Quora Question Duplication Detection: An ML Approach for Identifying Semantically Equivalent Questions

|  |
| --- |
|  |
| Soham Pande  *Department of Artificial Intelligence and Data Science*  Vishwakarma Institute of Technology.  Pune,India,411037  soham.pande22@vit.edu |

***Abstract*—This research paper explores the field of Natural Language Understanding (NLU) by tackling the problem of identifying duplicate questions, using the Quora dataset as the primary source of data. Our study involves a comprehensive examination of the dataset and the application of machine learning techniques, specifically Random Forest and XGBoost models. Importantly, our results emphasize the efficiency of a straightforward Continuous Bag of Words neural network model, surpassing the performance of more intricate recurrent and attention-based models. The previous problem on Quora, characterized by numerous duplicate questions, has led to increased ambiguity and a diminished user experience. We also rigorously conduct error analysis, revealing subtleties and subjectivity in the dataset's labeling process. This investigation underscores the value of neural network-based approaches in tackling the complexities of duplicate question detection, which significantly contributes to the broader landscape of NLU research.**

***Keywords— Natural Language Understanding, Duplicate Question Detection, Quora Dataset, Machine Learning, Neural Network Models, Error Analysis, Subjectivity Analysis.***

# **Introduction**

The Quora dataset presents a critical challenge in the field of Natural Language Understanding (NLU): the task of determining whether pairs of questions have identical meanings, thus classifying them as duplicates. Quora, a popular question-and-answer platform, serves as a valuable resource for users seeking answers and insights across a wide range of topics. However, the increasing volume of questions has led to a proliferation of duplicate inquiries, potentially impeding users from accessing high-quality responses. Furthermore, responders may hesitate to address the same question repeatedly.

Recognizing duplicate questions is instrumental in alleviating these issues and optimizing the user experience. It not only streamlines the burden on responders but also facilitates the redirection of users to the most pertinent responses, enhancing overall user satisfaction.

This task necessitates robust Natural Language Understanding (NLU) capabilities. Achieving effective Natural Language Understanding (NLU) necessitates the creation of meaningful representations of human language, which is a formidable challenge with far-reaching implications for various Natural Language Processing (NLP) tasks, including translation, summarization, and reading comprehension. At the core of this challenge lies the ability to determine if two sentences convey the same meaning, requiring the model to grasp nuanced aspects of language, such as quantification, tense, modality, and syntactic ambiguity.

Given the intricate nature of this task, the Quora dataset presents a compelling opportunity for exploration. In this paper, we embark on a comprehensive investigation of machine learning models to assess their performance on this dataset. Our methodology departs from previous approaches that incorporated complex models like Support Vector Machines (SVM), gradient boosted trees, and deep neural networks. Instead, we adopt a more straightforward baseline approach utilizing linear models. We conduct a thorough analysis and discussion of the performance of these models.

The essence of duplicate question detection lies in binary classification, where the objective is to classify questions of varying lengths as either duplicates or non-duplicates. The pivotal challenge is to transform sentences into numerical representations suitable for machine learning algorithms. The prevalent industry practice involves manual feature engineering, often combined with tree-based models, like random forests. This aligns with Quora's current approach (Dandekar, 2017) and can be further enhanced by incorporating bag-of-words-based models (Siu, 2016).

Although conventional methods have demonstrated effectiveness, the rise in neural network research has brought forth a wide array of deep learning techniques for sentence classification and the construction of representations (Sutskever et al., 2014; Collobert and Weston, 2008). Notably, substantial progress has been made in Natural Language Inference (NLI), a task involving the determination of entailment, contradiction, or neutrality between pairs of sentences. Inspired by work on the Stanford Natural Language Inference corpus (SNLI) (Bowman et al., 2015), our neural network explorations draw from these advancements.

In this paper, we present a comprehensive examination of duplicate question detection, spanning traditional feature engineering to state-of-the-art neural network approaches, aiming to shed light on their effectiveness and applicability.

# **Literature Review**

[1]This research leverages natural language processing technology to develop an interactive project management platform, introducing an innovative approach within the construction sector. The system streamlines contract administration processes through the integration of the Progressive Scale Expansion Network (PSENet), Convolutional Recurrent Neural Network (CRNN), and Bi-directional Recurrent Neural Networks Convolutional Recurrent Neural Network (BRNN-CNN) toolkits. This process effectively organizes papers, drastically reduces the amount of human mistakes, and clears up ambiguities through real-time exact communication. It acts as a cutting-edge remedy for huge real-time document flows, successfully fostering collaboration and communication among all contract stakeholders.

