Quora Question Similarity Detector Report

Avi Malhotra, Krishan Gupta, Om Sangwan, Nandani Yadav, Yash Joshi

INSY-669 - Text Analytics - Group Project

1 Introduction:

Quora, as a leading knowledge-sharing platform, faces the significant challenge of duplicate questions which can hamper the user experience and dilute the quality of information exchange. The project's core objective is to utilize the Quora Question Pairs dataset, a collection of over 400,000 potential question duplicate pairs, to create a model capable of detecting semantically similar questions.

2 Dataset Description:

The Quora Question Pairs dataset, hailing from the platform itself, is a rich repository of over 400,000 question pairs, each tagged with a binary flag indicating if they are duplicates. The dataset provides pairs of questions (question1, question2) along with a binary 'is_duplicate' label that signifies whether the pair contains two questions with the same intent.

3 Exploratory Data Analysis (EDA):

- Duplicate Distribution: We found more non-duplicate than duplicate questions, suggesting class imbalance.
- Question Length and Common Word Analysis: Questions are typically short, with a distribution peak at lower word counts. Duplicates tend to have more common words.
- Word Frequency and Uniqueness: There's a prevalence of stop words among the most common words, yet the dataset has a vast array of unique words, indicating rich semantic variety.
- Word Length Difference: Violin plots reveal that word count differences are not a definitive indicator of duplicates.

4 Methodology:

Our approach encompassed data preprocessing and feature engineering, vectorization and dimensionality reduction, followed by extensive model testing and evaluation. Key steps included data cleaning, feature extraction, and the application of TFIDF and BERT embeddings for vectorization.

4.1 Data Preprocessing and Feature Engineering

- Data Cleaning: Removing stopwords, handling punctuation, interpreting LaTeX expressions, and expanding contractions.
- Feature Extraction: Quantified aspects such as word commonality, question length, and word count differences between question pairs.

4.2 Vectorization and Dimensionality Reduction

- Lemmatization & Vectorization: Utilizing both Spacy and NLTK lemmatizers. Employed TFIDF for its emphasis on term uniqueness, and BERT embeddings for contextual nuances.
- Dimensionality Reduction: Applied Singular Value Decomposition (SVD) to TFIDF vectors.

4.3 Model Testing and Evaluation

A broad spectrum of models was initially tested to establish baseline performance. This included 6 models, with evaluations based on accuracy and F1 score metrics. Subsequent fine-tuning of the top-performing 3 models (Gradient Boosting, SVC, MLP) through grid search cross-validation aimed to optimize hyperparameters, and experimentation with different embedding with a particular focus on F-1 score.

Refer to Appendix Table 1 for initial model testing metrics and Table 2 for fine-tuned results of the top models.

4.4 Final Model Evaluation

The Gradient Boosting Classifier's efficacy was further affirmed through threshold optimization, showcasing robust performance across various metrics. The model was particularly optimized with a threshold of 0.3, balancing precision and recall effectively.

Refer to Appendix Table 3 for top model performance at different thresholds.

4.5 Pipeline Development and Deployment

A comprehensive pipeline using function and deployement via Streamlit has made our insights accessible and interactive.

5 Impact and Conclusion

5.1 Impact at Quora

Efficient content management directly contributes to operational cost reductions for platforms like Quora. By streamlining the process of content curation and moderation, resources can be better allocated, leading to potential savings. Furthermore, an improved user experience can drive user engagement, increasing the attractiveness of the platform for advertisers and thereby boosting revenue.

5.2 Beyond Quora

The utility of our approach extends to various other applications, including bot detection, plagiarism detection, document retrieval and search, question-answering systems, customer support chatbots, and legal and compliance analysis.

5.3 Conclusion

Our project not only addresses a critical issue faced by a leading knowledge-sharing platform but also lays the groundwork for future advancements in natural language processing and machine learning.

Appendix

Table 1 - Initial Model Testing Metrics

Model	Accuracy	F1	Precision
GradientBoosting	0.703000	0.589698	0.602802
KNN	0.593000	0.524799	0.466831
LogisticRegression	0.668625	0.539385	0.556679
MLP	0.685625	0.582382	0.572579
NaiveBayes	0.575750	0.588404	0.461285
SVC	0.684750	0.548456	0.584668

Table 1: Initial model testing metrics.

Table 2 - Fine Tuned Testing Metrics - Top Models

Model	Accuracy	F1 Score
SVC	NLTK Embeddings: 0.6865, Spacy Embeddings: 0.6870, NLTK Embeddings BERT: 0.6850, Spacy Embeddings BERT: 0.6805	NLTK Embeddings: 0.5246, Spacy Embeddings: 0.5229, NLTK Embeddings BERT: 0.5732, Spacy Embeddings BERT: 0.5731
MLP	NLTK Embeddings: 0.7005, Spacy Embeddings: 0.6945, NLTK Embeddings BERT: 0.6715, Spacy Embeddings BERT: 0.6820	NLTK Embeddings: 0.6067, Spacy Embeddings: 0.5951, NLTK Embeddings BERT: 0.5509, Spacy Embeddings BERT: 0.5875
GradientBoosting	NLTK Embeddings: 0.7185, Spacy Embeddings: 0.7135, NLTK Embeddings BERT: 0.6905, Spacy Embeddings BERT: 0.6950	NLTK Embeddings: 0.6044, Spacy Embeddings: 0.6051, NLTK Embeddings BERT: 0.5524, Spacy Embeddings BERT: 0.5643

Table 2: Fine-tuned testing metrics for top models.

Table 3 - Top Model Performance at Different Thresholds

Threshold	Precision	Recall	Accuracy	F-beta Score
0.3	0.60	0.78	0.76	0.62
0.25	0.55	0.88	0.73	0.59
0.35	0.57	0.66	0.73	0.58
0.45	0.59	0.53	0.73	0.57
0.4	0.56	0.62	0.72	0.57
0.2	0.48	0.91	0.66	0.53
0.5	0.57	0.41	0.71	0.52
0.55	0.58	0.34	0.71	0.51
0.6	0.60	0.28	0.71	0.49
0.65	0.70	0.22	0.72	0.49
0.7	0.80	0.13	0.71	0.38

Table 3: Top model performance at different thresholds.