Semantic Similarity Classification

Kristine Hambardzumyan
ASDS, YSU

1. Introduction

Online question and answer platforms such as Quora contain a large number of user generated questions, many of which express the same intent using different wording. Automatically identifying whether two questions are semantically equivalent, commonly referred to as duplicate question detection, is an important task for improving content organization, search efficiency, and user experience. Correctly identifying duplicate questions reduces redundant content, improves the organization of answers, and helps users access relevant information more efficiently.

From a machine learning perspective, duplicate question detection can be framed as a binary semantic similarity classification problem, where the goal is to determine whether two natural-language questions convey the same meaning. While early approaches relied on surface-level lexical similarity, such methods often fail when semantically equivalent questions use different vocabulary or sentence structure. This motivates the use of representation learning and deep learning models that can capture deeper semantic relationships.

This project examines the duplicate question detection task using the Quora Question Pairs (QQP) dataset from the GLUE benchmark. The project explores and compares two modern approaches:

1. a sentence embedding based similarity method,
2. a fine-tuned transformer based classification model.

The primary objective is to evaluate how well these methods capture semantic equivalence, analyze their strengths and limitations, and understand the trade-offs between simpler embedding based techniques and more expressive deep learning architectures. Model performance is evaluated using appropriate classification metrics, with particular emphasis on F1 score, which is well-suited for the moderately imbalanced nature of the dataset.

2. Related Work

Early approaches to duplicate question detection relied on surface-level lexical features such as word overlap, edit distance, and TF-IDF similarity. While efficient, these methods often fail when semantically equivalent questions use different wording or structure. The introduction of word embeddings, including Word2Vec and GloVe, improved semantic modeling by representing words in continuous vector spaces, though sentence representations built from averaged embeddings remained limited.

More recently, deep learning models such as RNNs, CNNs, and especially transformer based architectures have significantly advanced sentence-pair modeling. In particular, pretrained models like BERT capture contextual and relational information through self-attention and have become the standard for paraphrase and duplicate detection tasks. At the same time, sentence embedding models such as Sentence-BERT offer efficient semantic similarity computation for large-scale applications. This project builds on these developments by comparing a sentence embedding–based similarity method with a fine-tuned BERT classifier on the QQP dataset.

3. Methodology and Model Architecture

This project addresses duplicate question detection as a binary classification problem, where each input consists of a pair of questions (q_1, q_2), and the model predicts whether they express the same semantic meaning. Two complementary methodologies are explored: a sentence embedding based similarity approach and a transformer based classification model. This allows comparison between a lightweight semantic similarity method and a more expressive deep learning classifier.

3.1 Problem Formulation

Given two questions q_1 and 2, the task is to predict a binary label:

$$y = \begin{cases} 1, & \text{if } q_1 \text{ and } q_2 \text{ are duplicates} \\ 0, & \text{otherwise} \end{cases}$$

The models differ in how they represent the questions and how the final decision is produced.

3.2 Method 1: Sentence Embedding Based Similarity

Architecture

The first method uses a pretrained Sentence Transformer model to encode each question independently into a fixed-dimensional embedding vector. Let:

$e_1 = f(q_1), e_2 = f(q_2)$
where $f(\cdot)$ denotes the sentence embedding model.
In this project, embeddings are generated using a compact transformer based sentence encoder optimized for semantic similarity tasks.

Similarity and Decision Rule

The semantic similarity between the two questions is computed using cosine similarity:

$$\text{sim}(q_1, q_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{\|\mathbf{e}_1\| \, \|\mathbf{e}_2\|}$$

A decision threshold τ is applied:

$$\hat{y} = \begin{cases} 1, & \text{if } \text{sim}(q_1, q_2) \geq \tau \\ 0, & \text{otherwise} \end{cases}$$

The threshold τ is selected by maximizing the F1 score on the validation set. This tuning step is crucial because the similarity score is continuous, while the task requires a discrete classification.

