A Question Pairs Similarity Detection With Data Mining Applications Using Natural Language Processing And Machine Learning: QUORA

B.Sankara Babu

Associate Professor Computer Science and Engineering Gokaraju Rangaraju Institute of Engineering and Technology

bsankarababu81@gmail.com

ABSTRACT

Data mining is the ability to extract the most relevant information from raw data and process the relevant information into a separate list. In general, the word ontology is termed an alternate name for data mining, in which extracting useful information from the big data sources is the most common task present in the ontology domain. In current days the semantic web is becoming a more trending topic for a lot of new research inventions. One of the recent research inventions is finding the most common question pair similarity from several questions and answers which are published on certain topics on the semantic web. This is becoming a challenge for the developers to prove this task and hence this is the main motivation for me to design this current work in which one will get a lot of useful information from this research topic. For testing this current model we assume QUORA as the sample data source in which a group of several question-and-answers is asked, answered, edited, and organized by its community of users. On this site lot of end, users can collaborate with each other for certain questions, share their suggestions, and edit the answers which are already submitted by others. This collaboration is taken as a thread on a single question with a list of similar/related questions so that users would not have to answer similar questions once again. By using the several various Natural Language Processing (NLP) concepts from the given dataset and applying several ML algorithms on that input features, we try to fund our most distinct question and answers and remove the duplicates which are present for a certain topic.

Key Words:

Ontology, Machine Learning, Natural Language Processing, QUORA, Distinct Questions, Duplicate Questions, Semantic Web, Research Inventions.

1. INTRODUCTION

Now a days there is a huge increase of text data present on the world wide web. Hence in order to manage those valuable information and process the information, we need some best algorithms which can outperform than other algorithms in order to retrieve all the relevant information from the data repositories. In general, mining is considered as only process which is used to retrieve any sort of information from structured or relational data and there is some sort of confusion for the end users while choosing the algorithm, Most of the users are confused in which algorithm to be chosed and extract the most related information. As we all know that if any user want to extract any text information from structured or question and answers or from large paragraphs with relational data, we have several methods like clustering, visualization, classification and summarization..

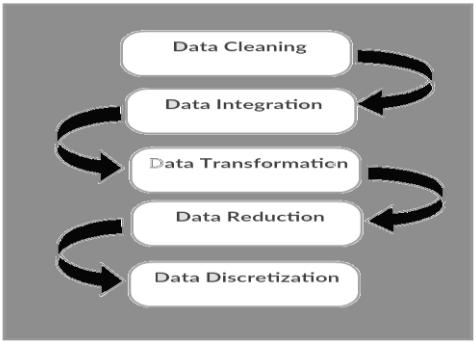


Figure 1. Represents the Data Pre-Processing Steps

For any data mining application, data pre-processing is considered as essential step for converting the high level instructions into machine level instructions. If the data pre-processing is not applied on the input data, it would be very inconsistent and the model could not generate accurate results.

2. LITERATURE SURVEY

Literature survey is that the most vital step in the software development process. Before developing the new application or model, it's necessary to work out the time factor, economy, and company strength. Once all these factors are confirmed and got approval then we can start building the application.

MOTIVATION

1) Automatic Generation of Multi-Modal Dialogue from Text Based on Discourse Structure Analysis.

AUTHORS: H. Prendinger

In this paper the author mainly try to concentrate on how to classify the question and answers from a text or passage which is taken as input. In general this text can be classified in two ways:

a) Dialogue and interactive Q/A system

b) Education assessment.

2) Experiments with Interactive Question-Answering".

AUTHORS: D. Moldovan

In this paper the author mainly concentrated on the interactive question and answer system for education related domain. Here the authors try to generate the question and answers based on the individual user wish and try to compare the same question and answers with some other paper which tries to identify the domain knowledge first and then find out the context related to that domain.

3) Automatic Question Generation from Text – an Aid to Independent Study SIGCUE Outlook.

AUTHORS: J. H. Wolf

In this proposed work the author discussed the difficulty while generating the question and answers for given context. For example the author compared the current task with some common example like create an exam paper is a very complex task and the generation of MCQs for the educational system is very time-consuming and hence the instructors require a lot of workload in order to generate appropriate question and answers for that related domain.

4) An Evaluation of Preprocessing Techniques for Text Classification

AUTHORS: Ammar Kadhim et al.

In this proposed work the author mainly discussed about the importance of text data pre-processing and how this plays a vital role in the text classification applications. The author described that text pre-processing is the ability to reduce multiple forms of words into single word and extract the main features present in that word.