[2]This paper examines the traditional approaches to risk management in significant transport projects, concentrating on risk assessment. Expert comments gained through risk workshops have always been crucial. The uniqueness of this strategy hasn't received enough attention, though. This study intends to evaluate the similarities in project risks among 70 significant transport projects carried out using various techniques. The study thoroughly examines risk registers using natural language processing (NLP) and the deep learning technique Word2vec. Surprisingly, a considerable degree of similarity between risk registers for various projects was found, emphasizing the possibility of a data-driven strategy to establish a common risk register while taking into account the particular hazards associated with each project. The key contributions of this work are developing a relationship between risk distinctiveness and project delivery strategies in transport projects and taking a novel way to analyze risk registers at the project level.

[3]This paper delves into the collaboration between a university and the industry to develop an AI-based simulation platform for social work education in this conceptual article. The research highlights the role of Natural Language Processing (NLP) as a pedagogical innovation and conducts a critical evaluation of the ongoing project, assessing both the possibilities and limitations of NLP within the realm of social work education. The research offers "lessons learned" based on the case study and is grounded in the Community of Inquiry (CoI) paradigm. It promotes the active participation of social work educators in the development of pedagogies made possible by cutting-edge AI technologies, including artificial intelligence, natural language processing, and virtual simulation.

[4]This paper discusses the idea of technical debt, which is a compromise between developer shortcuts and programme quality. Prior studies, concentrating on design and requirement debt, found self-admitted technical debt through source code comments. This work provides an automated identification approach using Natural Language Processing (NLP), as opposed to manual methods. The research outperforms previous keyword-based techniques in detecting self-admitted technical debt after analyzing 10 open-source projects. Notably, the study uses very little comment usage to achieve amazing accuracy in identifying specific phrases connected to design and requirement debt. This method represents a significant development since it ensures precise technical debt detection with little data, demonstrating its effectiveness and promise for real-world use.

[5]This paper discusses the crucial problem of construction project scheduling quality, highlighting the difficulties encountered during the planning and design phases. Although schedule quality is frequently compromised by time restrictions, little has been done to review and maintain schedules throughout the building phase. Schedule quality can only be manually diagnosed, which takes time and is arbitrary. Using task ontology and dependency-based information schema, this study provides a unique semantic-based logic reasoning and representation technique. By automatically separating building techniques and tasks, this methodology maintains consistent project timetables. The system's effectiveness is demonstrated by the assessment, which provides academics and practitioners with an automated method to spot schedule flaws and keep high-quality schedules over the course of a project.

[6]This study explores how pharmacoepidemiological findings are communicated in the media representation of the CNODES isotretinoin research. The study uses natural language processing to analyze 26 news stories and 3 CNODES publications. The analysis pinpoints separate media coverage clusters, exposing differences in vocabulary and topics. Notably, all of the publications used more complicated vocabulary than what is advised for reading about health. The study emphasizes the significance of using NLP strategies to evaluate the efficacy of medication safety communication. This research provides crucial insights into the public's comprehension and highlights areas for development in upcoming communication strategies. It also throws light on the dissemination difficulties experienced by drug safety studies in the media.

# **Methodology**

1. **Exploratory Data Analysis:**

The dataset employed in this study consists of the Quora Question Pairs dataset, which includes a training set with 404,290 question pairs and a test set containing 2,345,795 question pairs. This dataset was initially made available as part of a Kaggle competition [1].

To facilitate the calculation of additional performance metrics and to support further error analysis on our prediction models, we opted to create our own test set based on the provided training set. Consequently, our data exploration and model evaluation were primarily centered on the training set, which consists of 404,290 question pairs.

The dataset includes various fields for each sample point:

* id: A unique identifier for each question pair
* qid1: The ID of the first question   
  qid2: The ID of the second question
* question1:The text of first question
* question2: The text of second question
* is\_duplicate: A binary value indicating whether the questions are duplicates of each other. (0 signifies they are not duplicates, and 1 signifies they are duplicates)

One notable aspect of the dataset is class imbalance. Specifically, 255,027 question pairs (63.08%) are labeled as non-duplicates (0), while 149,263 question pairs (36.92%) are labeled as duplicates (1). Managing this class imbalance played a crucial role in our model development and evaluation.