Characteristics

This method is computationally efficient and does not require supervised fine-tuning of the encoder. However, it treats the two questions independently and relies on a fixed similarity metric, which may limit its ability to model complex interactions between sentence pairs.

3.3 Method 2: BERT based Sequence Pair Classification

Architecture

The second method employs a pretrained BERT model fine-tuned specifically for binary sequence pair classification. The two questions are concatenated into a single input sequence using BERT's standard format:

[CLS] q_1 [SEP] q_2 [SEP]

This combined sequence is processed by the BERT encoder, which uses multi-head self-attention to model contextual interactions between all tokens in both questions.
The final hidden representation corresponding to the [CLS] token is used as a pooled representation of the question pair. This vector is passed to a linear classification head:

$$\hat{y} = \text{softmax}(W\mathbf{h}_{\text{CLS}} + b)$$

where W and b are trainable parameters.

Training Procedure

The BERT model is fine-tuned end-to-end using cross-entropy loss on the training set. To mitigate overfitting and improve generalization, several regularization strategies are applied:

- Early stopping based on validation F1 score
- Weight decay to penalize large parameter values
- Label smoothing to reduce overconfidence on noisy labels

Training runs for only a few epochs, with early stopping applied to automatically stop the process once validation performance stops improving.

Characteristics

Unlike the embedding based approach, this model jointly processes both questions, allowing it to capture fine-grained semantic relationships and word-level interactions. While more computationally expensive, it is expected to achieve higher performance due to its expressive capacity and task-specific fine-tuning.

3.4 Model Selection and Evaluation Metric

Due to the moderate class imbalance in the dataset, F1 score is used as the primary metric for model selection and early stopping. While training and validation loss provide insight into optimization behavior, F1 score better reflects the quality of duplicate detection by balancing precision and recall.
The best model checkpoint is selected based on validation F1 score and subsequently evaluated on the held-out test set to obtain final performance estimates.

4. Experiments and Results

This section describes the experimental setup, data splits, evaluation protocol, and exploratory findings that guide the subsequent modeling decisions. Detailed model architectures and quantitative performance comparisons are presented in the following subsection.

4.1 Experimental Setup

All experiments were conducted using Python and the Hugging Face ecosystem, including the datasets and transformers libraries. Training and evaluation were performed in Google Colab with GPU acceleration when available. To ensure reproducibility and robustness against runtime interruptions, intermediate checkpoints, evaluation artifacts, and plots were saved to Google Drive.
The cleaned Quora Question Pairs dataset was split into train, validation, and test sets using stratified sampling to preserve the original label distribution. The validation set was used exclusively for hyperparameter tuning, threshold selection, and early stopping, while the test set was reserved for final evaluation.

4.2 Dataset Splits

After preprocessing and filtering, the dataset was divided as follows:

- Training set: 291,076 pairs
- Validation set: 36,385 pairs
- Test set: 36,385 pairs

The splits were constructed using a two-stage stratified split to maintain consistent class proportions across all subsets. The class distribution per split is shown in Table 1.

Split	Non-duplicate (0)	Duplicate (1)
Train	183,574	107,502
Validation	22,947	13,438
Test	22,947	13,438

Table 1: the class distribution per split

The identical class counts in the validation and test sets confirm that stratification was applied correctly. The dataset exhibits a moderate class imbalance, with non-duplicate pairs forming the majority class.

4.3 Evaluation Metrics

Given the class imbalance and the nature of the duplicate detection task, F1 score was selected as the primary evaluation metric. While accuracy is also reported for completeness, it can be misleading in imbalanced settings, as high accuracy may be achieved by favoring the majority class.
Precision and recall are additionally reported to provide insight into the trade-off between false positives (incorrectly identifying non-duplicates as duplicates) and false negatives (failing to identify true duplicates).
Confusion matrices are used to visualize classification behavior across classes.
Model selection and early stopping decisions are based on validation F1 score, ensuring alignment between training objectives and final evaluation criteria.

