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 (Deep learning)

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

In summary, 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, Deep learning, model,

neural network, Decision tree induction, 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, Quora used a cutting-edge data science technique to organise the data properly.

Research Problem

Quora uses advanced data science techniques to prevent question duplication and maintain content quality on the platform. The goal is to recommend existing questions to users, rather than allowing them to post new ones. This helps to keep the knowledge base organized and trustworthy.

Example.

1. Does Mount Everest exist in Nepal?

And

1- Does The mount Everest of height 8848m exists in Nepal?

Duplicate questions on Q&A websites can make it difficult for users to find the information they need. These questions can be time-consuming and complex to read through.

Q1: How can we detect duplicate questions on Quora using

machine learning and deep learning methods?

Q2: How can we achieve the best possible prediction results

on detecting semantically similar questions ?

This Work

We used machine learning algorithms to extract attributes from a dataset of question pairs, used feature engineering, and tested several methods to establish a baseline. Our results improved but removing some features did not affect them. We surpassed the standard set by previous literary works and chose our top four deep learning architectures. Our best deep learning model achieved an

accuracy of 7828333333333334%.

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.

DATASET

The data collection, exploratory data analysis, data visualization, and data cleaning process.

Data collection

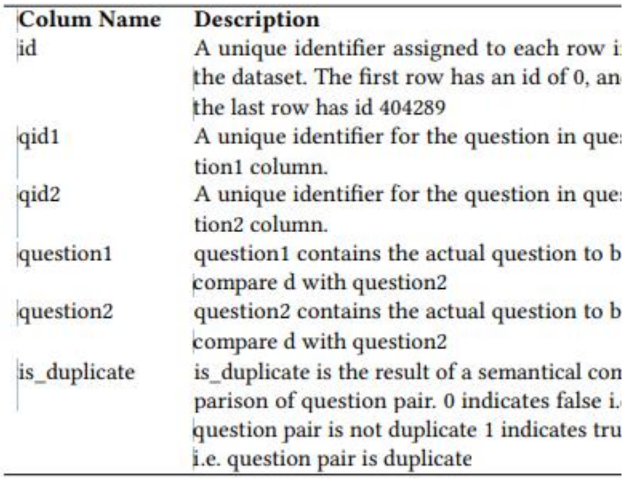
The research study's data comes from the First Quora Dataset Released on Kaggle for a Quora-sponsored competition.

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 Exploration

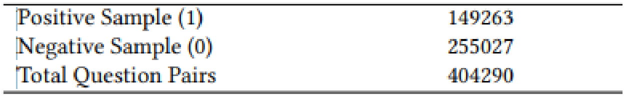
We analyzed the dataset and found that it contains six columns that provide information about the duplicate questions. The columns are as:



Colum Name	Description
id	A unique identifier assigned to each row i the dataset. The first row has an id of 0, an the last row has id 404289
qid1	A unique identifier for the question in que: tion1 column.
qid2	A unique identifier for the question in que: tion2 column.
question1	question1 contains the actual question to b compare d with question2
question2	question2 contains the actual question to b compare d with question2
is_duplicate	is_duplicate is the result of a semantical con parison of question pair. 0 indicates false i. question pair is not duplicate 1 indicates tru i.e. question pair is duplicate

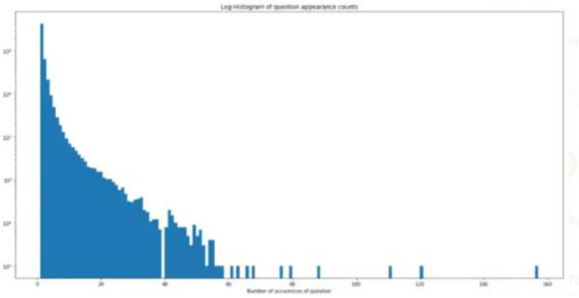
Dataset Representation

The dataset contains information on question pairs and their class labels. Positive samples are duplicate questions, while negative samples are non-duplicates



Positive Sample (1)	149263
Negative Sample (0)	255027
Total Question Pairs	404290

The histogram shows the distribution of question occurrences in the dataset, with the majority occurring less than 60 times.



Distribution of question occurrence in dataset (histogram)

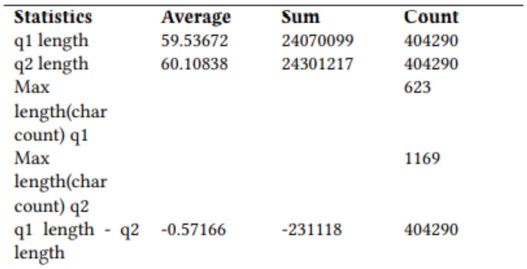
The x-axis represents the number of occurrences and the y-axis shows the number of questions with that occurrence count.

Data Cleaning

We computed, additional stats on our dataset that helps us explore

the data and make decision in eliminating redundant data rows.