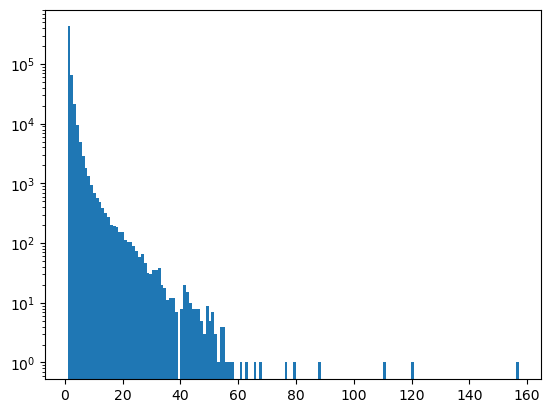


Fig. 1: Number of questions vs Repetition count

Moreover, although each question pair in the dataset is distinct, individual questions within these pairs may be duplicated. Approximately 79.22% of the questions appear more than once, with some questions recurring as many as 158 times across different pairs. The dataset comprises a total of 537,933 unique questions, and 111,780 of these questions occur in multiple question pairs. A visual representation of the distribution of question repetitions is presented in Figure 1.

It's worth noting that the character set within our dataset extends beyond ASCII characters. We identified 6,228 questions that contain non-ASCII characters, which are distributed across 8,744 question pairs. Furthermore, two pairs in the dataset contained empty strings for one of their questions.

As additional information, it's worth noting that the dataset includes 537,933 unique questions, with 111,780 questions appearing in multiple pairs. This comprehensive dataset description provides valuable insights into the dataset's characteristics, challenges, and the steps taken to address them throughout our analysis and modeling processes.

1. **Bag of Words (BoW) approach**

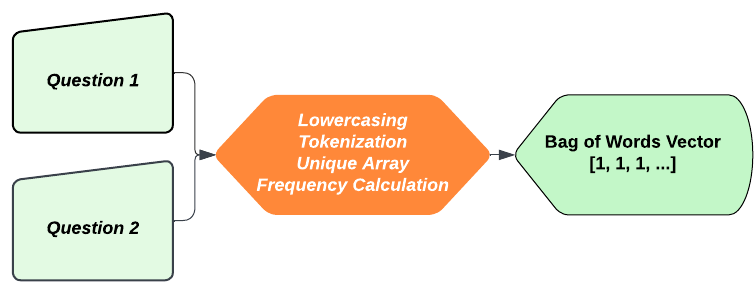
****

Fig. 2: Bag Of Words (**BoW**) Approach

In this study, we adopt the Bag of Words (BoW) approach to address our research objectives efficiently. BoW simplifies text analysis by treating documents as unordered collections of words.

The BoW algorithm includes the following steps:

* Text Preprocessing: We preprocess textual data by tokenization, lowercasing, and removing punctuation, stop words, and noise.
* Feature Extraction: We convert text into a numerical matrix, with each unique word serving as a feature, creating a high-dimensional feature space.
* Vectorization: The BoW representation is typically large and sparse. Dimensionality reduction techniques like TF-IDF may be employed to reduce complexity.

Applying the BoW approach results in the inclusion of approximately 3000 new numerical features in the dataset. This addition enhances its analytical capacity and facilitates more robust analysis and modeling.

1. **Data Pre-processing**

In the data preprocessing phase, several techniques were applied to enhance the quality and consistency of the dataset. These preprocessing steps included:

* Lowercasing: Both questions were converted to lowercase to ensure uniformity in text case throughout the dataset.
* Whitespace Removal: Extraneous white spaces were removed to streamline the text and maintain data cleanliness.
* Special Character Conversion: Special characters were transformed into their respective string equivalents to eliminate any potential irregularities in the text.
* Decontracting Words: The decontraction process was applied to standardize contractions (e.g., "can't" to "cannot") for better text analysis.
* HTML Tag and Punctuation Removal: HTML tags and punctuation marks were removed from the text to improve text clarity and facilitate subsequent analysis.