4.4 Exploratory Data Analysis Results

Exploratory data analysis (EDA) was conducted to better understand the structure of the dataset and to motivate modeling choices.

4.4.1 Class Distribution

Figure 1 illustrates the number of duplicate question pairs (label = 1) in each split. As expected, the training split contains a substantially larger number of samples, while validation and test splits have equal sizes and identical class proportions.

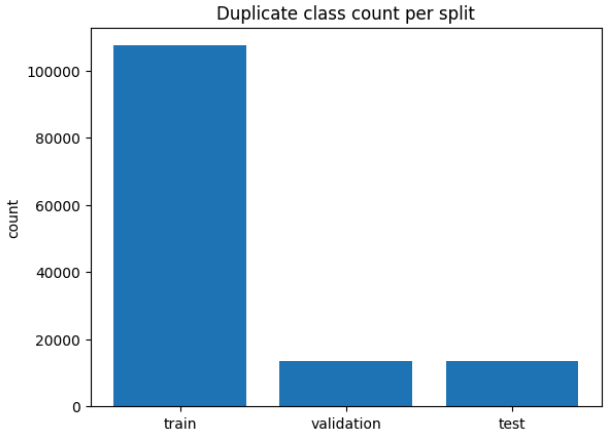

Figure 1: the number of duplicate question pairs in each split

This analysis confirms:

- consistent label distributions across splits,
- absence of data leakage,
- and the need for evaluation metrics robust to class imbalance.

4.4.2 Question Length Distribution

The total word length of each question pair was computed as the sum of the word counts in question1 and question2. Figure 2 shows the distribution of this total length for the training set.

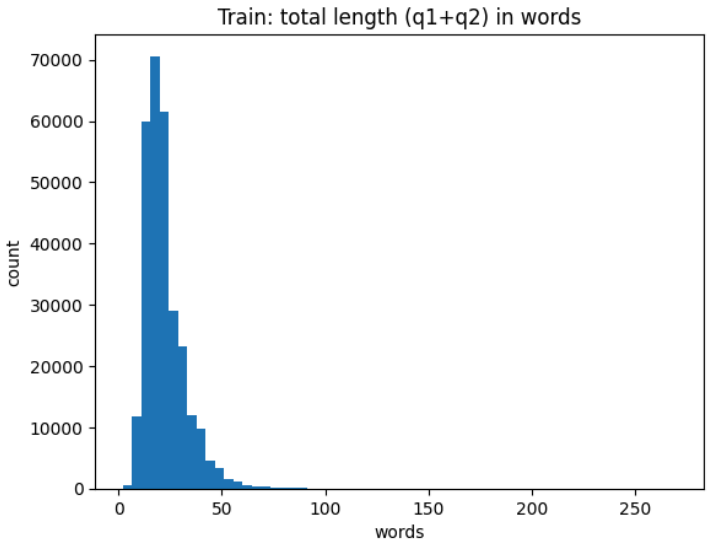

Figure 2: the distribution of this total length
for the training set

Most question pairs are relatively short, with the majority falling well below 60 words in total. The distribution is right-skewed, with a small number of longer question pairs forming a long tail. This observation supports the choice of a moderate maximum token length during tokenization, allowing efficient training without truncating most inputs.

4.4.3 Lexical Overlap Analysis

To assess surface-level similarity between question pairs, Jaccard overlap was computed on a random sample of 5,000 training examples. Figure 3 compares the overlap distributions for duplicate and non-duplicate pairs.

