Duplicate Question Detection in Q&A Platforms: A Comparative Study of Traditional and Deep Learning Approaches

Jwalin Thaker

(Software Engineer (AI/ML), Independent Researcher)
Ahmedabad, India
jwalinsmrt@gmail.com

Abstract

With the world getting more and more connected, the amount of data being generated is also increasing at an alarming rate. Identifying data duplicacy and relevancy is a very important task in the field of data science and an interesting problem. This paper dives into one of the biggest data pool of questions and answers present on the internet, Quora, and presents a comprehensive analysis of duplicate question detection using Quora's question-pair dataset. We compare traditional machine learning approaches (Random Forest, XGBoost) with modern deep learning architectures (LSTM networks) for semantic similarity detection. Our experiments demonstrate that LSTM- based models achieve superior performance (78% validation accuracy) compared to conventional methods (72-74% accuracy), highlighting the importance of sequence modeling for natural language understanding tasks. The study provides insights into fea- ture engineering challenges, model scalability, and computational trade-offs in real-world NLP applications.

Keywords: Duplicate Question Detection, LSTM, Random Forest, Xgboost, Natural Language Processing

I. Introduction

Online question-and-answer platforms like Quora face signif- icant challenges in content moderation, with duplicate questions estimated to account for 15-20% of new submissions [**?**]. These redundancies strain platform resources and degrade user experience through fragmented answer threads. Traditional rule-based detection systems prove inadequate due to natural language variations, necessitating more sophisticated semantic analysis approaches like those proposed in [**?**]. The problem of duplicate question detection has been extensively studied in various Q&A platforms, with notable work by Zhang et al. [**?**] on Stack Overflow data demonstrating the complexity of this task.

Our contribution includes:

- A systematic comparison of tree-based models and recur- rent neural networks
- Feature engineering insights for textual similarity tasks
- Practical implementation considerations for large-scale NLP systems

- Error analysis of different architectural approaches

II. Related work

Detection of duplicate questions is a very long standing problem. This same problem is been solved by many other ways and we have taken two of the approaches as a reference. The first one is of Dey, Kuntal, Ritvik Shrivastava, and Saroj Kaushik, wherein they have demonstrated the various machine learning algorithms such as traditional approach of SVM. Hand picked and heterogeneous features were used. They used words overlap, negation modelling. The data was extremely pre- processed to perform well. Wang et.al [**?**] are the only published results on Quora dataset. They got a very good result and they used modern NLP techniques. They observed that the encoding procedure does not provide interaction between the two input sequences. As a result, they proposed a bilateral LSTM model. They used the approach of matching aggregation which proved to be performing better than the CNN and LSTM that they tested.

Wan et al. [**?**] proposed a deep architecture for semantic matching with multiple positional sentence representations, which has shown promising results for similar tasks. Addi- tionally, Mou et al. [**?**] explored natural language inference using tree-based convolution and heuristic matching, providing valuable insights into structural approaches for text comparison. The work by Rodrigues et al. [**?**] specifically addresses ways of asking and replying in duplicate question detection, which is directly relevant to our problem domain.

In our project we will however use both the techniques of traditional approach and modern technique and compare the results. We also consider semantic textual similarity evaluation approaches as described by Agirre et al. [**?**], which provide standardized methods for comparing text similarity systems.

III. Approach

In order to detect the duplicate questions we have perform initial data exploration and analysis. Data pre-processing is performed to prepare the raw data to make it suitable for building and training models. We checked our dataset for null values and duplicates and removed them if found (¡10 count). Post cleanup we created some visualizations to look at distribution of some attribute(s) and get valuable insight on how the data was structured.

In terms of algorithm implementation, we started with two of the traditional algorithms namely Random Forest and XG Boost. Random Forest is a supervised Machine Learning algorithm and it is the most widely used algorithm because of its accuracy, simplicity and flexibility and can be used for classification and regression tasks, combined with its non-linear nature makes it highly adaptable to a range of data and situations.