Statistics on question 1 and question 2



Statistics	Average	Sum	Count
q1 length	59.53672	24070099	404290
q2 length	60.10838	24301217	404290
Max length(char count) q1			623
Max length(char count) q2			1169
q1 length - q2 length	-0.57166	-231118	404290

Mostly these questions short length questions are one word, one and two length questions are just the question marks and special characters, foreign characters. We discard as these data rows in the data cleaning process. In Table we can see that the q2 length on an average is greater, and therefore, we have an average negative difference. Thus, we have 404218 data rows in our machine learning experiments, and we continue with the usual data with 404290 rows for our deep learning experiments.

BACKGROUND

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

Feature Engineering

We dropped the first three columns id, qid1, and qid2 from the initial raw dataset and created additional useful features so that we

have two columns question1, question2, and class label is_duplicate and 28 new derived features, Therefore initially, we have total of

thirty-one columns in dataset provided as input to the machine learning classifiers.

Original Feature

1. **Question 1 dataset**: This is the question1, column in the dataset.

2**. Question 2 dataset**: This is the question2, column in the dataset.

3**. Is duplicate**: Class label represented as 1 for duplicates and 0 for non-duplicates.

Basic Features

4. **Length of question1**: Length of the question1, includes all the characters, punctuation and white spaces.

5. **Length of question 2**: Length of the question2, includes all the characters, punctuation and white spaces.

6. **Difference in the length of questions**: Difference between the length of corresponding question1 and question2.

7. **Number of characters in q1**: Distinct number of characters excluding white spaces in corresponding question1.

8. **Number of characters in q2**: Distinct number of characters excluding white spaces in corresponding question2.

9. **Number of words in q1**: Number of words in question1 including repeated words.

10. **Number of words in q2**: Number of words in question2 including repeated words.

11. **Number of common words in q1 and q2**: Distinct common words in corresponding question1 and question2.

Fuzzy Feature

12. **Qratio**: Qratio feature is the quick ratio comparison of the two question strings and has value range from 0 to 100. More similar

questions have a higher score.

13. **Wratio**: Wratio feature is the weighted ratio that uses different algorithms to calculate the matching score and returns the best ratio for two question strings. Score range from 0 to 100.

14. **Partial ratio**: Partial ratio feature calculates the best score for partial string matching against all sub strings of the greater

length and returns the best score. Score range from 0 to 100.

15. **Token set ratio**: calculates similarity between two strings and assigns a score ranging from 0 to 100, indicating the level of similarity.

16. **Token sort ratio**: Token sort feature tokenizes the string and then sort the strings alphabetically and join back into strings. It

then compares the transformed strings using ratio to return score.

Score range from 0 to 100.

17**. Partial token set ratio**: Partial token set feature is similar to token set ratio except that after it tokenizes string it uses partial ratio in place of ratio to calculate the matching score. Score range from 0 to 100.

18. **Partial token sort ratio**: Partial token sort ratio is similar to token sort ratio except that it uses partial ratio in place of ratio, after sorting the token to compute matching score. Score range from 0 to 100.

MACHINE LEARNING MODELS

Decision tree induction

Decision tree learning is a method used to create a model that predicts a value based on input factors. It is commonly used for data classification and represented visually as a flowchart. The tree is constructed using a greedy algorithm and improved by removing branches that represent noise or outliers. The knowledge represented in the tree can also be extracted and represented as IF-THEN rules for human understanding.

Strengths and Weakness of Decision Tree Methods

The strengths of decision tree methods are:

Decision trees are able to generate understandable rules.

Decision trees perform classification without requiring much computation.

Decision trees are able to handle both continuous and categorical variables.

Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision tree methods

Decision trees are a machine learning method used for classification and prediction tasks, but have limitations like difficulty in estimating continuous attributes, errors with many classes and small training sets, computational cost, and handling non-rectangular regions. To improve performance it's important to use techniques like pruning to avoid overfitting the data.

Avoiding over-fitting the data

Decision tree learning can be prone to overfitting, which can be avoided by stopping tree growth early or post-pruning. Criteria such as using a separate set of examples, statistical tests, or the Minimum Description Length principle can be used to determine the correct final tree size. Identifying duplicate records is a common problem in data integration and is often solved using string similarity metrics. Data integration aims to achieve a unique, complete, and correct representation of every object through data cleansing and identifying duplicate objects. Decision tree is a classifier in the form of a tree structure, where each
node is either:
- **A leaf node** - indicates the value of
the target attribute (class) of
examples, or
- **A decision node** - specifies some test
to be carried out on a single
attribute-value, with one branch and
sub-tree for each possible outcome
of the test.

A decision tree is used for classification by starting at the root and following the branches to a leaf node, which provides the classification of the instance..

A decision tree consists of 3 types of nodes:
1. Decision nodes - commonly
represented by squares
2. Chance nodes - represented by
circles
3. End nodes - represented by triangles

Advantages of Decision trees:

Decision trees are a type of machine learning algorithm used for classification and prediction tasks that have advantages such as simplicity and ability to determine outcomes but also have disadvantages such as bias and complexity. They form a hierarchy of branches by applying decision rules to input values, which can be visualized as an inverted tree. Decision trees can complement or substitute other forms of analysis.

