Duplicate Question Detection in Technical Q&A Forums

Viola Qiu
viola.qiu@berkeley.edu

William Shu
william.shu@berkeley.edu

Abstract

Duplicate question detection is essential for maintaining knowledge quality in large technical Q&A communities, where users may describe the same issue with different terminology, error messages, or code snippets. While prior work focuses mainly on general-domain datasets such as Quora Question Pairs (QQP), it is unclear whether these findings extend to longer and more complex technical posts. In this study, we evaluate lexical and transformer-based models on a reconstructed dataset derived from the sentence-transformers stackexchange-duplicates corpus. We examine titles, bodies, and full posts to assess how input length and linguistic complexity affect model performance. Transformer models substantially outperform a TF-IDF baseline, and long-context encoders achieve the highest accuracy on full posts. Our results show that although several trends observed in QQP carry over, technical-domain duplicate detection introduces length-dependent behaviors that highlight the need for domain-specific evaluation.

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. As a 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 domains.

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 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 (Warner and 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 (Warner and 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

4.1 Effects of Model Architecture and Text Length

Transformer models substantially outperform the TF-IDF baseline across all three datasets. As shown in Table 1, the baseline achieves only 61–67% F1, indicating that lexical overlap alone is not sufficient for identifying technical-domain duplicates. All transformer architectures exceed 95% F1, underscoring the importance of contextual representations for this task.

Model	Dataset	Max Len	Acc.	F1
TF-IDF + LR	Title	N/A	69.62%	66.65%
	Body	N/A	64.26%	61.58%
	Post	N/A	66.41%	64.83%
BERT-base	Title	64	95.71%	95.67%
	Body	512	97.87%	97.87%
	Post	512	98.59%	98.59%
RoBERTa-base	Title	128	96.31%	96.28%
	Body	512	97.52%	97.51%
	Post	512	98.73%	98.72%
DeBERTa-base	Title	128	97.52%	97.53%
	Body	256	98.21%	98.21%
	Post	256	99.12%	99.12%
ModernBERT-base	Title	256	97.64%	97.64%
	Body	1024	98.73%	98.73%
	Post	1024	99.32%	99.32%
SBERT (MiniLM-L6)	Title	128	95.77%	95.75%
	Body	256	97.49%	97.49%
	Post	256	98.40%	98.40%

Table 1: Performance of all models across Title, Body, and Post tasks. Transformer models outperform the lexical baseline by a large margin, with long-context encoders performing best on full posts.

Performance also varies with input length. Titles are the most challenging setting: they are short and often missing key details such as configuration steps, environment information, or error messages. All models obtain their lowest scores on titles. Bodies and full posts provide considerably more information and lead to consistent improvements across all transformer architectures. These longer descriptions contain additional signals—logs, parameter values, and multi-step explanations—that help distinguish between issues that appear similar on the surface. Figure 1 illustrates this trend across model families.

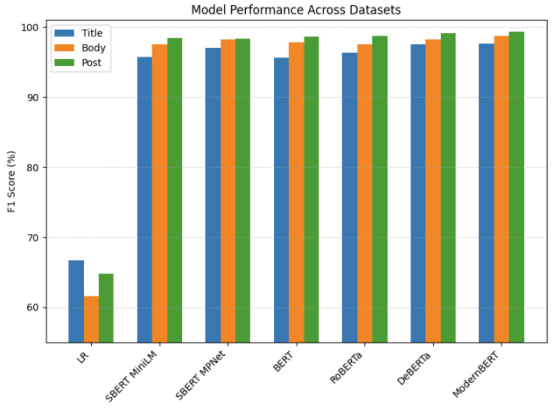

Figure 1: Model performance across the title, body, and post datasets. Transformer models outperform lexical baselines, with ModernBERT achieving the highest scores.

Among the evaluated architectures, ModernBERT benefits most from increased length. Its extended-context capacity allows it to retain information that other encoders truncate, resulting in the strongest performance on both bodies and full posts (up to 99.32% F1). This length-dependent behavior differs from general-domain datasets such as QQP, where questions are short and rarely require long-range reasoning. In technical Q&A forums, full-sequence context is often needed to understand the underlying issue, making models with longer context windows especially effective.

4.2 Accuracy–Efficiency Trade-offs

Cross-encoders achieve the highest overall accuracy, particularly on longer inputs. ModernBERT, which serves as the representative cross-encoder in Figure 2, obtains the strongest results across all datasets. Its full cross-attention mechanism enables rich token-level interactions between the paired questions, but this also leads to higher computational cost and slower inference.

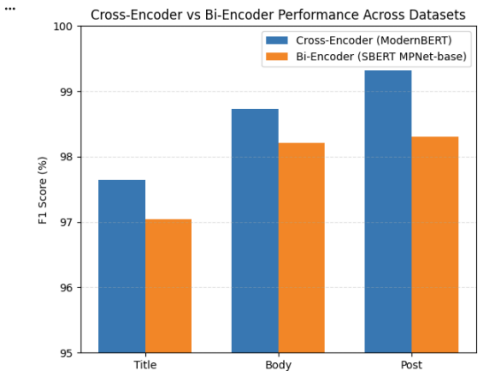

Figure 2: Comparison of cross-encoder and bi-encoder performance across datasets. Cross-encoders deliver higher accuracy, while bi-encoders offer greater efficiency.

Bi-encoders offer a different set of advantages. SBERT encodes each question independently and removes cross-attention, enabling efficient batched inference and large-scale retrieval. As shown in Figure 2, SBERT MPNet-base trails ModernBERT by only 0.6–0.7 F1 points on bodies and posts, despite being far more efficient to run. This modest gap makes bi-encoders appealing for retrieval-heavy or latency-sensitive applications.

Within the cross-encoder family, RoBERTa-base provides a reasonable compromise between performance and computational overhead. It matches or nearly matches the accuracy of DeBERTa-base while requiring fewer resources. For deployment scenarios with throughput constraints, RoBERTa offers a practical alternative to heavier long-context models.

4.3 Relation to Prior Work on General-Domain Duplicate Detection

Several trends observed in QQP carry over: transformer encoders outperform lexical baselines, and cross-encoders outperform bi-encoders. However, the technical domain introduces differences that shift performance patterns. Longer inputs improve accuracy, especially for long-context models. This effect does not appear in QQP, where question length is uniform and short. Lexical baselines are also weaker in the technical setting due to higher variability in terminology and problem framing.

Overall, transformer architectures transfer well across domains, but technical duplicate detection places greater emphasis on context length and domain-specific detail. As a result, general-domain benchmarks only partially predict performance in technical Q&A environments.

5 Conclusion

We investigated duplicate question detection in a technical Q&A domain using lexical and transformer-based models. All transformer architectures outperform the TF-IDF baseline, confirming that surface-level similarity is insufficient for technical posts. Performance varies by input length: models achieve their lowest scores on titles and improve steadily on bodies and full posts, with long-context encoders showing the largest gains. These trends differ from general-domain datasets such as QQP, where questions are short and less sensitive to context length. Cross-encoders remain the most accurate models, while bi-encoders provide competitive performance with lower computational cost. RoBERTa offers a strong accuracy–efficiency balance, and ModernBERT performs best when longer text is available. Overall, the results indicate that while transformer-based approaches transfer well across domains, technical duplicate detection requires attention to context length and domain-specific structure. Future work may explore retrieval–reranking pipelines, domain adaptation, and more fine-grained modeling of technical content such as code and error logs.

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*.

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*.

Benjamin Warner and et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for efficient long-context modeling. *arXiv:2412.13663*.

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.