UNIVERSITÀ DEGLI STUDI DI PISA

CORSO DI LAUREA MAGISTRALE IN

DATA SCIENCE AND BUSINESS INFORMATICS

TEXT ANALYTICS

PROF. ANDREA ESULI

QUORA QUESTION PAIRS

Luftjan Saliaj 606507

Marco Ciompi 537856

Francesco Salerno 534622

21/9/2021

Anno Accademico 2020/2021

INTRODUCTION

Sentences or questions similarity is of the utmost importance in many applications and machine learning approaches have applied it to solve this problem in many fields. Questions similarity is very hot topic these days and many researchers are approaching the problem to solve it with a reasonable accuracy.
In this project, we deep dive into it with different algorithms/experiments to find the optimal solution t the problem along with the comparison of multiple models.
The dataset used for this project is obtained from the Kaggle competition “Quora Question Pairs” and it is openly available on the website Kaggle.com. In our approach we implemented many classical machine learning models like logistic regressions, random forest and so on. Subsequently we configured BERT transformers with different model type.
To run the code which we enclose in the submission, we used the Kaggle platform in order to get the most out of GPU Accelerator.
However as is clear from the code related to the file submission some of the task which we will present have been run on a little portion of the dataset due to problem related with RAM limitation and computer efficiency.

https://www.kaggle.com/fsalerno4/text-analytics-project

1. Exploratory Data Analysis

1.1 Data Definition

The dataset used in this project is acquired from the Kaggle platform which mainly consist of three files: Train, Test, Submission. For our experiments we used train dataset which we split for training and testing purpose to generate results (80/20). Brief insight of the dataset of the training file shown in fig. 1 below.

Dataset Statistics

Number of Variables	6
Number of Rows	404290
Missing Cells	3
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	103.9 MB
Average Row Size in Memory	269.5 B
Variable Types	Numerical: 3 Categorical: 3

Dataset Insights

`id` is uniformly distributed	Uniform
`qid1` and `qid2` have similar distributions	Similar Distribution
`qid2` is skewed	Skewed
`question1` has a high cardinality: 290456 distinct values	High Cardinality
`question2` has a high cardinality: 299174 distinct values	High Cardinality
`is_duplicate` has constant length 1	Constant Length

Figure 1: Training Dataset insights

1.2 Data Distribution and Statistics

The goal of this project is to find the similarity (meaning or intent of the questions) of the two questions which are available in the dataset to classify whether the two questions are similar or not. First, we performed the exploratory data analysis of the data to get some insights.
In the dataset, the is_duplicate column are the labels that are the required target. The attribute is_duplicate assumes value 0 when the questions are not similar and 1 when they are similar. Fig. 2 shows the distribution of the values.

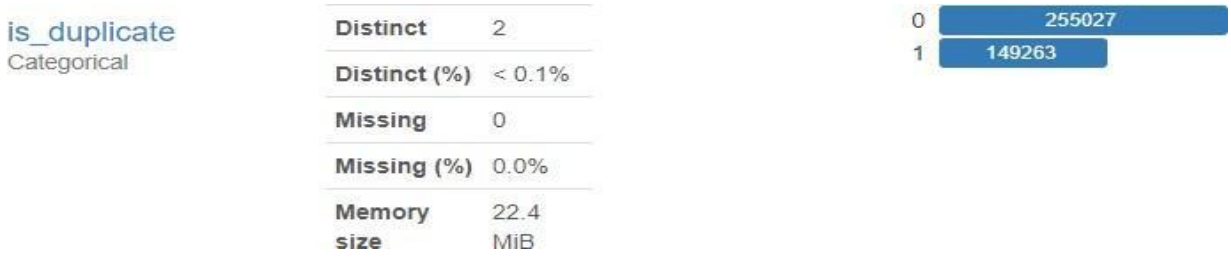

Figure 2: Distribution of the values

The distribution of labels columns is clearly shown in fig. 3 below, here 36.9% questions are similar while 63.1% are not similar. As the distribution is not uniformly distributed and hence the evaluation can be measured with respect of non-uniform distributions.

