Duplicate Question Detection in Technical Q&A Forums

Viola Qiu
viola.qiu@berkeley.edu

William Shu
william.shu@berkeley.edu

Abstract

1 Introduction

Duplicate questions are a recurring challenge in large technical Q&A communities such as StackExchange. When multiple users independently describe the same underlying issue, they often use different terminology, error messages, or snippets of code. In result, answers become fragmented and harder to discover. Automatically identifying duplicates not only reduces redundancy for moderators, but also helps users find existing solutions more quickly, improving the overall efficiency of information retrieval.

Most existing research on duplicate-question detection focuses on the Quora Question Pairs (QQP) dataset, where questions are short, conversational, and often lexically similar. Technical forums differ substantially: posts are longer, contain domain-specific words, and may describe the same problem in different ways. As a result, it is unclear whether conclusions drawn from QQP could be generalized to technical centralized discussions.

In this project, we investigate duplicate detection in a technical domain using data derived from the sentence-transformers/stackexchange-duplicates published on HuggingFace. We analyze how linguistic complexity affects model performance by evaluating several modeling paradigms, ranging from simple models to modern transformer-based representations. In this project, we investigate duplicate detection in a technical domain using data derived from the sentence-transformers/stackexchange-duplicates corpus. We analyze how linguistic complexity affects model performance by evaluating several modeling paradigms, ranging from lexical similarity to modern transformer-based representations, on questions of varying length and structure. Our goal is to evaluate how different modeling approaches, ranging from lexical similarity to transformer-based architectures, perform on detecting technical duplicate questions, and to assess whether methods successful on QQP continue to perform well when applied to longer, domain-specific StackExchange posts.

2 Background

Duplicate question detection has been explored across many Q&A platforms, from general purpose forums to technical ones. Researchers use approaches from classical lexical models to modern transformer-based architectures. In general domains, datasets such as the Quora Question Pairs (QQP) collection have been widely used; (Sharma et al., 2019), for example, show that both classical and neural models can perform well on short, conversational question pairs. However, such datasets contain relatively simple language compared with technical environments, where posts often include code, error messages, and longer problem descriptions.

Transformer-based pretrained encoders have become central to semantic similarity tasks because they capture sentence-level meaning more effectively than lexical features. BERT (Devlin et al., 2019) first demonstrated strong performance on pairwise classification tasks through bidirectional pretraining. RoBERTa improves BERT's training procedure (Liu et al., 2019), while DeBERTa adds disentangled attention for better generalization (He et al., 2021). Sentence-BERT (Reimers and Gurevych, 2019) further adapts transformer encoders into a bi-encoder framework for efficient similarity scoring. ModernBERT (Inan et al., 2024) extends this line of work by optimizing memory use and supporting longer input sequences, which is particularly relevant for domains where posts can span several hundred tokens.

Technical Q&A sites such as StackExchange introduce additional complexity beyond what

general-purpose datasets capture. Questions may contain multi-step explanations, configuration details, or extensive logs, and duplicates can be phrased in structurally different ways. These observations motivate us to compare a range of lexical and pretrained transformer approaches on technical-domain data and evaluate if the existing approaches will still work well on detecting the technical specific questions.

Our study follows this direction by evaluating TF-IDF and linear regression as baseline, alongside several transformer families, including BERT, RoBERTa, DeBERTa, ModernBERT, and SBERT, on a reconstructed StackExchange duplicate question dataset, sentence-transformers/stackexchange-duplicates.

3 Methods

3.1 Dataset Construction

We construct our dataset from the publicly available sentence-transformers/stackexchange-duplicates dataset on HuggingFace, which provides positive duplicate pairs for three configurations: title pair, body pair, and post pair. Because the original dataset contains only duplicate pairs, additional processing is required to generate corresponding non-duplicate examples and produce a balanced classification dataset.

For each configuration, we treat every duplicate pair as an undirected edge in a graph whose nodes represent questions. We then extract connected components to be duplicate clusters: all questions within a cluster are all considered duplicate. This prevents accidental label leakage, since duplicates from the same cluster must not appear across different splits of the dataset.

Clusters are randomly partitioned into train (80%), validation (10%), and test (10%) sets. Positive pairs are kept as is. To construct negative examples, we sample an equal number of cross-cluster question pairs within each split–ensuring that sampled questions originate from different clusters and therefore represent distinct issues. This procedure provides balanced datasets for all three text types.

3.2 Baseline

As a baseline, we evaluate a TF-IDF representation combined with a Logistic Regression classifier. For each question pair, the two texts are concatenated with a separator token, and a TF-IDF vectorizer is applied to produce a sparse lexical representation. A Logistic Regression model is trained on these vectors to predict whether the pair is a duplicate. This baseline quantifies how far surface-level lexical similarity can go on technical Q&A data, where paraphrasing, long problem descriptions, and embedded code often limit the effectiveness of purely lexical approaches.

