Balochistan University of Information Technology, Engineering Management Sciences

Machine Learning

CEP Project

PROJECT TITLE:	DUPLICATE QUESTION RECOGNITION
PROJECT CATEGORY:	Natural language processing problem
DOMAIN:	Application project
ALGORITHM:	Decision Tree

Submitted to:

Ms.Saniya Ashraf

Machine Learning Lecturer,

BUITEMS, Department of Software Engineering

MEMBERS:

LAIBA KHAWAJA (53907)

MALAIKA NOOR (52900)

AHMED YOUNAS (53137)

Contents

DUPLICATE QUESTION RECOGNITION FOR QUORA

ABSTRACT

Duplicate detection is an important task in data cleaning that aims to identify multiple representations of the same real-world object. One way to achieve this is by combining decision tree and fuzzy similarity matching, which can increase the accuracy of duplicate detection or improve the efficiency of the computations.

KEYWORDS

Quora, Duplicate question, Machine learning, model, Random forest, XGBoost, fuzzy matching.

INTRODUCTION

Quora is a question-and-answer website that encourages its users to exchange information, voice their opinions, and demonstrate their subject-matter expertise on a range of subjects, it has 400,000 distinct topics and domain experts as its users, allowing consumers to acquire knowledge directly from the subject matter experts.

With the knowledge base's expanding repository, Quora must maintain user confidence and content quality by removing irrelevant, redundant, and dishonest material. In order to prevent question repetition.

MOTIVATION

The goal is to identify duplication of questions. It is an interesting use case of NLP. NLP is a field of study that focuses on the interactions between human language and computers. It allows computers to understand, interpret, and generate human language.

One specific NLP technique Quora could use is text similarity detection. This technique uses algorithms to compare the text of different questions to determine if they are similar or identical. By identifying and flagging duplicate questions, Quora can prevent question repetition and ensure that users are only seeing unique and highquality content.

Example.

1. What is the capital of Pakistan?

And

1- Is Islamabad capital of Pakistan?

Such Duplicate questions can be merged to improve the user experience.

INTENDED EXPERIMENTS

Q1: How to detect duplicate questions on Quora using machine learning?

Q2: How to achieve the best possible prediction results?

OUR APPROACH

We used Machine Learning. The problem can be solved using deep learning as well. The dataset is large we extracted features from the dataset and used feature engineering, tested methods. **In the conducted competition the maximum accuracy achieved was around 80% to 85% whereas our project has achieved an accuracy of 78% which is a better approach.**

THE DATASET

Data collection

The data used is Quora Dataset Released on Kaggle for a competition.

The dataset has a total of 404290 rows, corresponding to 404290 question pairs, and it is 55.4 MB in size overall. The data was in a file named "train.csv." The dataset was collected from:

https://www.kaggle.com/c/quora-question-pairs.

Data Shape:

We analyzed the dataset and found that it contains 404290 Rows and six columns that provide information about the duplicate questions. The columns are as:

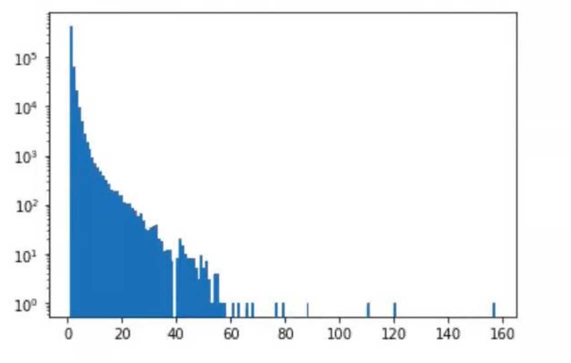
Column Name	Description
id	Identifier to rows
qid	Identifier to question1 column
qid2	Identifier to question2 column

question1	Question to be compared with question2.
question2	Question to be compared with question1
Is_duplicate	Result in values 0 indicating not duplicate and 1 indicating duplicate

Dataset Representation

The dataset contains Positive samples are duplicate questions, while negative samples are non-duplicates

The distribution of question in the dataset is shown by the histogram, with the majority appearing less than 60 times.