3. PROPOSED DATASET

In this proposed work, we try to use Quora dataset which is globally released. In this dataset we can get more than 4 lakhs question pair for training the model and a Testing dataset of 23 lac question pair. The following are the some of the main attributes present in that dataset.

Data fields available in the dataset:

a) id - the id of a training set question pair

b) qid1, qid2 - unique ids of each question (only available in train.csv)

c) question1, question2 - the full text of each question

d) is_duplicate - the target variable, set

Now let us consider small examples on similar question pairs and dis-similar question pairs.

SIMILAR QUESTION PAIRS

How can I be a good geologist?
What should I do to be a great geologist?

DISSIMILAR QUESTION PAIRS

What is the step by step guide to invest in share market in India?
What is the step by step guide to invest in share market?

4. PROPOSED METHODOLOGY

In this section we try to discuss about the proposed algorithms which are used for showing the performance of our current objective. In order to prove the performance of our current application, we try to divide the methodology into two phases:

1) Feature Extraction Using NLP

In this NLP we can extract the features in two ways:

A) Basic Feature Set

B) Fuzzy Feature Set

2) Similarity prediction using machine learning model

In this level we try to compare the current application with several ML models and then check performance of all the models and decide which model gives best accurate result out of several models. In our current application XGBOOST gives best accuracy in order to predict the best similarity question and pair.

1) Feature Extraction Using NLP

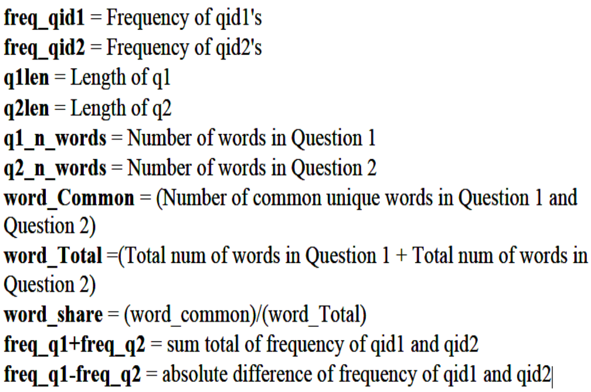
In this NLP we can extract the features in two ways:

A) Basic Feature Set

B) Fuzzy Feature Set

A) Basic Feature Set

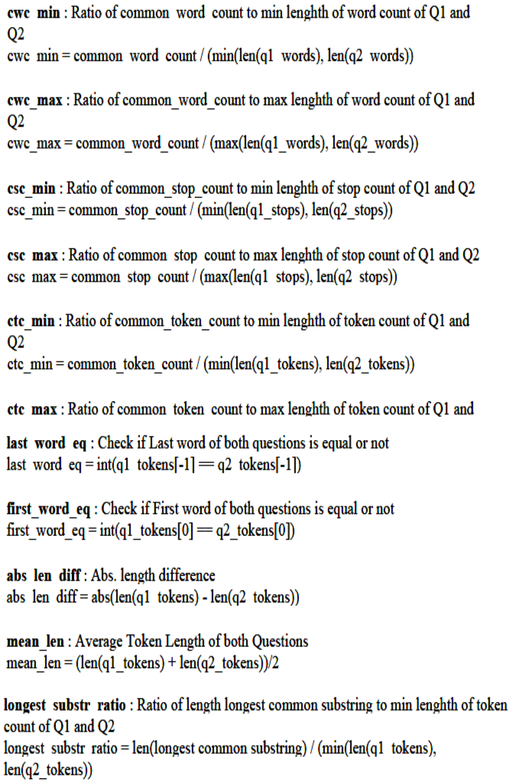
The following are the some of the attributes of basic feature set,



freq_qid1 = Frequency of qid1's
freq_qid2 = Frequency of qid2's
q1len = Length of q1
q2len = Length of q2
q1_n_words = Number of words in Question 1
q2_n_words = Number of words in Question 2
word_Common = (Number of common unique words in Question 1 and Question 2)
word_Total =(Total num of words in Question 1 + Total num of words in Question 2)
word_share = (word_common)/(word_Total)
freq_q1+freq_q2 = sum total of frequency of qid1 and qid2
freq_q1-freq_q2 = absolute difference of frequency of qid1 and qid2

From the above features set we can see there are nearly 11 distinct features present in that basic feature set.Each and every feature is having distinct importance and if we come with word_Total and Word_Common, they both are main features present in the basic feature set.