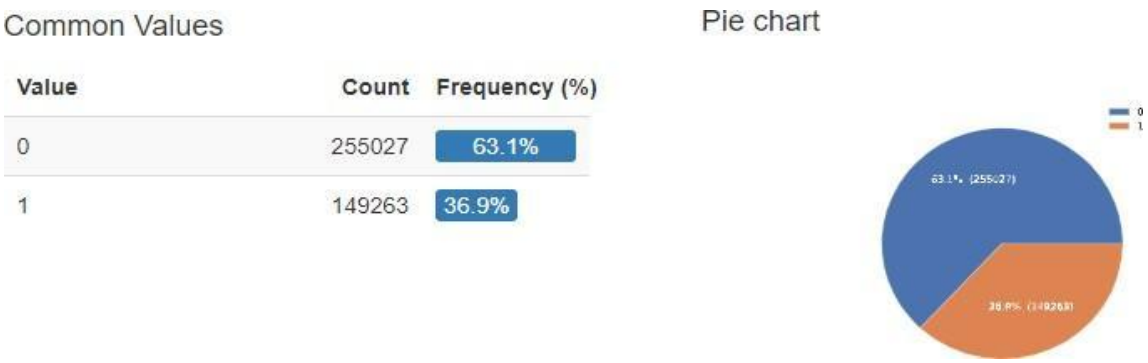

Figure 3: Distribution of labels

After analyzing the labels, we've individually analyzed the question1 column and question2 columns and the statistics are shown in Fig. 4 and Fig. 5 respectively. Distinct instances percentage is 71.8% and there is only 1 missing entity.

As shown in the figures above, the most repeated sentence in question one is “How do I improve my English speaking? “, while in question2 the most repeated question is “How can you look at someone's private…“
Figure 4: Question2 column Statistics

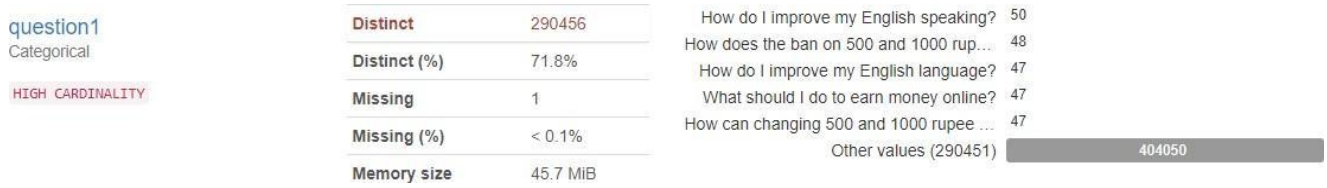

Figure 5: Question1 column statistics

The words shown in the fig. 6 indicate the most repeated words and it is clearly shown that the most repeated words are how, what, why respectively, which are dominant in the figure. Those word clouds are the original ones without any modifications.
The word cloud of the question1 is shown in Fig. 6:

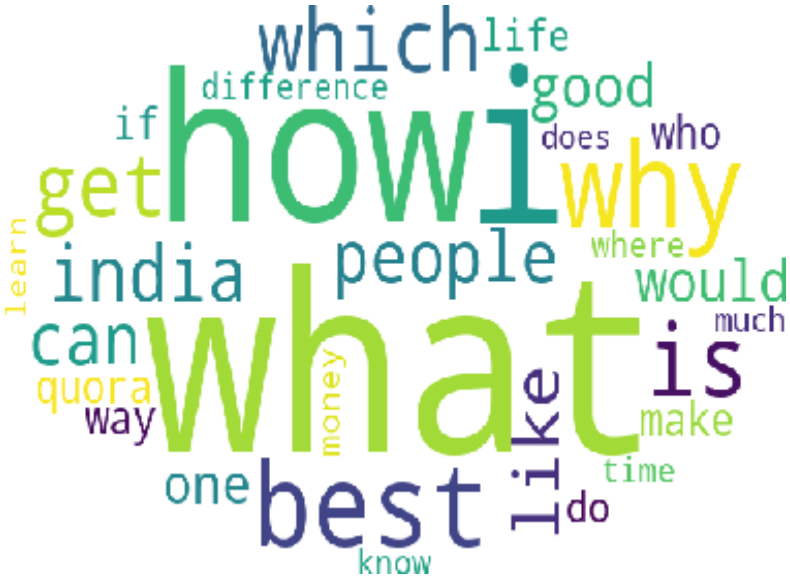

Figure 6: Word Cloud of Question1 Column

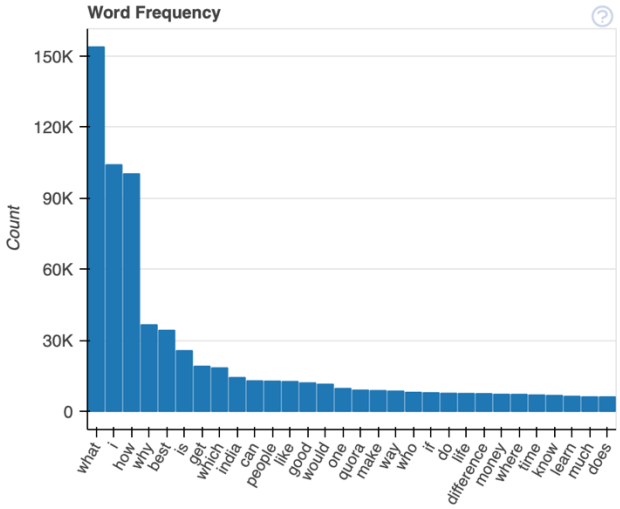

Figure 7: Word Frequency of Question1

The word frequency is shown in Fig. 7
In the figure all the words are shown with the count appeared int the questions and it is clearly shown that the word “what” is appeared almost 150k while I how and why also appeared in majority. Some words appeared uniquely and that basically
describe the similarity with ease.

Word cloud of question2 column is shown in Fig. 8. Here same like question1 the most repeated words are What, I, how, why. In the figure below there is always a split of words occurrence with the most repeated to the left side and on the right side is showing the minority one’s.