In addition to data preprocessing, we undertook basic feature engineering by developing various functions that introduced seven new features into the dataset. These newly created features are as follows:

* q1\_len: The length of the first question.
* q2\_len: The length of the second question.
* q1\_words\_number: The number of words in the first question.
* q2\_words\_number: The number of words in the second question.
* words\_common: The count of common words shared between both questions.
* total\_words: The total number of words in both questions combined.
* word\_share: The ratio of common words to the total number of words (common/total).

As a result of these data preprocessing steps and the incorporation of additional features, the dataset now encompasses a total of 6007 features, which collectively contribute to a more comprehensive and analytically rich dataset for our research analysis.

1. **Advanced Feature Engineering**

In this phase of advanced feature engineering, our aim was to further enhance the predictive power of our model beyond what basic feature engineering had initially provided. Through careful analysis and insights drawn from the Kaggle competition and the research community, we identified the need to incorporate additional parameters into our dataset. These new features fall into three distinct categories: token features, length-based features, and fuzzy features.

To better comprehend these feature additions, it is essential to establish some key terminology. Firstly, "token" refers to all the words present in a given question. "Stop words," on the other hand, denote words that contribute little semantic meaning to a sentence, such as "a," "the," or "of." Thus, "words" refer to tokens excluding stop words.

Building upon this foundation, we introduced a set of features inspired by successful approaches observed in other solutions within the Kaggle competition:

A. Token Features:

* cwc\_min: The ratio of the number of common words to the length of the smaller question.
* cwc\_max: The ratio of the number of common words to the length of the larger question.
* csc\_min: The ratio of the number of common stop words to the smaller stop word count among the two questions.
* csc\_max: The ratio of the number of common stop words to the larger stop word count among the two questions.
* ctc\_min: The ratio of the number of common tokens to the smaller token count among the two questions.
* ctc\_max: The ratio of the number of common tokens to the larger token count among the two questions.
* last\_word\_eq: A binary attribute indicating whether the last word in the two questions is the same (1 if it is, 0 otherwise).
* first\_word\_eq: A binary attribute indicating whether the first word in the two questions is the same (1 if it is, 0 otherwise).

B. Length-Based Features:

* mean\_len: The mean length (number of words) of the two questions.
* abs\_len\_diff: The absolute difference between the lengths (number of words) of the two questions.
* longest\_substr\_ratio: The ratio of the length of the longest common substring between the two questions to the length of the smaller question.

C. Fuzzy Features:

* fuzz\_ratio: Fuzzy ratio score.
* fuzz\_partial\_ratio: Fuzzy partial ratio.
* token\_sort\_ratio: Token sort ratio.
* token\_set\_ratio: Token set ratio.

By incorporating these additional features, our dataset now boasts a total of 15 new attributes. When combined with the existing 6007 features, this comprehensive set of 6022 features equips our model with a richer and more informative dataset, enabling us to improve accuracy and enhance the predictive capabilities of our research.