3.3 Transformer Cross-Encoder Models

We evaluate several pretrained transformer encoders using the standard cross-encoder formulation for sentence-pair classification. Each question pair is linearized as:

```
[CLS] question1 [SEP] question2 [SEP]
```

and processed jointly by the encoder. This allows every token in one question to attend to every token in the other, enabling rich cross-question interaction. The final hidden state of the classification token is passed to a feed-forward layer to predict whether the pair is a duplicate.

Following prior work on duplicate-question detection in non-technical domains, we fine-tune four encoder families: BERT-base (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), DeBERTa-base (He et al., 2021), and ModernBERT-base (Inan et al., 2024).

Maximum sequence lengths vary by dataset (e.g., shorter for titles, longer for bodies and full posts), with ModernBERT supporting the largest contexts due to its memory-efficient architecture. Each model is fine-tuned independently for title, body, and post duplicate detection.

3.4 SBERT Bi-Encoder Models

We also evaluate bi-encoder architectures using Sentence-BERT (SBERT; (Reimers and Gurevych, 2019)), which differs fundamentally from the cross-encoder formulation. Instead of jointly encoding the question pair, SBERT encodes each question independently into a fixed-dimensional embedding.

The two embeddings are then combined using concatenation and element-wise absolute difference, and passed to a small classification layer to predict whether the pair is a duplicate. Because the encoder processes each question separately, this approach removes token-level interaction between the questions, trading some expressive power for greater scalability and enabling embeddings to be cached or reused.

We fine-tune two SBERT variants reflecting different trade-offs between speed and representational capacity: all-MiniLM-L6-v2, a lightweight six-layer encoder optimized for efficiency, and all-mpnet-base-v2, a larger model designed to produce higher-quality embeddings. As with the cross-encoder models, each SBERT variant is trained independently on the title, body, and post datasets.

3.5 Training and Evaluation Setup

All models are fine-tuned separately for the title, body, and post datasets. We use the AdamW optimizer and apply consistent hyperparameters within each dataset type, adjusting only the maximum sequence length to accommodate input length differences (shorter for titles, longer for bodies and full posts). Early stopping based on validation loss is used to prevent overfitting.

Evaluation follows the cluster-aware splits described earlier: all questions belonging to the same duplicate cluster remain within the same split to avoid label leakage. Models are selected based on validation performance and evaluated on accuracy and F1 score, with F1 providing a more balanced measure in scenarios where false positives and false negatives carry asymmetric costs. Using three textual views, titles, bodies, and full posts, allows us to examine how model performance scales with increasing linguistic complexity.

4 Results and Discussion

Figure 1 compares model performance across the title, body, and post datasets. As expected, accuracy declines as inputs become longer and more variable, but ModernBERT and RoBERTa cross-encoders consistently outperform lexical and lighter transformer baselines, indicating the value of richer token interactions for technical text.

Figure 1: Model accuracy across the title, body, and post datasets. Performance drops on longer inputs, with ModernBERT retaining the highest scores.

Figure 2 highlights how F1 varies with question length. The cross-encoder models maintain stable precision and recall on short and medium examples, but performance decreases on the longest posts, suggesting further gains could come from length-aware truncation or hierarchical encoding.

Finally, Figure 3 compares the cross-encoder and bi-encoder paradigms. Cross-encoders achieve higher accuracy but at greater computational cost, while SBERT-based bi-encoders provide faster inference with a modest drop in quality. This trade-off suggests bi-encoders are suitable for retrieval-style prefiltering, with cross-encoders reserved for high-precision reranking.

5 Conclusion

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Hakan Inan and 1 others. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for efficient long-context modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Lav Sharma, Laurel Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *arXiv preprint arXiv:1907.01041*.