The following are the essential conditions for decision tree mining:

a. Attribute-value description:

An item or situation must be able to be expressed in terms of a predetermined set of characteristics. This implies that either we must discretize continuous properties or that the algorithm must have done so.
b. Predefined classes (target attribute values): The target attribute values for examples have been established beforehand in supervised data.
c. Discrete classes: A case does or does not belong to a particular class, and there must be more cases than classes.
d. Sufficient data: Usually hundreds or even thousands of training cases.

DESCRIPTION OF MODELS AND RESULTS

EVALUATION

This section discusses evaluation metrics and comparative analysis of the results.

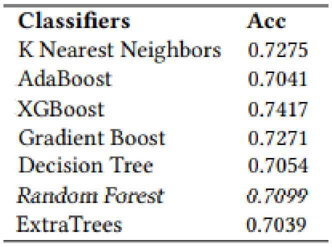
Evaluation Metrics

The selection of metrics is the most crucial step in the evaluation of our models as it influences how we measure the performance of our model against each other and the baselines.

Accuracy: Accuracy is the ratio of the total number of correct predictions made by the models to the total number of predictions requested to the model.

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.



Classifiers	Acc
K Nearest Neighbors	0.7275
AdaBoost	0.7041
XGBoost	0.7417
Gradient Boost	0.7271
Decision Tree	0.7054
Random Forest	*0.7099*
ExtraTrees	0.7039

Conclusion

Given the increased used of online forums to ask questions, the task of detecting similarity of questions is of much significance now. It is important to get a high accuracy of results for classification of duplicate questions in order to provide a good user experience.
Decision trees give considerably good scores, ensemble methods outperform all other models with scores of Random Forests being the accuracy score is 0.7875. Thus we can say that Extra Trees performed best for this dataset and we can apply deep learning concepts to achieve greater scores.

Learnings

We found that the best performance in classifying duplicate questions was achieved using the Random Forest algorithm with bagging. We used MLP However, there is room for improvement by trying other deep learning models or by incorporating pre-trained word embeddings for faster processing. These methods would require more computational resources.

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] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, “Wisdom in the social crowd: An analysis of Quora,” WWW 2013 - Proc. 22nd Int. Conf. World Wide Web, pp. 1341–1351, 2013.
[2] S. Viswanathan, N. Damodaran, and A. Simon, Advances in Big Data and Cloud Computing, vol. 750, no. January. Springer Singapore, 2019.
[3] A. Tung and E. Xu, “Determining Entailment of Questions in the Quora Dataset,” pp. 1–8, 2017.
[4] E. Dadashov, S. Sakshuwong, and K. Yu, “Quora Question Duplication,” pp. 1–9, 2017.
[5] T. Addair, “Duplicate Question Pair Detection with Deep Learning.”
[6] N. Jiang, Lili, Chang, Shuo, Dandekar, “Semantic Question Matching with Deep Learning,” Blog Post. [Online]. Available: https://www.quora.com/q/quoraengineering/Semantic-Question-Matching-with-DeepLearning. [Accessed: 04-May-2019].
[7] M. R. Morris, J. Teevan, and K. Panovich, “What do people ask their social networks, and why?,” p. 1739, 2010.
[8] S. A. Paul, L. Hong, and E. H. Chi, “Who is Authoritative? Understanding Reputation Mechanisms in Quora,” no. 2010, 2012.
[9] M. Nicosia and A. Moschitti, “Accurate Sentence Matching with Hybrid Siamese Networks,” pp. 2235–2238, 2017.
[10] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” Proc. 2014 Conf. Empir. Methods Nat. Lang. Process., pp. 1532–1543, 2014.
[11] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts, “A Fast Unified Model for Parsing and Sentence Understanding,” Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap., pp. 1466–1477, 2016.
[12] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
[13] D. Bogdanova, C. dos Santos, L. Barbosa, and B. Zadrozny, “Detecting Semantically Equivalent Questions in Online User Forums,” pp. 123–131, 2015.
[14] Z. Wang, W. Hamza, and R. Florian, “Bilateral multi-perspective matching for natural language sentences,” in IJCAI International Joint Conference on Artificial Intelligence, 2017.
[15] Y. Homma, S. Sy, and C. Yeh, “Detecting Duplicate Questions with Deep Learning,” 30th Conf. Neural Inf. Process. Syst. (NIPS 2016), no. Nips, pp. 1–8, 2016.
[16] J. O. JOSEPHSEN, “Similarity Measures for Text Document Clustering,” Nord. Med., vol. 56, no. 37, pp. 1335–1339, 1956.
[17] F. Gers, “Long short-term memory in recurrent neural networks,” Neural Comput., 2001.
[18] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A Decomposable Attention Model for Natural Language Inference,” Proc. 2016 Conf. Empir. Methods Nat. Lang. Process., pp. 2249–2255, 2016.
[19] S. Robertson, “Understanding inverse document frequency: On theoretical arguments for IDF,” J. Doc., 2004.

***Referencing method:**
IEEE = Institute of Electrical and Electronics Engineers
Sample citations [1] or [8, 10] -- List References numerically, in the order that you have cited them.