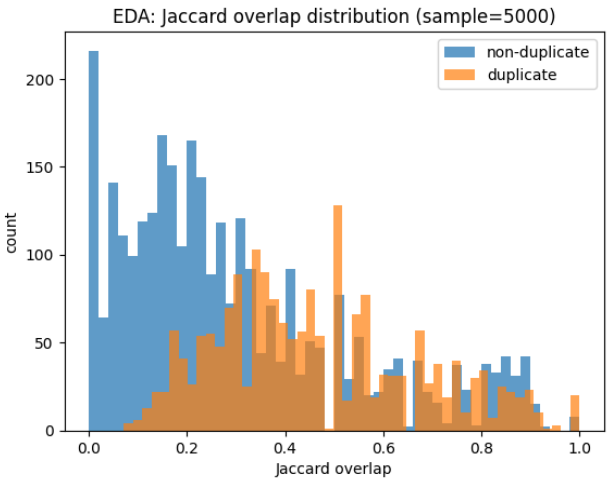

Figure 3: the overlap distributions for duplicate and non-duplicate pairs

Duplicate pairs generally exhibit higher lexical overlap, but there is significant overlap between the two distributions. Notably:

- many non-duplicate pairs share common words without being semantically equivalent,
- some duplicate pairs exhibit relatively low lexical overlap due to paraphrasing.

This result highlights the limitations of purely lexical similarity measures and motivates the use of semantic representations and deep learning models that can capture meaning beyond word overlap.

4.5 Summary of Experimental Context

The experimental setup establishes a controlled and reproducible environment for evaluating duplicate question detection methods. The EDA results reveal that:

- the dataset is moderately imbalanced,
- most question pairs are short,
- and lexical overlap alone is insufficient for reliable duplicate detection.

These findings justify the use of semantic embedding models and fine-tuned transformer architectures, which are evaluated in the subsequent sections.

4. Experiments and Results (Model Performance)

This subsection presents the quantitative results obtained using the two proposed methods: a sentence embedding based similarity approach and a fine-tuned BERT classifier. All results are reported on the held-out test set, which was not used during training, threshold tuning, or model selection.

4.6 Sentence Embedding Based Similarity Results

The first method uses a pretrained Sentence Transformer to encode each question independently into fixed-dimensional embeddings. Duplicate detection is performed by computing cosine similarity between embeddings and applying a decision threshold.

Threshold selection

The similarity threshold was tuned on the validation set by maximizing the F1 score. The optimal threshold was found to be:

- **Validation threshold:** τ=0.74
- **Validation F1:** 0.742

This threshold was then fixed and applied to the test set without further adjustment.

Test set performance
Using the selected threshold, the Sentence Transformer method achieved the following test performance:

Metric	Value
Accuracy	0.763
Precision	0.627
Recall	0.886
F1 score	0.734

Table 2: Sentence Transformer test performance

The corresponding confusion matrix (Figure 4) shows that this approach achieves high recall for duplicate pairs but suffers from a relatively large number of false positives, leading to lower precision.

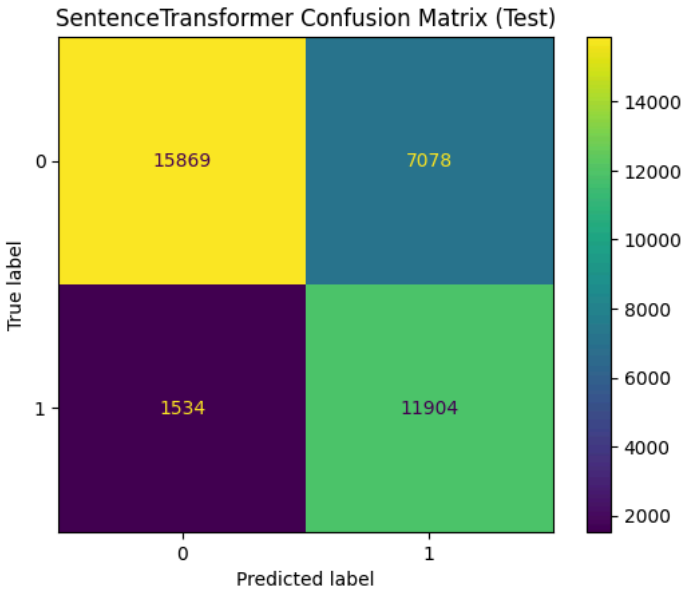

Figure 4: Sentence transformer confusion matrix

This behavior is expected for embedding based similarity methods, which rely on global semantic proximity and may incorrectly classify topically related but non-duplicate questions as duplicates.