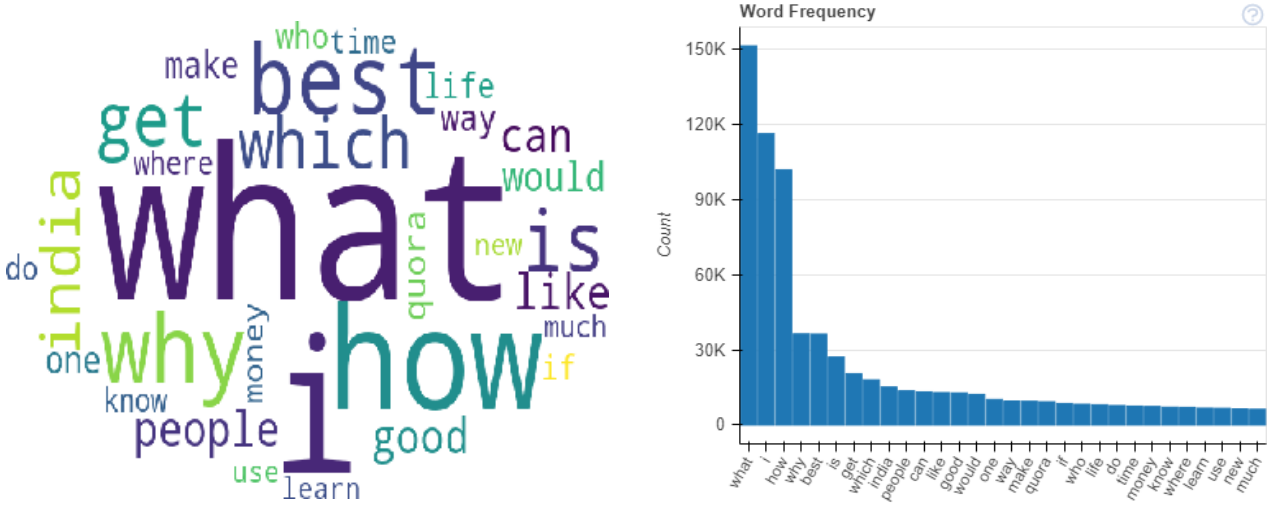

Figure 8: Question2 Column Word Cloud & Word Frequency

2. Data Common Preprocessing

We applied the following processes both to test set and training set.
However we clarify that some of the tasks such as the previous data Understanding have been done over the original data.
In order to preprocess the dataset, we removed alphanumeric characters by using regular expressions. Moreover, we converted text to lower cases and removed punctuations:

“!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~”.

Subsequently, stop words have been removed to avoid irrelevant redundancy in the data, except for the tag questions (respectively: What, Why, How, Where, Which ….) that have been included in the model training.

For the purpose of Natural Language Processing in the preprocessing steps we used nltk library.
For cleaning the data from non alphanumerical string we used the regular expression functions form the re library
After that using the function WordLemmatizer form nltk we lemmatized the words, which where not present in the set composed by all stopword from where we subtracted the questions tag

We then created a string containing all the values of the attribute "joined", removing all capital letters and punctuation to obtain some information about the text that we were going to use for classification.

NLP ANALYSIS

Before starting with nlp analysis we were needed to create a text where we list all the words to carry out with word tokenization.
For this purpose what we have done is to create a new column called “combine” where we put together the text derived from the question1 and question2 columns.
After some text/string process we obtained a text with all words.
Firstly, we created this string into tokens to figure out the total size of the text and the vocabulary used, resulting in 4823054 tokens,
and 211164 words, respectively.

We also counted the frequency of these words, respectively the first sentence present this frequency count:

[('what', 177668),
('step', 1424),
('guide', 315),
('invest', 1573),
or in general as was easy to get the most frequent are the question tag and som ether stop words which differently

from spacy are not present in nltk library like for example people.

[('what', 177668),
 ('how', 126190),
 ('best', 70109),
 ('why', 46357),

We also get the tokens which compare close to each other within each sentence

[(('best', 'way'), 11491),
 (('donald', 'trump'), 5294),
 (('year', 'old'), 3572),
 (('best', 'book'), 3201),

Our analysis proceeded with the topic modeling algorithm set to obtain 5 topics and the 10 most important words for each topic, the results we obtained are:

Topic 0: best, india, way, money, movie, online, get, make, english, best way
Topic 1: like, quora, question, thing, world, girl, people, know, love, first
Topic 2: indian, trump, job, note, work, india, 500, 1000, donald, donald trump
Topic 3: people, life, difference, one, become, better, get, someone, use, instagram
Topic 4: best, get, good, learn, book, start, make, way, language, weight

In the next phase which was related to word embedding we used the sent_tokenize methods to extract the context and word_tokenize to experiment on the most similar words based on those we provided through the skip gram methodology (we give a word as input and find the context). The algorithms used were Word2Vec and FastText. In the first case we provided as input the words:The choice of words given as input was made then following the frequencies of this words such as “stock” which appear a lot of times but also following our idea of what could be interesting to investigate.

3. Data Modelling

3.1 Classical Machine Learning Algorithms

For the purpose of modeling, different models have been implemented such as Logistic Regressions, k-Nearest Neighbors and Random Forest
Firstly we decided to train this model on a little portion of the dataset and also on all the train dataset.As we have our data in the text format and machine learning model need numerical data, a transformation is required on the textual data to be converted to numerical data.
Before doing that we splitted the training set in train and test using training_test_split setting the test percentage to 0,20. After that we started implementing a model which was able to fit and transform our text data composed by question 1 and question 2 columns. We used the function CountVectorizer setting the min_df parameter to 5. Also we did that over the test set. After that we were able to get also the features names. Moreover we continued doing the features selection using the function SelectKBest trained over the train and applied to both training and test. For weighting we used the TfidfTransformer functions and repeating the same process as before training the model over the train and applying it also to train and test set. After that we present the result obtained by the three classifier we mentioned above.