B) Fuzzy Feature Set



cwc min : Ratio of common word count to min lenghth of word count of Q1 and Q2
cwc min = common word count / (min(len(q1 words), len(q2 words))

cwc_max : Ratio of common_word_count to max lenghth of word count of Q1 and Q2
cwc_max = common_word_count / (max(len(q1_words), len(q2_words))

csc_min : Ratio of common_stop_count to min lenghth of stop count of Q1 and Q2
csc_min = common_stop_count / (min(len(q1_stops), len(q2_stops))

csc max : Ratio of common stop count to max lenghth of stop count of Q1 and Q2
csc max = common stop count / (max(len(q1 stops), len(q2 stops))

ctc_min : Ratio of common_token_count to min lenghth of token count of Q1 and Q2
ctc_min = common_token_count / (min(len(q1_tokens), len(q2_tokens))

ctc max : Ratio of common token count to max lenghth of token count of Q1 and

last word eq : Check if Last word of both questions is equal or not
last word eq = int(q1 tokens[-1] == q2 tokens[-1])

first_word_eq : Check if First word of both questions is equal or not
first_word_eq = int(q1_tokens[0] == q2_tokens[0])

abs len diff : Abs. length difference
abs len diff = abs(len(q1 tokens) - len(q2 tokens))

mean_len : Average Token Length of both Questions
mean_len = (len(q1_tokens) + len(q2_tokens))/2

longest substr ratio : Ratio of length longest common substring to min lenghth of token count of Q1 and Q2
longest substr ratio = len(longest common substring) / (min(len(q1 tokens), len(q2_tokens))

From the above fuzzy features set we can see there are nearly 11 distinct features present in that basic feature set.Each and every feature is having distinct importance and they are used to calculate the most common question answer pair's similarity.

2) Similarity Prediction Using Machine Learning Model

In this current application we try to use Xgboost a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. This Xgboost is involved mainly in extracting the unstructured data (images, text, etc.) to outperform all other algorithms or frameworks. This is considered as best algorithm by most of the users because it comes to small-to-medium structured/tabular data, and this decision tree is considered as best in all aspects.

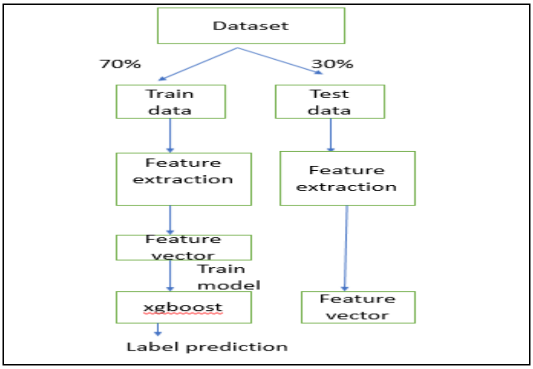


Figure 2. Represents the Architecture of our Proposed Model

STEP WISE EXPLANATION

The following is the step wise procedure of our current model. This is as follows:

Step 1: Initially we try to load the input dataset which contains a lot of valuable information on certain topic. Here the data set is Quora dataset which contains a lot of question and answers pairs on certain topic, which is unstructured.

Step 2: Now we need to apply pre-processing technique on the given dataset and then divide the whole dataset into test and train. Here we take 70 percent of data as train data and 30 percent of data as test data.

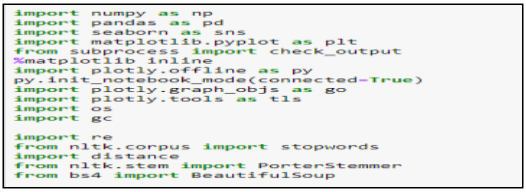
Step 3: In this stage we try to extract the features from both test and train data and then form a feature vector from that given input data.Now once feature vector is formed ,we try to calculate the main features and then train the system with those features.

Step 4: Now we apply XgBoost ML Model and then try to classify which queries are distinct and which are duplicated. Based on that query result we try to give labeling for those input queries and then classify the best desired result.

5. EXPERIMENTAL RESULTS

For showing the performance of our proposed application, we try to deploy the current application using Python as programming language. First we will import all the necessary libraries and then load the input dataset to find out the most accurate similarity between question and answer pairs.