The second algorithm that we used in this approach was XG Boost algorithm. It is an implementation of gradient boosted decision trees designed for speed and performance as described by Chen and Guestrin [**?**]. It is a decision tree based ensemble algorithm that uses a gradient boosting framework and predicts problems involving unstructured data same as images, texts etc. in neural networks. It generally outperforms all other traditional algorithms or frameworks.

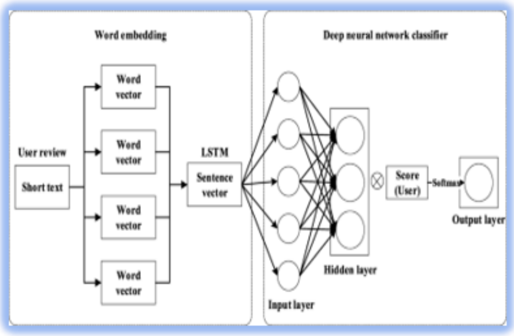


Fig. 1. Random Forest working

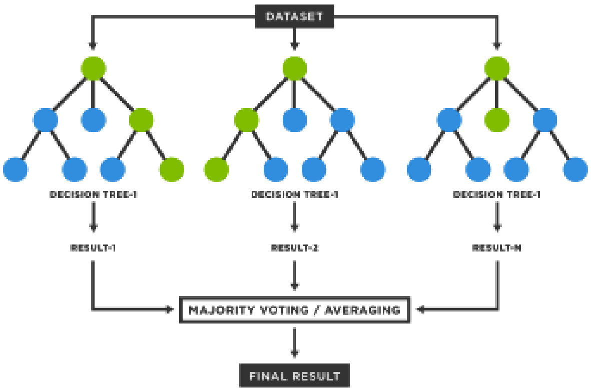


Fig. 2. XGBoost working

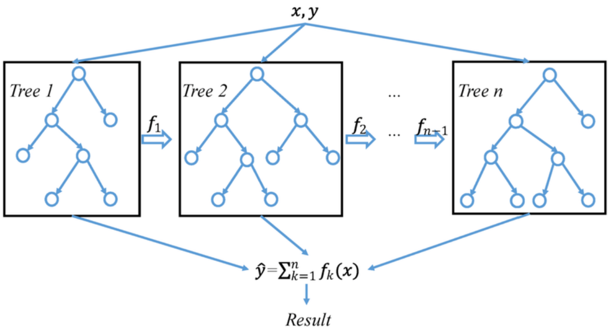
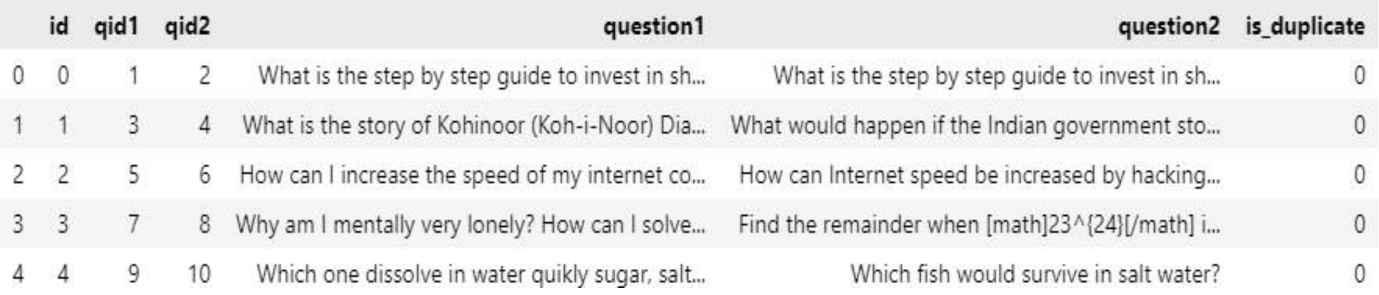