The x-axis for no. of occurrences and the y-axis for no. of questions with that occurrence count.

Data Cleaning

We computed, additional stats on our dataset that helps us explore the data and make decision in eliminating redundant data rows.

BACKGROUND

This section briefly explains the features extracted from the raw dataset and various machine learning used in the experiments.

Preprocessing:

To improve data a function is made named ‘preprocess()’ as a question is passed, it will apply some transformations on the data:

i) Lowercase the question and will strip the whitespaces.
ii) Special characters are replaced into string equivalents.
iii) ‘math’ appeared 900 times in dataset that does not have any meaning is removed.
iv) Some numbers are replaced with string equivalents.
v) Words are de-contracted (dictionary is used from the link: https://stackoverflow.com/a/19794953)
vi) Html tags are removed.
vii) Punctuations are removed.

Feature Engineering:

For the features, some basic and advanced engineering is done on the dataset.

The original features are:

Features
id
qid1
qid2
question1
question2
Is_duplicate

While making the final dataframe the features dropped are id,qid1,qid2,question1 and question2.

Basic feature engineering:

New features are created:

Features	Description
q1_len	Char length of q1
q2_len	Char length of q2
q1_num_words	No. of words in q1
q2_num_words	No. of words in q2
word_common	No. of common unique words
word_total	Total words in q1+Total words in q2
word_share	Common words divided by total words

7 new features created.

Advanced feature Engineering:

Here, three types of advanced features are added:

Token features, Length-based Features, and Fuzzy features.

Token Features:

Token features refer to the characteristics or properties of individual words or tokens in text that are used to represent the meaning and context in natural language processing (NLP) tasks. These features are used in the preprocessing step before the model is trained and are used to improve the performance of NLP tasks.

Features:

- **cwc_min**: This is the ratio of the number of common words to the length of the smaller question
- **cwc_max**: This is the ratio of the number of common words to the length of the larger question
- **csc_min**: This is the ratio of the number of common stop words to the smaller stop word count among the two questions
- **csc_max**: This is the ratio of the number of common stop words to the larger stop word count among the two questions
- **ctc_min**: This is the ratio of the number of common tokens to the smaller token count among the two questions
- **ctc_max**: This is the ratio of the number of common tokens to the larger token count among the two questions
- **last_word_eq**: 1 if the last word in the two questions is same, 0 otherwise
- **first_word_eq**: 1 if the first word in the two questions is same, 0 otherwise

A function 'fetch_token_features' is made in which Tokens are made, non-stop words, stop words, common non-stop words, common stop words, common tokens, last words similarity are fetched that provide all the above 8 features.

Length Based Features:

Length-based features are characteristics that are derived from the length of text or sequences. They are used in NLP and machine learning to represent the structural and stylistic properties of text.

Features:

- **mean_len**: Mean of the length of the two questions (number of words)
- **abs_len_diff**: Absolute difference between the length of the two questions (number of words)
- **longest_substr_ratio**: Ratio of the length of the longest substring among the two questions to the length of the smaller question.

A function 'fetch_length_features' is made. That takes row, tokens, absolute length features, average token length of both the questions and longest substring ratio are extracted as all 3 new features as above.

Fuzzy Features:

Fuzzy features uses string matching techniques to find approximate string matches and refer to features that are not clearly defined or have a degree of uncertainty in them. They are often used in natural language processing (NLP) and machine learning to represent the meaning and context of ambiguous words or phrases.

Features:

- **fuzz_ratio**: fuzz_ratio score from fuzzywuzzy
- **fuzz_partial_ratio**: fuzz_partial_ratio from fuzzywuzzy
- **token_sort_ratio**: token_sort_ratio from fuzzywuzzy
- **token_set_ratio**: token_set_ratio from fuzzywuzzy.

The ratios are method of measuring similarity between two strings:

In fuzz ratio, qratio is measuring similarity based on the token-sort-ratio in the range 0 to 100

The partial_ratio is method of alike to qratio, but considers order of the characters in the strings, while qratio compares the strings after tokenizing, sorting, and normalizing them. Range is 0 to 100.