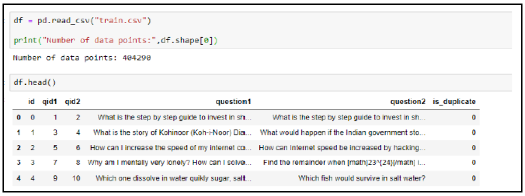
IMPORT LIBRARIES



```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from subprocess import check_output
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import os
import gc

import re
from nltk.corpus import stopwords
import distance
from nltk.stem import PorterStemmer
from bs4 import BeautifulSoup
```

In the above window we can clearly see there are several libraries and packages used to prove the current objective. Hence we try to load all those necessary libraries and import them into our application.

LOAD INPUT DATASET



```
df = pd.read_csv("train.csv")

print("Number of data points:",df.shape[0])
```

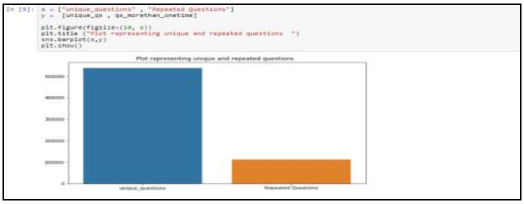
```
Number of data points: 404290
```

```
df.head()
```

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

In the above window we can clearly see there are some important fields like data points, number of datapoints and necessary question pairs and its corresponding result whether they are distinct or duplicated. Here for is_duplicate attribute we have two values i.e 0 or 1. If the questions are similar and duplicated with answers then it is marked as ‘1’. If they are not duplicated then marked with ‘0’.

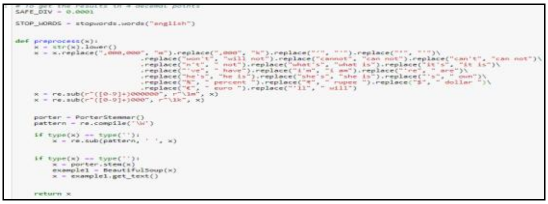
DATA VISUALIZATION



```
In [9]: x = ["unique_questions" , "Repeated Questions"]
        y =  [unique_qs , qs_morethan_onetime]

        plt.figure(figsize=(10, 6))
        plt.title ("Plot representing unique and repeated questions  ")
        sns.barplot(x,y)
        plt.show()
```

From the above window we can clearly identify data visualization in chart manner with two main parameters such as: unique questions and duplicated questions.

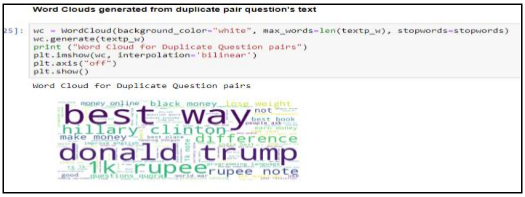
DATA PRE-PROCESSING



```
# To get the results in 4 decemal points
SAFE_DIV = 0.0001

STOP_WORDS = stopwords.words("english")


def preprocess(x):
    x = str(x).lower()
    x = x.replace(",000,000", "m").replace(",000", "k").replace("′", "'").replace("’", "'")\
                           .replace("won't", "will not").replace("cannot", "can not").replace("can't", "can not")\
                           .replace("n't", " not").replace("what's", "what is").replace("it's", "it is")\
                           .replace("'ve", " have").replace("i'm", "i am").replace("'re", " are")\
                           .replace("he's", "he is").replace("she's", "she is").replace("'s", " own")\
                           .replace("%", " percent ").replace("₹", " rupee ").replace("$", " dollar ")\
                           .replace("€", " euro ").replace("'ll", " will")
    x = re.sub(r"([0-9]+)000000", r"\1m", x)
    x = re.sub(r"([0-9]+)000", r"\1k", x)


    porter = PorterStemmer()
    pattern = re.compile('\W')

    if type(x) == type(''):
        x = re.sub(pattern, ' ', x)


    if type(x) == type(''):
        x = porter.stem(x)
        example1 = BeautifulSoup(x)
        x = example1.get_text()


    return x
```

From the above window we can clearly identify several pre-processing methods while extracting the features from the input data.

Word Cloud Formed by Duplicate Pair Question

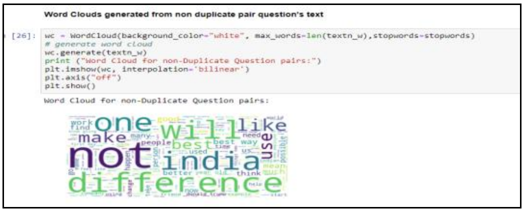


Word Clouds generated from duplicate pair question's text

```
25]: wc = WordCloud(background_color="white", max_words=len(textp_w), stopwords=stopwords)
     wc.generate(textp_w)
     print ("Word Cloud for Duplicate Question pairs")
     plt.imshow(wc, interpolation='bilinear')
     plt.axis("off")
     plt.show()

     Word Cloud for Duplicate Question pairs
```

From the above window we can clearly identify several words which are extracted from most duplicated question pairs and hence they all are formed as one word cloud from duplicated pair questions.