Fig. 3. LSTM working

Our final algoirthm involved adopting modern NLP method- ology, named LSTM which houses a deep learning architecture on a ARNN. We used techniques like word2vec which is a common method of generating word embeddings and has many real-life applications. LSTM leverages underlying neural network model to learn word succession and learn word associations from large corpus of data. The

LSTM architecture, originally proposed by Hochreiter and Schmidhuber [**?**], is a neural network with feedback connections that can handle both single data points and full data sequences and has the capacity of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other. A probable function figrue looks as show below –



	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

Fig. 4. Dataset sample

IV. IMPLEMENTATION

A. *Dataset Description*

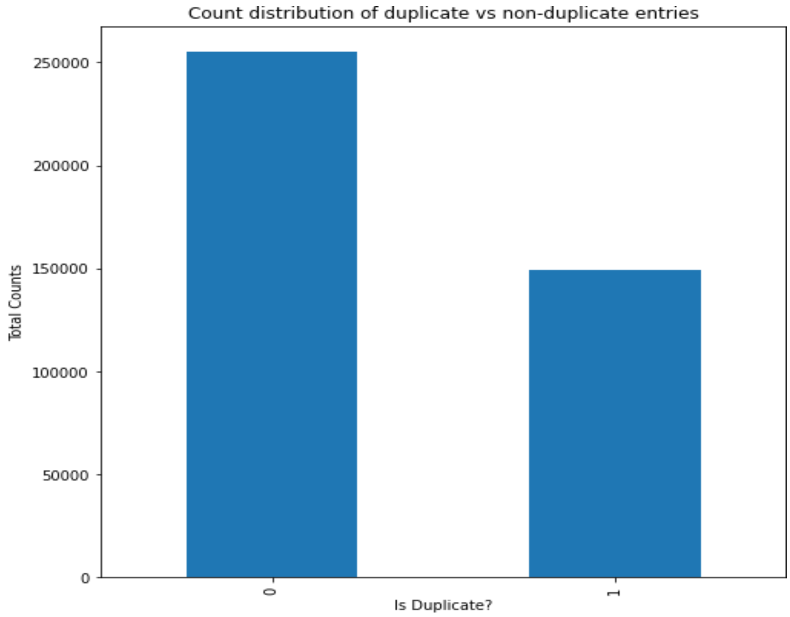
For this project, we have used the dataset which was released by Quora. It is a publicly available dataset and contains around 400k+ labeled question pairs with six features in total. The link of the dataset is: https://www.kaggle.com/datasets/quora/question- pairs-dataset. The field descriptions are as shown below in the table -

Fields	Description
id	unique identifier for the question pair
qid1	unique identifier for the first question
qid2	unique identifier for the second question
question1	full Unicode text for the first question
question2	full Unicode text for the second question
is duplicate	1 if the questions are duplicate, 0 otherwise

Here in this dataset we assume that the questions that are marked as duplicates in the dataset are truly duplicates of each other. Of all the questions, around 250,000 questions are not duplicates and 150,000 are duplicates of each other. Because the dataset has been labelled by humans manually, there can be chances of some noise in the dataset. The information of the data is given in following images -
Figure 6 shows us during EDA that most of the questions have length between 30-60 and that the dataset also includes longer questions with more than 150 characters in it, making them quite important for contextual handling.

B. *Data preprocessing*

After performing cleaning 101s on the dataset, we performed alterations related to textual dataset; removing all the punctua- tion makes, broken numbers, links and nonsensical white spaces, helping us counter the manually sourced data discrepancy issue. Then, we divided our dataset into three sets of training, validation and test sets. The training set contained around 195k entries whereas the validation and test entries contained around 65k entries.