1. **Models**

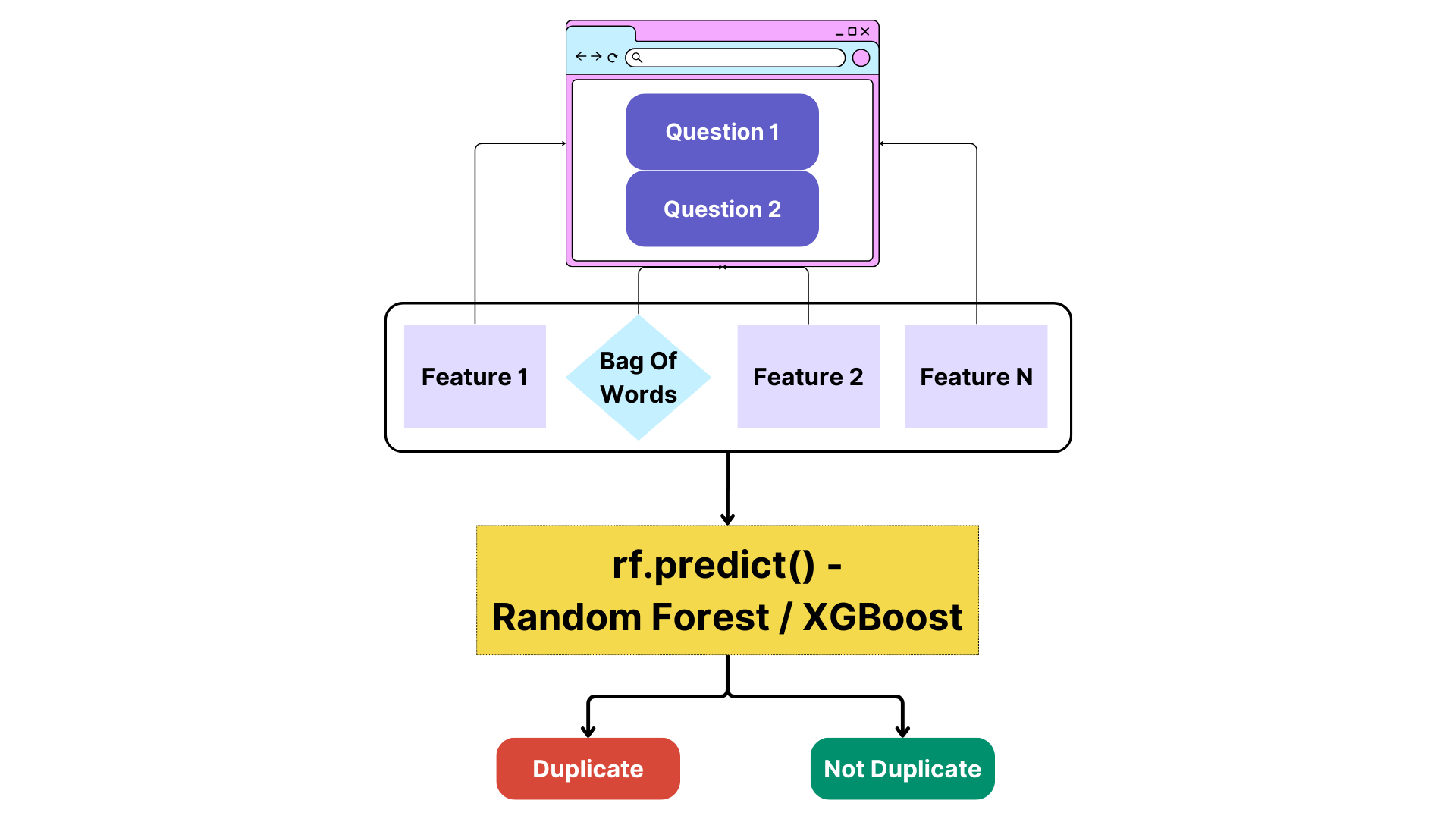
****

Fig. 3: Program Flow

With the completion of our data preprocessing phase, we now possess a clean and structured dataset, primed for predictive modeling. Our approach involves utilizing machine learning models to forecast the desired output. As each new feature is extracted, it is seamlessly integrated into the evolving data frame, enriching our dataset with valuable information.

In this research, we have specifically opted to employ two machine learning models: Random Forest and XGBoost. These choices are motivated by their suitability for our research objectives.

Random Forest is an ensemble learning method known for its robustness and ability to handle both categorical and numerical data effectively. It is capable of handling high-dimensional datasets, making it particularly well-suited for our feature-rich dataset. Moreover, its ensemble nature, which combines multiple decision trees, often results in improved accuracy and generalization.

XGBoost, short for Extreme Gradient Boosting, is another ensemble learning algorithm that has gained popularity for its exceptional performance in a variety of machine learning tasks. XGBoost excels in scenarios where feature engineering is critical, as it can handle complex relationships between features. Its efficient implementation and optimization for both classification and regression tasks make it a powerful choice for our research.

In conclusion, we have chosen Random Forest and XGBoost as our machine learning models due to their robustness, versatility, and ability to handle high-dimensional data effectively. These models are well-aligned with the complexities of our dataset and are expected to provide accurate and reliable predictions for our research objectives.