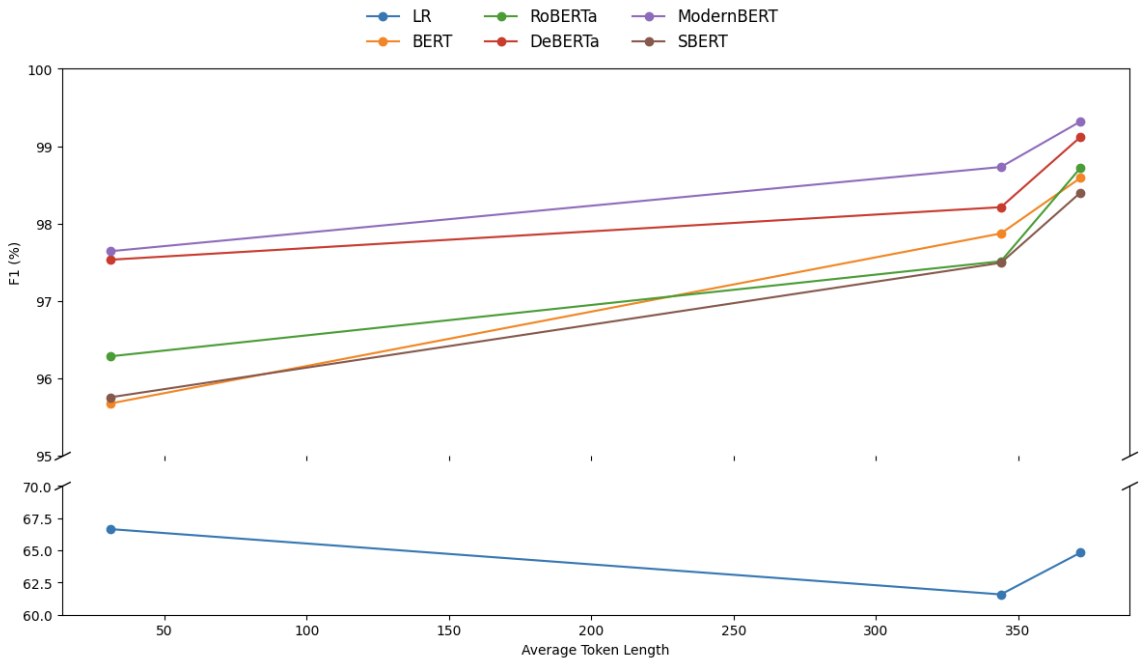

Figure 2: F1 score as a function of question length. Longer questions remain challenging despite modern encoders.

Contributions

- William Shu: dataset construction, SBERT experiments, DeBERTa cross-encoder experiments, GitHub repo setup, LaTeX setup, and drafting the introduction, background, and methods sections.
- Viola Qiu: EDA, baseline model, BERT-base experiments, RoBERTa cross-encoder experiments, ModernBERT experiments, and drafting the methods, Results and Discussion, Conclusion, and final paper submission.

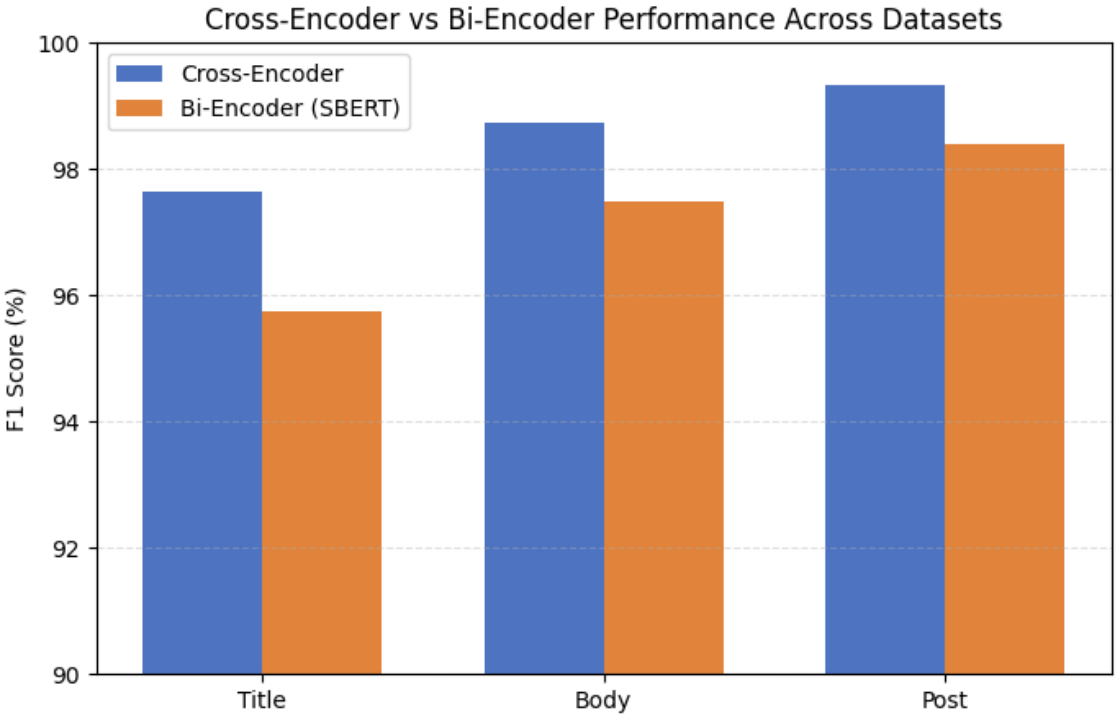

Figure 3: Cross-encoder vs. bi-encoder performance. Cross-encoders lead on accuracy; bi-encoders offer speed and scalability.