– **Fig. 5. Distribution of *is duplicate* column 1s and 0s**

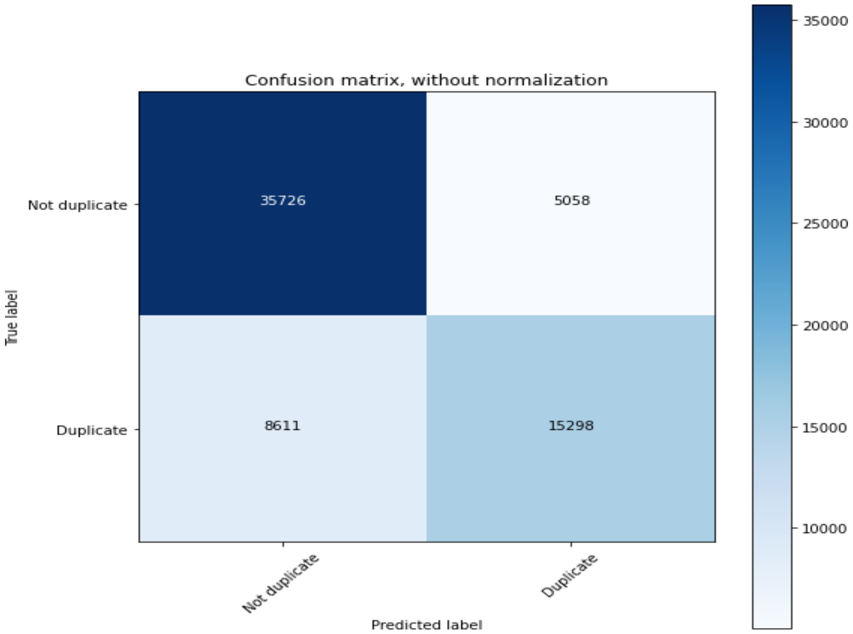


Fig. 6. Question length and respective counts

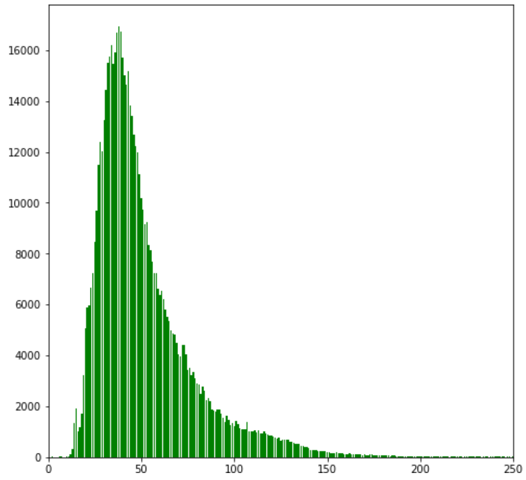


Fig. 7. Duplicate/Not Duplicate Confusion Matrix

For the sake of understanding the data, we first used only 30k rows but we did not get the results that we wanted and hence assumed that further feature engineering would improve these naive model accuracies. Feature Engineering generally refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning. For gaining more accuracy we proposed some new features like - character length in question, words in question, shared/unique words etc. For measuring document similarity, we also explored techniques like those proposed by Kusner et al. [**?**], which leverage word embeddings to compute document distances.

C. *Deep Learning Modeling*

To test the dataset against a deep learning neural network, we leverage NLP practices and model a LSTM with left and right input layers, an Adam optimizer governing ”mean square error” loss over Manhattan distance (exponent negative) calculated on the dataset. We set the parameters as follows -

1. *Number of hidden variables = 50*
2. *Gradient clipping norm = 1.25*
3. *Size of batch = 64*
4. *Number of epoch = 20 (Computation limitation)*

The next section portrays how these models fared after several trial and errors and how the results and various evaluation metrics look like. We also considered topic modeling approaches such as Latent Dirichlet Allocation [**?**] to capture thematic similarities between questions, though this was ultimately less effective than our sequence-based models.