The token_sort_ratio is method that does Tokenizing, or splitting up input strings into component words, arranging the tokens alphabetically, and comparing the resultant strings.

The token_set_ratio method like the token_sort_ratio function, it tokenizes the input strings (splits them into individual words) and then compares the resulting sets of tokens. It is useful for comparing two strings and determining their resemblance based on the words they have in common, independent of their sequence.

*we need ‘fuzzywuzzy’ library by installing it using ‘pip install fuzzywuzzy’
*then import the library.
A function is made named fetch_fuzzy_features’ that takes row, gives fuzz ratio, fuzz partial ratio, token sort ratio, token set ratio, and new features are returned as the 4 above.

Exploratory Data Analysis – EDA on new features:

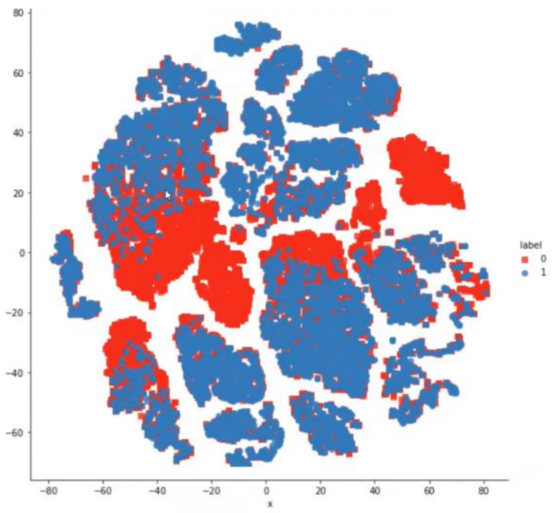
To check if the new features are useful graphs are being plot, using pairplot.

TSNE:

T-Distributed Stochastic Neighbor Embedding

t-SNE is a technique to reduce high-dimensional data and visualize it in a lower dimensional space, while preserving the similarity of data points used to reveal patterns and clusters in the data that would not be obvious otherwise. Can also be used as a preprocessing step to improve machine learning algorithms and understand data structure. 15 features are replaced with the 2 features of TSNE to check the difference between the duplicate and non-duplicate questions.

2d plotting:



Conclusion:
the differences of 0 and 1 is visible to some extent. So there is some form of explain ability after adding these new features hence giving helpful explanation.

Working:

A dataframe was made that has basic and advanced features in it. Count vectorizer is called with max features set to 3000. The questions are transformed in to bag of words. Since the total questions were 60k, after train_test split gave shape (30,000 , 6000) . After the concatenation dataframe with 6023 features is created that has all are new features in it.

MACHINE LEARNING MODELS

Decision tree induction

The Random Forest Classifier is an ensembeling technique in Machine learning for classification and regression issues that builds decision trees during training and outputs the class that represents the mean prediction of all the individual trees or the mode of the classes.

The Strengths of Random Forest

The strengths of decision includes:

- It can handle a large number of features, including categorical and continuous features.
- It is robust to outliers and can handle missing data well.
- It typically has high accuracy and is able to handle non-linearly separable data.

- It can be used for both classification and regression tasks.
- It gives feature importance, which is a measure of how much each feature contributes to the decision making process.

.

The Weaknesses of Random Forest

the weaknesses of a random forest classifier include:

- It can be computationally expensive, especially for large datasets.
- It can be difficult to interpret the output, as it is a "black box" model.
- It may overfit if the number of trees is too high.
- It can be sensitive to the parameters used to create the model.

The Random Forest classifying Model gave an accuracy of 78.46%

XG Boost classifier:

A strong use of the gradient boosting method is XGBoost- eXtreme Gradient Boosting. Large datasets may be handled with ease and it is built to be very scalable. It is helpful for resolving classification, regression, and ranking issues. It is an ensemble approach, which means that the predictions from several models (decision trees) are combined to get the final forecast.

The XG boost classifier gave an accuracy of 79.36%

EVALUATION

This section discusses evaluation and comparative analysis of the results.