IV. **Result And Discussion**

In our initial experimentation with basic feature engineering for both Random Forest and XGBoost, we achieved an accuracy rate of 76 percent. However, recognizing the potential for improvement, we embarked on advanced feature engineering, which led to notable enhancements in predictive performance. Specifically, Random Forest achieved an accuracy of 78 percent, while XGBoost demonstrated an even higher accuracy of 79 percent. At first glance, this might suggest that XGBoost outperforms Random Forest in terms of accuracy.

|  |  |  |
| --- | --- | --- |
| **Algorithm/Model** | **Data Size** | **Accuracy** |
| Random Forest | 30000 | 78% |
| 160000 | 81% |
| XGBoost | 30000 | 79% |
| 160000 | 80% |

Table 1: Varying accuracy based on sample size

Nonetheless, when evaluating the models beyond accuracy, we must consider the implications of a greater number of false positives. In the context of our problem, minimizing false positives is of paramount importance. When a non-duplicate question is erroneously marked as duplicate, it represents a significant drawback in terms of user experience, an outcome we aim to avoid. Therefore, despite its slightly lower accuracy, we argue that Random Forest may be a more suitable choice due to its potential to yield fewer false positives.

However, there is always room for improvement. In future research, strategies such as undersampling or oversampling could be explored to address the class imbalance within the dataset. Additionally, while we trained our models with a subset of 3000 data points due to computational limitations, augmenting the dataset with more examples could further enhance accuracy.