V. Results & Evaluations

Our attempt at trying traditional classifiers like XG Boost and Random Forest proved to be a little inefficient and the accuracy that we got was quite low than an acceptable value. The Random Forest method gave an accuracy of 74.48% with a 2% room to grow with proposed feature engineering. Similarly, with the XGBoost method, we achieved 72.1% accuracy with a 3% room of improvement. These are the validation accuracies over exhaustive algorithm run. These numbers fall short than the expectation set by these highly used techniques but can be defended, as the problem statement involves extracting the underlying character and word pattern in a sentence. This resonates more with a NLP use case than a conventional ML problem.

Thus, our LSTM implementation, in its vanilla state gave an improved results over the dataset and generously tackled the overfitting issue. The training accuracy came out to be 81% while the validation and testing accuracy came out to be 78% in just 20 epochs (with 4-5% room of improvement). Figures 7 and 8 show the confusion matrix and ROC curve of the LSTM deep learning architecture. These results align with findings from Wan et al. [**?**], who demonstrated that deep architectures with multiple positional sentence representations can effectively capture semantic relationships between text pairs.

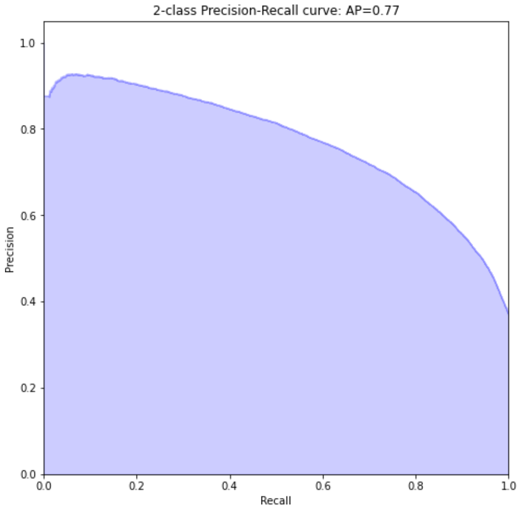


Fig. 8. Precision-Recall ROC curve of LSTM implementation

VI. CONCLUSION

With this project, we tried our best to give the solution for a long lasting data problem encountered by QA forums about detecting duplicate questions on the dataset provided by Quora. We explored two different approaches to solve this problem. The first one was using traditional classifiers namely XG Boost [**?**] and Random Forest and the other approach was using modern NLP technique like word2vec and LSTM classifier [**?**]. Our results are promising with the use of LSTM, with good performances as compared to previous approaches such as those described by Rodrigues et al. [**?**].

This work can be extended in a variety of ways. Because of the computation limitations, we could first try the same vanilla LSTM model on a better device. Data balancing techniques can also be applied to remove an bias that the model might inherently induce. The architecture can also tried by using GRU instead of LSTM as some of the recent NLP studies have shown that this technique is reliable and also shows guaranteed better results. (catch - the computation cost is very high). Future work could also incorporate semantic textual similarity evaluation frameworks as proposed by Agirre et al. [**?**] to standardize performance comparisons across different approaches.

Data Availability

The Quora question pair dataset used in this study is publicly available through the Kaggle platform (https://www.kaggle. com/datasets/quora/question-pairs-dataset).

Conflict of Interest

The author declares no conflict of interest in the preparation and publication of this research.

References

[1] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," *arXiv preprint arXiv:1702.03814*, 2017.

[2] Y. Zhang, D. Lo, X. Xia, and J.-L. Sun, "Multi-factor duplicate question detection in stack overflow," *Journal of Computer Science and Technology*, vol. 30, pp. 981–997, 2015.

[3] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[4] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," *arXiv preprint arXiv:1512.08422*, 2015.

[5] J. Rodrigues, C. Saedi, V. Maraev, J. Silva, and A. Branco, "Ways of asking and replying in duplicate question detection," in *Proceedings of the 6th joint conference on lexical and computational semantics (* SEM 2017)*, 2017, pp. 262–270.

[6] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez Agirre, R. Mihalcea, G. Rigau Claramunt, and J. Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," 2016.

[7] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International conference on machine learning*. PMLR, 2015, pp. 957–966.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.