	Model Name	Accuracy	Precision	Recall	F1_score
0	KNeighborsClassifier	0.69100	0.701047	0.69100	0.678903
1	LogisticRegression	0.74045	0.736037	0.74045	0.707119
2	RandomForestClassifier	0.76900	0.770509	0.76900	0.734495

Table 2 Classical Models evaluation results

From the above evaluation results, it's clear that random forest performs well with 76.69% accuracy which is a good result compared to the others. Also, the f1-score of random forest shows its dominancy over the others.

3.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT is the bidirectional encoder representation from transformers and it is very powerful in the field of natural language processing these days. BERT models are known as state-of-the-art models due to it's pretrained models. BERT models used attention mechanisms which learn contextual relations in the textual data.
Sentences are encoded with the BERT tokenizer, which encode all the tokens to the input ids inserting special tokens as given below.

- Original Question: Method to find separation of slits using Fresnel biprism?
- Encoded: [101, 20569, 1106, 1525, 8865, 1104, 19884, 1116, 1606, 175, 4894, 8967, 16516, 1643, 18502, 136, 102]
- Decoded: [CLS] Method to find separation of slits using Fresnel biprism? [SEP]

For implementing BERT, simpletransformers library is used due to its simplicity and easy implementation approaches. In this project BERT, ROBERTA, DistilBERT, ALBERT have been implemented after the preprocessing of the data. Data is fed to these models for getting the pattern in the data with the BERT mechanism. The models mentioned above are fine tuned with the processed data and the models are evaluated with binary cross entropy as loss function. These models are then evaluated on accuracy, precision, recall and F1 Score accordingly. After the evaluation of the different models, fine tuning has been done on the models for achieving the following results
Overview of the model's evaluation is shown in Table 3 below.

	Model Type	Model Name	Accuracy	Precision	Recall	F1_score
0	roberta	roberta-base	0.820660	0.821383	0.820660	0.806458
1	bert	bert-base-cased	0.791646	0.795779	0.791646	0.777711
2	distilbert	distilbert-base-uncased	0.767884	0.768024	0.767884	0.748724
3	albert	albert-base-v2	0.771636	0.770407	0.771636	0.751464

Table 3: BERT Models Evaluation Results

Table 3 shows the evaluation results which in comparison with the classical models performed very well and the results are quite promising.. To compare the above model types Roberta performed well and the accuracy is 80.06%, the other metrics (precision, recall and F1 score) are also showing the dominancy over the rest of the

models. Model type BERT is also in competition with the Roberta and the accuracy is almost 80%. So, the overall performances of these models are over the line. There is always room to improve any machine learning models, so these models can be further optimized by tuning the hyper- parameters further.

CONCLUSION

In our approach for handling the task of questions similarity we implemented different models, both classical and deep neural networks. Multiple experiments have been performed and the evaluation results of the outcome is satisfactory. With post analyzing the metrics it has been concluded that the BERT models outperform the classical models. The main challenges that have been faced is the computational time and training time of BERT models, and also high machine specifications are required for achieving results with no latency. With the usage of GPU's, the model's performance in terms of computations increased. Overall, the results achieved with this project are satisfactory and further enhancement is possible with fine tuning the hyper- parameters.
We are open to think that might probably be some better possibilities to manage with preprocessing textual data.
We were not able to get the submission over the kaggle platform due to the fact that the ram offered by the website was not sufficient.

We are continuing our analysis in kaggle platform.

Latest version available on the following link:
https://www.kaggle.com/fsalerno4/text-analytics-project