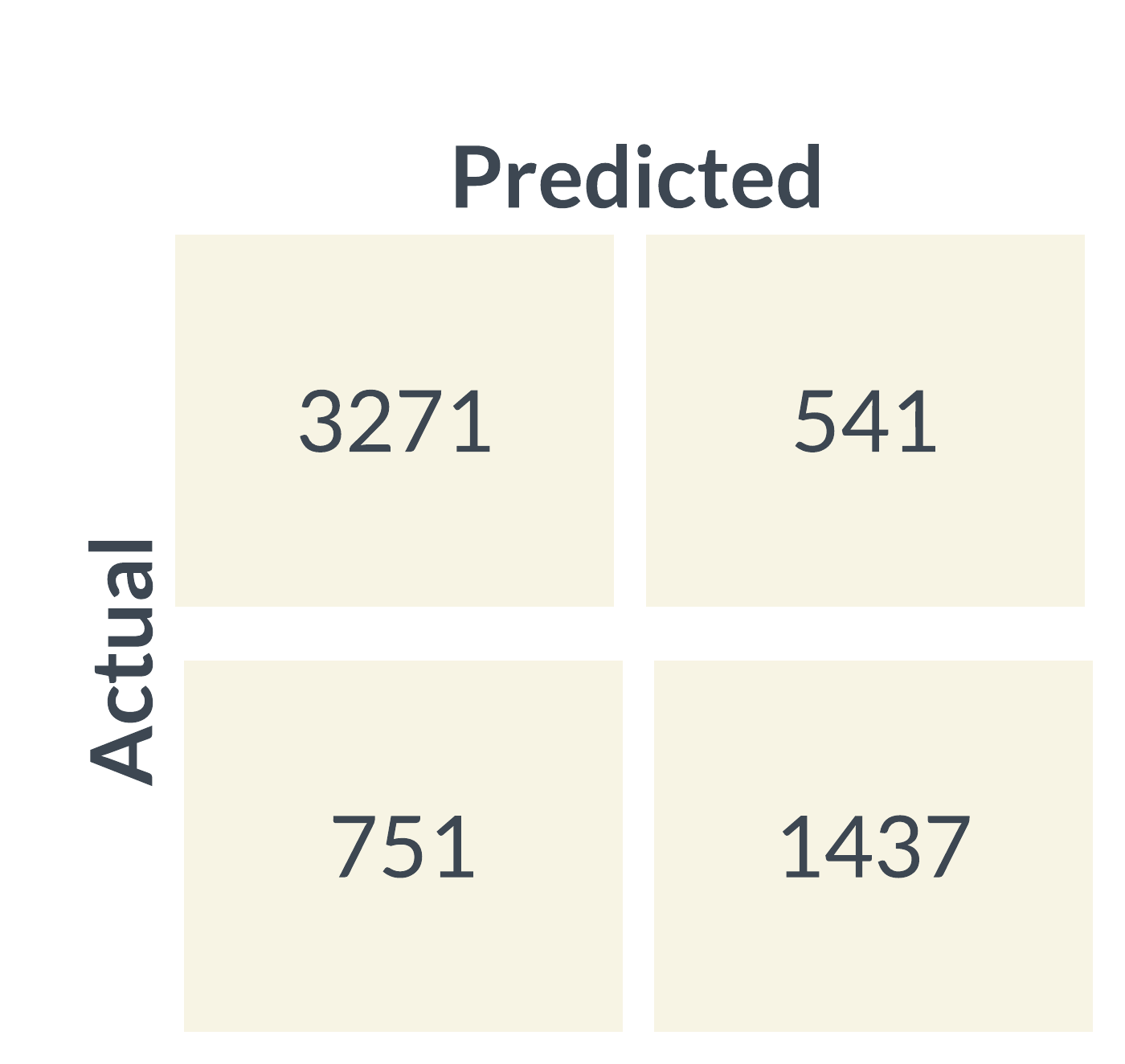


Fig. 4: Confusion matrix for random forest

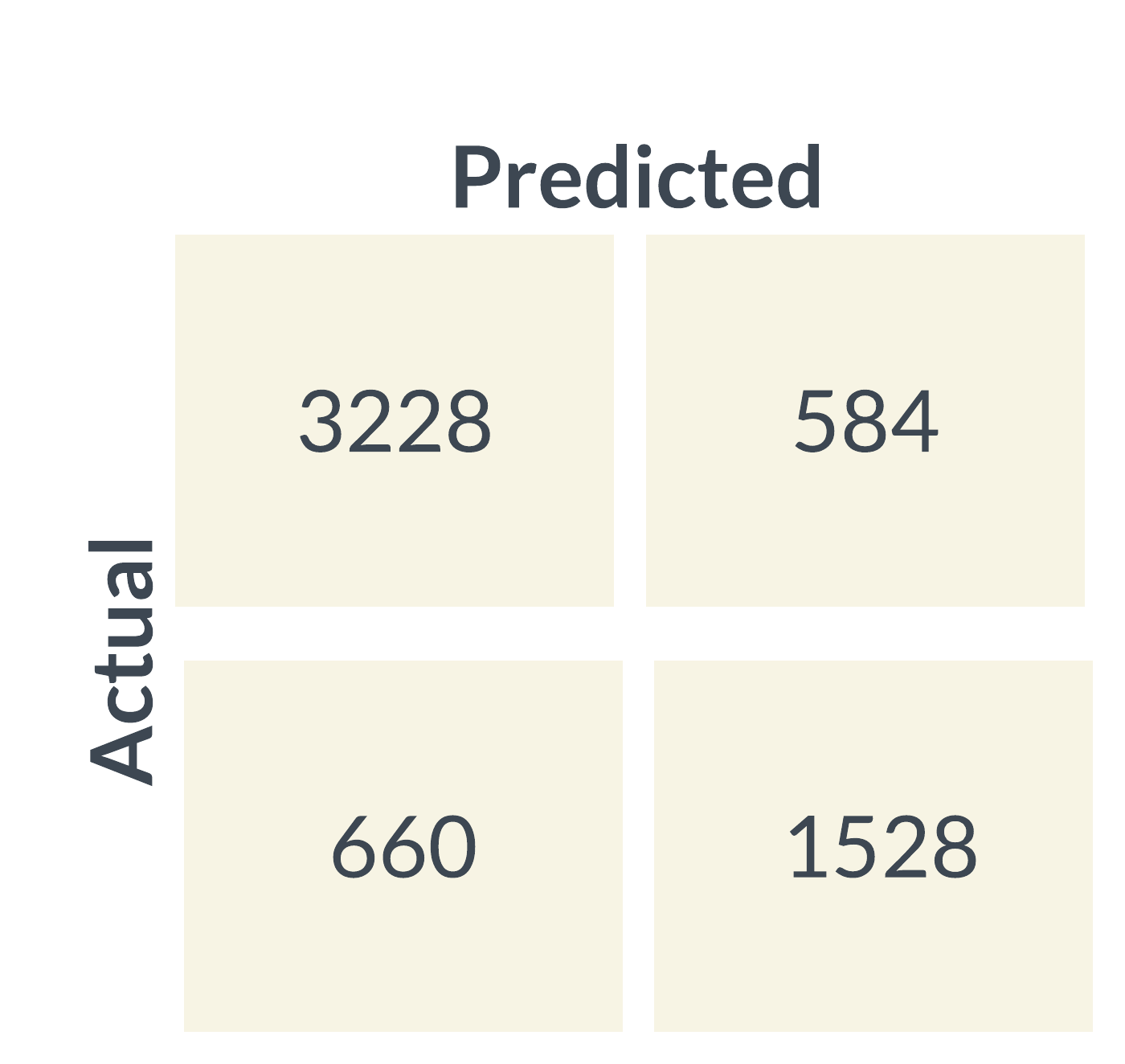


Fig. 5: Confusion matrix for random forest

Furthermore, we recognize several avenues for potential enhancements. Implementing techniques like "Stemming" to reduce words to their root form and hyperparameter tuning to optimize model performance could yield substantial improvements. Expanding our scope beyond just Random Forest and XGBoost, we recommend exploring a broader range of machine learning algorithms, including Support Vector Machines (SVM) and Logistic Regression, to assess their effectiveness in this context.

Additionally, the creation of additional features could enhance model capabilities. Lastly, it is worth noting that while we employed the Bag of Words (BoW) technique for question vectorization, utilizing more advanced approaches like Word2Vec may yield even more accurate results. These considerations underscore the potential for ongoing research and refinement in pursuit of improved accuracy and model effectiveness.

V. **Limitations**

1. Data Size and Quality: The study relied on a relatively small dataset with potential data quality issues. A larger and cleaner dataset would enhance the reliability and generalizability of our findings.
2. Algorithm Selection: We limited our analysis to two machine learning algorithms due to resource constraints. Exploring a broader range of algorithms could uncover more effective approaches for the problem at hand.

VII. **Conclusion**

In our quest to tackle the issue of duplicate questions within the Quora dataset, we conducted a comprehensive evaluation of various machine learning models. Surprisingly, our most successful model proved to be a simple Continuous Bag of Words neural network. Our findings highlight the Quora dataset's potential as a valuable resource for advancing Natural Language Understanding tasks through machine learning techniques.

**References**

[1] Chen, Jieh-Haur, et al. "Smart project management: interactive platform using natural language processing technology." Applied Sciences 11.4 (2021).

[2]Erfani, Abdolmajid, Qingbin Cui, and Ian Cavanaugh. "An empirical analysis of risk similarity among major transportation projects using natural language processing." Journal of Construction Engineering and Management 147.12 (2021).