Word Cloud Formed by Non-Duplicate Pair Question

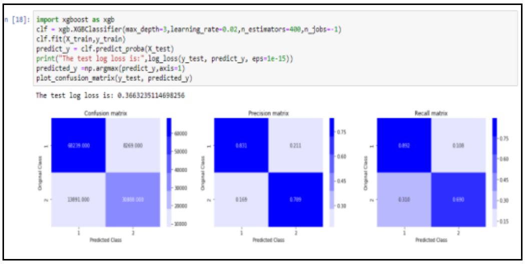


Word Clouds generated from non duplicate pair question's text

```
[26]: wc = WordCloud(background_color="white", max_words=len(textn_w),stopwords=stopwords)
      # generate word cloud
      wc.generate(textn_w)
      print ("Word Cloud for non-Duplicate Question pairs:")
      plt.imshow(wc, interpolation='bilinear')
      plt.axis("off")
      plt.show()

      Word Cloud for non-Duplicate Question pairs:
```

From the above window we can clearly identify several words which are extracted from most non-duplicated question pairs and hence they all are formed as one word cloud from non-duplicated pair questions.

XgBoost Performance



```
n [18]: import xgboost as xgb
        clf = xgb.XGBClassifier(max_depth=3,learning_rate=0.02,n_estimators=400,n_jobs=-1)
        clf.fit(X_train,y_train)
        predict_y = clf.predict_proba(X_test)
        print("The test log loss is:",log_loss(y_test, predict_y, eps=1e-15))
        predicted_y =np.argmax(predict_y,axis=1)
        plot_confusion_matrix(y_test, predicted_y)

        The test log loss is: 0.3663235114698256
```

From the above window we can clearly identify confusion matrix generated for the given dataset by using XgBoost ML algorithm. The performance is measured and represented in three charts such as: Confusion Matrix, Precision Matrix, Recall matrix.

6. CONCLUSION

In this proposed work, we for the first time developed a novel methodology of finding the most common question and answer pair similarity on Quora dataset. For this I have tested a large number and variety of machine learning models to solve the duplicate question problem posed by the Quora dataset. By performing several theoretical and experimental analyses, I finally got the best performing from XGBoost Model. In general I strongly believe the Quora dataset is a useful resource to further explore the task of Natural Language Understanding with machine learning techniques and for sure I will extend my research on some more new models and try to increase the performance of other ML models.

7. REFERENCES

1.Mora, H., Ferrández, A., Gil, D., & Peral, J. (2009). International Review of Research in Open and Distributed Learning. Open educational resources: New possibilities for change and sustainability, 10, 5.

2.Zillman, M. P. (2005). Academic and scholar search engines and sources.

3. Brace-Govan, J. (2003). A method to track discussion forum activity: the Moderators' matrix. Internet and Higher Education, 6, 303-325.

4.Cade, W. L., Copeland, J. L., Person, N. K., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. Proceedings of the 9 th International Conference on Intelligent Tutoring Systems. Berlin: Springer-Verlag, 470-479.

5.Allan, J., Carbonell, J., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. Landsdowne, VA.

6.Allan, J., Harding, S., Fisher, D., Bolivar, A., Guzman-Lara, S., & Amstutz, P. (2005). Taking topic detection from evaluation to practice. Annual Hawaii International Conference on System Sciences - Track 4 – 04, 1-10.

7.De Laat, M., Lally, V., Lipponen, L., & Simons, R.-J. (2007), Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for Social Network Analysis. International Journal of Computer-Supported Collaborative Learning, 2(1), 87-103.

8.Dennen, V. P. (2008). Looking for evidence in learning: Assessment and analysis methods for online discourse. Computers in Human Behavior, 24, 205-219.

9.De Wever, B., Schellens, T., Valcke, M., & van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. Computers & Education, 46, 6-28.

10.D'Mello, S., Olney, A., & Person, N., (2010). Mining collaborative patterns in tutorial dialogues. Journal of Educational Data Mining, 1, 1–37. 11.

11. Erlin, N. Y. & Rahman, A. A. (2009). Students' interactions in online asynchronous discussion forum: A social network analysis. International Conference on Education Technology and Computer, 25-29.