4.7 BERT based Sequence Pair Classification Results

The second method fine-tunes a pretrained BERT model for binary sequence-pair classification, jointly encoding both questions and learning task-specific decision boundaries.

Training behavior

Training was conducted for a maximum of four epochs, with early stopping and model selection based on validation F1 score. Across epochs, training loss decreased steadily, while validation loss began to increase after early epochs, indicating mild overfitting. However, validation F1 continued to improve, justifying selection based on F1 rather than loss.

Test set performance

The fine-tuned BERT model achieved substantially stronger performance on the test set:

Metric	Value
Accuracy	0.907
Precision	0.856
Recall	0.898
F1 score	0.877

Table 3: BERT model test performance

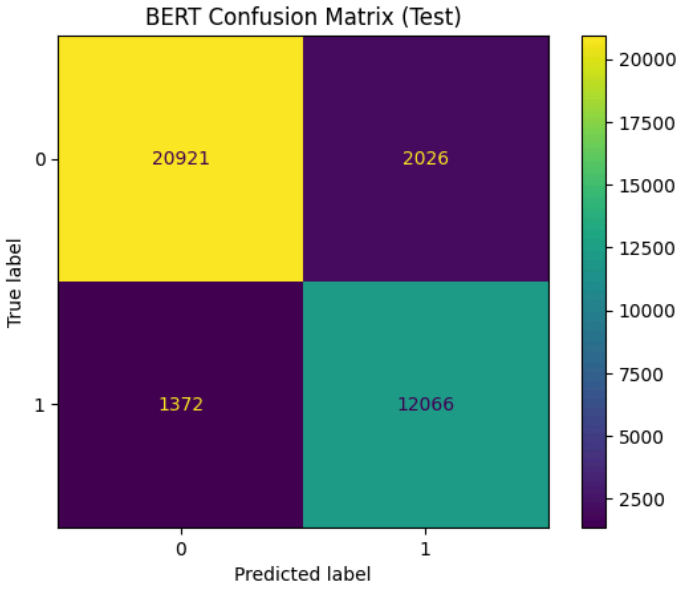

Figure 5: BERT confusion matrix

The confusion matrix (Figure 5) shows a significant reduction in both false positives and false negatives compared to the embedding based approach, indicating a better balance between precision and recall.

4.8 Comparative Analysis

Table 4 summarizes the test performance of both methods.

Model	Accuracy	Precision	Recall	F1
Sentence Transformer	0.763	0.627	0.886	0.734
BERT (fine-tuned)	0.907	0.856	0.898	0.877

Table 4: test performance of both methods

Figure 6 visualizes the F1 score comparison between the two approaches.

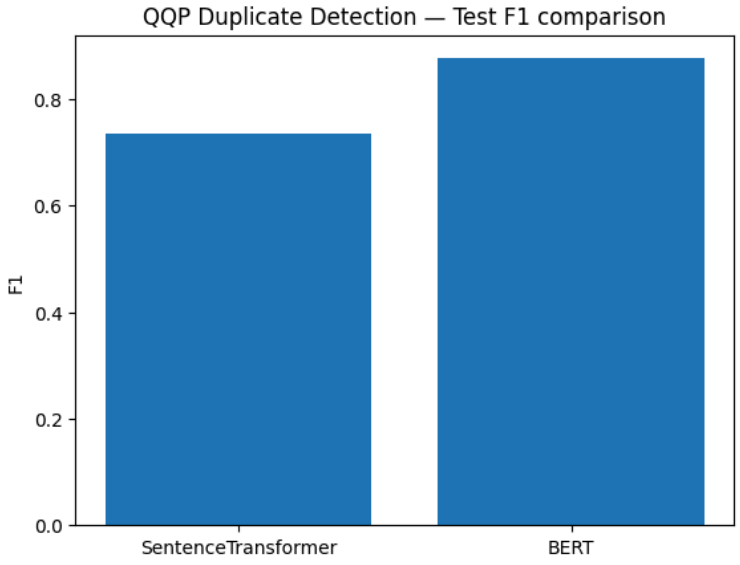

Figure 6: F1 score comparison between two approaches

The results demonstrate that:

- The Sentence Transformer method is efficient and achieves high recall but struggles with precision due to limited modeling of pairwise interactions.
- The BERT classifier significantly outperforms the embedding based approach across all metrics, particularly precision and overall F1 score.

This improvement can be attributed to BERT's ability to jointly model both questions and capture fine-grained semantic relationships beyond global sentence similarity.

4.9 Summary of Results

Overall, the experiments show that while sentence embedding based similarity provides a strong and computationally efficient baseline, fine-tuning a transformer model for sequence-pair classification yields substantially better performance on duplicate question detection. The observed improvements are consistent with prior findings in semantic similarity and paraphrase detection literature.

The next section discusses these results in more detail, including overfitting behavior, metric selection, and practical trade-offs between the two approaches.

5. Discussion

The experimental results clearly show the differences between embedding based similarity methods and fine-tuned transformer models for duplicate question detection.
The Sentence Transformer approach serves as a strong and computationally efficient baseline. Its high recall indicates that it successfully identifies many true duplicate question pairs. However, this comes at the cost of lower precision, as the model often labels semantically related but non-equivalent questions as duplicates. This behavior aligns with the exploratory data analysis, which showed that many non-duplicate pairs share substantial lexical overlap. Since sentence embeddings primarily capture overall semantic similarity, they are less effective at distinguishing true equivalence from general topical similarity.
In contrast, the BERT based classifier achieves consistently better performance across all evaluation metrics. By jointly encoding both questions and fine-tuning the model specifically for the duplicate detection task, BERT is able to capture more subtle semantic relationships and word-level interactions. This results in a noticeable reduction in both false positives and false negatives, as reflected in the confusion matrix, and leads to a significantly higher F1 score.
During training, validation loss increased after the initial epochs, while validation F1 continued to improve. This suggests mild overfitting in terms of probability calibration, but not in overall classification performance. As a result, model selection and early stopping were based on validation F1 score rather than loss, ensuring that training decisions remained aligned with the task objective. The use of regularization techniques such as weight decay, label smoothing, and early stopping helped control overfitting while maintaining strong performance.
Overall, the comparison between the two methods highlights an important trade-off. Embedding-based approaches are fast and simple, making them suitable for large-scale retrieval or candidate filtering. Transformer based classifiers, while more computationally expensive, provide substantially better accuracy. In real-world systems, these methods are often combined, using embeddings to narrow down candidates and a fine-tuned classifier for final decision making.

5.2 Conclusion

This project explored the task of duplicate question detection using the Quora Question Pairs dataset. Two approaches were evaluated: a sentence embedding–based similarity method and a fine-tuned BERT based sequence pair classifier.
The results show that although the embedding based approach offers an efficient and competitive baseline, the fine-tuned BERT model clearly outperforms it, achieving an F1 score of approximately 0.88 on the held-out test set. This improvement stems from BERT's ability to jointly model both questions and capture

deeper semantic relationships that go beyond surface-level similarity.
In conclusion, the experiments demonstrate the effectiveness of modern transformer based models for semantic similarity tasks and emphasize the importance of choosing evaluation metrics and training strategies that match the problem's goals. These findings highlight the practical value of fine-tuned transformer models for duplicate detection while also acknowledging the efficiency advantages of embedding based methods.

References

Quora Question Pairs (QQP) Dataset – GLUE Benchmark
https://gluebenchmark.com/tasks

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019).
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
https://arxiv.org/abs/1810.04805

Reimers, N., & Gurevych, I. (2019).
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
https://arxiv.org/abs/1908.10084