[3] Asakura, Kenta, et al. "A call to action on artificial intelligence and social work education: Lessons learned from a simulation project using natural language processing." Journal of Teaching in Social Work 40.5 (2020).

[4] da Silva Maldonado, Everton, Emad Shihab, and Nikolaos Tsantalis. "Using natural language processing to automatically detect self-admitted technical debt." IEEE Transactions on Software Engineering 43.11 (2017).

[5] Zhao, Xiaojing, Ker-Wei Yeoh, and David Kim Huat Chua. "Extracting construction knowledge from project schedules using natural language processing." The 10th International Conference on Engineering, Project, and Production Management. Springer Singapore, 2020.

[6] Mohammadhassanzadeh, Hossein, et al. "Using natural language processing to examine the uptake, content, and readability of media coverage of a pan-canadian drug safety research project: Cross-sectional observational study." JMIR formative research 4.1 (2020).

[7] Yin, Wenshan, et al. "Abcnn: Attention-based convolutional neural network for modeling sentence pairs." Transactions of the Association for Computational Linguistics 4 (2016): 259-272.

[8] Li, Xinyu, and Dan Roth. "Learning question equivalence with the use of word embedding." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.

[9] Qiu, Lin, et al. "Learning text matching models with weak supervision." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.

[10] Wang, Shuohang, and Jing Jiang. "A compare-aggregate model for matching text sequences." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.

[11] Wang, William Yang, et al. "Bert has a mouth, and it must speak: BERT as a Markov random field language model." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. 2020.

[12] Xu, Liling, et al. "BERT-MMR: A New MMR Document Ranking Model with Pretrained Language Model." arXiv preprint arXiv:2103.09537 (2021).

[13] Seo, Minjoon, et al. "BERT with history data for long document understanding." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

[14] Zhou, Yuhao, et al. "BERT as a strong baseline for biomedical relation extraction." Proceedings of the 3rd Clinical Natural Language Processing Workshop. 2019.