Baseline Model Classifiers

We trained our model and then evaluated the prediction on our test data set to achieve the baseline for our machine learning algorithms used in this research.

Table shows test accuracy of our baseline machine learning models.

Classifier	Accuracy
Random Forest	78.46%
XG Boost	79.26%

Learnings and Conclusion:

It is important to get a high accuracy of results for classification of duplicate questions in order to provide a good user experience. Decision tree gave 78.46% score and XGBoost gave 79.26% score although this is the best score but the cost of misclassification of question duplication is high in this problem, that will have effect on our user experience. By applying Confusion Matrix on both the algorithms, It is observed that Random forest does less mistakes. Thus we can say that Random forest performed best for this dataset and we can apply deep learning concepts to achieve greater scores.

Future work for improvement:

We can increase the data, more preprocessing steps such as stemming can also improve the working. Create more features, apply more algorithms, our approach was using of bag of words, replacing it with word2vec, tfidf and deep learning can further improve the overall working with better accuracy.

Acknowledgement

Special thanks to Ma'am Saniya for her mentorship during this project. The instructor put in a great deal of effort to facilitate our learning and demonstrated dedication to the task at hand. throughout the semester. We are grateful for all her efforts and for being so cooperative we wish her continued success in her future endeavors.

.

References

[1] SBERT Team. (n.d.). Quora Duplicate Questions [Online].
Available: https://www.sbert.net/examples/training/quora_duplicate_questions/README.html

[2] Kaggle Team. (2017, March 17). Quora Question Pairs [Online].
Available: https://www.kaggle.com/competitions/quora-question-pairs/discussion.

[3] Bhaskar, U. (2018, July 12). Quora Question pair similarity. [PDF].

Available: https://github.com/UdiBhaskar/Quora-Question-pair-similarity/blob/master/Quora%20Question%20pair%20similarity.pdf

[4] Kapadia, K. (2020, March 10). Quora Question Pair Similarity Problem. [Jupyter Notebook].
Available: https://github.com/Karan-kapadia/Quora-Question-Pair-Similarity-Problem/blob/master/karankapadia583%40gmail.com_m22.ipynb.

[5] Quora User. (n.d.). What machine learning techniques would you suggest for the Quora Question Pairs competition on Kaggle [Online].
Available: https://www.quora.com/What-machine-learning-techniques-would-you-suggest-for-the-Quora-Question-Pairs-competition-on-Kaggle

[6] Kalimuddin, K. (2022, April 7). Quora Question Pair Similarity Problem: Identify which questions asked on Quora that have already been asked. [Online]. Available:
https://medium.com/@Kalimuddin_/quora-question-pair-similarity-problem-identify-which-questions-asked-on-quora-that-have-already-7724bda71ec2

[7] GeeksforGeeks Team. (2022, June 29.). FuzzyWuzzy Python Library. [Online]. Available:
https://www.geeksforgeeks.org/fuzzywuzzy-python-library/

[8] Python Tutorials. (2020, January 1). Fuzzy Wuzzy Python Library [Video file]. Retrieved from
https://www.youtube.com/watch?v=ynTCUngbFHA

[9] NaikKrish. (2020, January 1). Video1 [Video file]. Retrieved from
https://youtube.com/watch?v=WjLjjx8wSz0

[10] 5 minute engineering. (2020, January 1). Ensemble Learning [Video file]. Retrieved from
https://youtube.com/watch?v=WjLjjx8wSz0

[11] Data Science Tutorials. (2019, April 28). Ensemble Method [Video file]. Retrieved from
https://www.youtube.com/watch?v=CV9PE3iTjPI

[12] Seatgeek, K (n.d.). FuzzyWuzzy: Fuzzy String Matching in Python. [Online]. Available:
https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/

[13] GeeksforGeeks Team. (2011, July 8.). Using CountVectorizer to Extracting Features from Text. [Online]. Available: https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/

[14] GeeksforGeeks Team. (2023, January 10). Removing Stop Words using NLTK in Python. [Online]. Available: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

*__Referencing method: IEEE__