the cosine similarity approach, however, this cannot be inferred from accuracy alone.

Table 3: Accuracy comparison

	Cosine Similarity	Naïve Bayes
Accuracy	61.8%	67%
F-measure	0.02	0.42
Precision	0.43	0.32
Recall	0.008	0.63

7.2 THRESHOLD SELECTION FOR COSINE SIMILARITY

The threshold value for cosine similarity defines the boundary between similar and dissimilar documents. A low threshold value makes the algorithm biased towards matches and a high threshold value makes it biased towards dissimilar values. A balanced value is required but the selection of value depends on the data itself. Table 4 describes the relation between threshold values and the measures we have selected for this experiment. A decreasing threshold value decreases accuracy but increases F-measure. For the right value, we wanted to strike a balance between the three and hence a value of 0.3 was selected.

Table 4: Threshold vs Accuracy Measures

Threshold	Accuracy	Precision	Recall	F-measure
0.8	61.9%	0	0	0
0.6	61.9%	0.5	0.003	0.006
0.4	61.9%	0.5	0.005	0.01
0.3	61.8%	0.43	0.008	0.02
0.2	60.8%	0.26	0.016	0.03

7.3 DATASET SPLITTING

Splitting a dataset into test and training dataset requires quite a bit of deliberation. The training dataset needs to be large enough to have a variety of cases for each class in the output. It also should have an even number of records for each class. The test dataset needs to be large enough that it included data points on the boundary of the classes. Table 7 shows the relation between training dataset size and accuracy measures.

Table 5: Training Set vs Accuracy Measures

Train Set	Accuracy	Precision	Recall	F-measure
70%	62.7%	0.22	0.47	0.3
80%	63.5%	0.24	0.49	0.32
85%	65.3%	0.27	0.5	0.35
90%	67%	0.32	0.63	0.42
95%	72%	0.45	0.75	0.56

8 CONCLUSION AND FUTURE WORK

In the work we have done here, we use two methods for detecting duplicate questions. We compare these two approaches in depth using measures of accuracy, precision, recall, and f-measure. We found the accuracy of the Naïve Bayes classifier to be slightly more accurate than the cosine similarity approach. Looking at the confusion matrices for both approaches during our experiments led us to determine that accuracy alone is not the best measure for this task. We then experimented on quite a few measures and finally settled on f-measure, precision, and recall. We found that they, along with accuracy, provide a good measure for our experiment. Comparing the two approaches used, we found that Naïve Bayes has significantly better recall value than cosine similarity. Consequently, it also has a higher f-measure value. This led us to determine that Naïve Bayes is much better for this classification than cosine similarity.

For this experiment, we wanted to compare a machine learning approach for similarity detection with a traditional approach. The future step would logically be to compare various machine learning approaches with each other. Further expansion of this experiment can also be carried out by comparing machine learning methods with neural networks. Particularly, we would want to compare the gain in accuracy measures on the same dataset versus that on an expanded dataset which benefits deep learning methods more than traditional machine learning methods.

Beyond the methods for classification, we would like to expand the scope of this experiment by expanding the dataset to include more information than just the two questions. Taking more metadata into account for detecting similarity can greatly improve this process. We would like to include metadata like the time the question was posted, the location the question was posted from, and the tags associated with the question. We would also like to expand the dataset to include answers to questions as metadata. This would greatly improve the classifier as similar answers to questions would mean that the questions are also similar. Another interesting scope expansion would be to make the dataset more varied by including questions from various fields like technology, medicine, and science.

9 REFERENCES

[1] Poon, H., and Domingos, P., “Joint inference in information extraction”, Proceeding of the Twenty-Second AAAI Conference on Artificial Intelligence, pp. 913–918, 2007.

[2] Lin, D., and Pantel, P., “Discovery of inference rules for question answering”, Natural Language Engineering, Vol. 7, No. 2, 2001.

[3] Achananuparp P., Hu X., Xiaohua Zhou, Xiaodan Zhang, “Utilizing Sentence Similarity and Question Type Similarity to Response to Similar Questions in Knowledge-Sharing Community”, WWW 2008 Workshop on Question Answering on the Web, April 22, Beijing, China 2008.

[4] Erkan, G., and Radev, D., “Lexrank: Graph-based lexical centrality as salience in text summarization”, Journal of Artificial Intelligence Research Vol. 22, pp. 457–479, 2004.

[5] Ko, Y., Park, J., and Seo, J., “Improving text categorization using the importance of sentences”, Information Processingand Management Vol. 40, No.1, pp. 65–79, 2004.

[6] Papineni, K., Roukos, S., Ward, T., and Zhu, W., “Bleu: a method for automatic evaluation of machine translation”, Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318, 2002.

[7] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 990–998, Las Vegas, Nevada, USA, 2008. ACM.

[8] M.F. Reza and R. Matin. Application of data mining for identifying topics at the document level. In Informatics, Electronics Vision (ICIEV), 2013 International Conference on, pages 1–6, Dhaka, Bangladesh, 2013. IEEE

[9] Daniel Micol, Oscar Ferr ´ andez, Rafael Mu ´ noz, and Manuel Palomar. ˜ Dlsite-2: Semantic similarity based on syntactic dependency trees applied to textual entailment. In United States of America, pages 73–80, Rochester, NY, USA, 2007. Association for Computational Linguistics

[10] Daniel Micol, Oscar Ferr ´ andez, Rafael Mu ´ noz, and Manuel Palomar. ˜ Dlsite-2: Semantic similarity based on syntactic dependency trees applied to textual entailment. In United States of America, pages 73–80, Rochester, NY, USA, 2007. Association for Computational Linguistics.

[11] Rui Wang and Gunter Neumann. Recognizing textual entailment using ¨ sentence similarity based on dependency tree skeletons. In Proceedings of the ACL-PASCAL Workshop on Textual

Entailment and Paraphrasing, RTE ’07, pages 36–41, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[12] Mandreoli, F., Martoglia, R., and Tiberio, P., “A syntactic approach for searching similarities within sentences”, Proceeding of International Conference on Information and Knowledge Management, pp. 656–637, 2002.

[13] Hatzivassiloglou, V., Klavans, J., and Eskin, E., “Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning”, Proceeding of Empirical Methods in natural language processing and Very Large Corpora, 1999.

[14] Mihalcea, R., Corley, C., and Strapparava, C., “Corpus-based and knowledge-based measures of text semantic similarity” Proceeding of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, 2006

[15] Achananuparp P., Hu X., Xiaohua Zhou, Xiaodan Zhang, “Utilizing Sentence Similarity and Question Type Similarity to Response to Similar Questions in Knowledge-Sharing Community”, WWW 2008 Workshop on Question Answering on the Web, April 22, Beijing, China 2008.

[16] G. Salton and C. Buckley, “Term-weighting Approaches in Automatic Text Retrieval,” Information Processing and Management, vol.24, no.5, 1988, pp.513−523.

[17] Le Zhang, Jingbo Zhu, and Tianshun Yao, Natural Language Processing Laboratory Institute of Computer Software & Theory, Northeastern University An Evaluation of Statistical Spam filtering techniques, ACM Transactions on Asian Language Information Processing, Vol. 3, No. 4, December 2004, pages 243–269