CS6120 NLP: Enhanced Social Media Short Text Retrieval System

https://github.com/hua2980/CS6120_project.git

Hua Wang 002659330

Rongxuan Zhou 002308464

April 16, 2025

1. Abstract

This report presents our work on designing and evaluating a hybrid retrieval system that combines BM25 (a classical lexical-based method) and SBERT-based dense embeddings for passage retrieval. We fine-tuned a Sentence-BERT model (SBERT) on the MS MARCO dataset to improve retrieval relevance. We also explored STS-B (Semantic Textual Similarity Benchmark) as a pre-fine-tuning step. Experimental results show that the hybrid approach outperforms standalone BM25 or standalone SBERT in terms of retrieval metrics such as precision@k. Our findings highlight the potential of combining lexical and semantic retrieval methods for more robust search applications.

2. Introduction

Modern information retrieval (IR) faces the challenge of retrieving relevant passages from large text corpora efficiently. Lexical-based methods like BM25 leverage exact term matching, but can fail when users' queries and documents use different wording. Dense retrieval methods based on neural embeddings, particularly SBERT, can capture semantic similarity beyond surface keywords, though they may struggle with domain-specific terminology or large corpora without additional indexing structures.

Our objective is to compare and combine these methods to produce a more robust hybrid search pipeline.

Specifically, we:

1. Conducted data exploration on MS MARCO to understand passage lengths, query distribution, etc.
2. Fine-tuned SBERT on STS-B to improve general semantic similarity capture.
3. Further fine-tuned on MS MARCO for passage retrieval using a MultipleNegativesRankingLoss.
4. Combined the lexical scores (BM25) and dense scores (SBERT embeddings) in a hybrid manner, with an HNSW index for efficient dense retrieval.

These steps aim to demonstrate that a hybrid method can achieve better coverage and accuracy than either approach alone.

3. Methodology

3.1 Data Exploration

We began with an exploratory data analysis on two core datasets—MS MARCO v1.1 (training, validation, and test splits) and STS-B—to guide our subsequent retrieval system design and model configurations. Detailed codes and visualizations are provided in `1_data_exploration.ipynb`. Below is a summary of key observations.

3.1.1 MS MARCO Dataset Characteristics

The MS MARCO dataset contains a large volume of query–document pairs. We analyzed several aspects of its textual properties, including passage length, word count, and sentence structure:

1. Passage Length Distribution

   - The passage lengths in MS MARCO exhibit a right-skewed distribution, with the majority of passages ranging from 200 to 600 characters. A smaller fraction extend beyond 1000 characters.
   - Figure 1: Illustrates this right-skewed pattern, highlighting a moderate cluster of shorter passages alongside a long tail of lengthier ones.

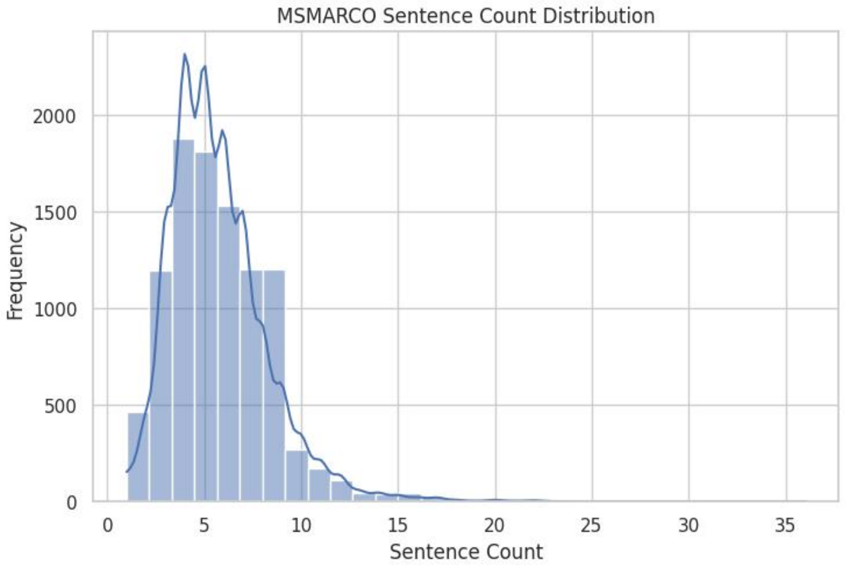

Figure 1

2. Word Count Distribution

- A similar trend emerges in the word count distribution. On average, each document contains roughly 100 words, indicating that MS MARCO passages are relatively short, making it suitable as a short-text retrieval benchmark.
- Figure 2: Shows that most passages lie in the 50–150 words range.

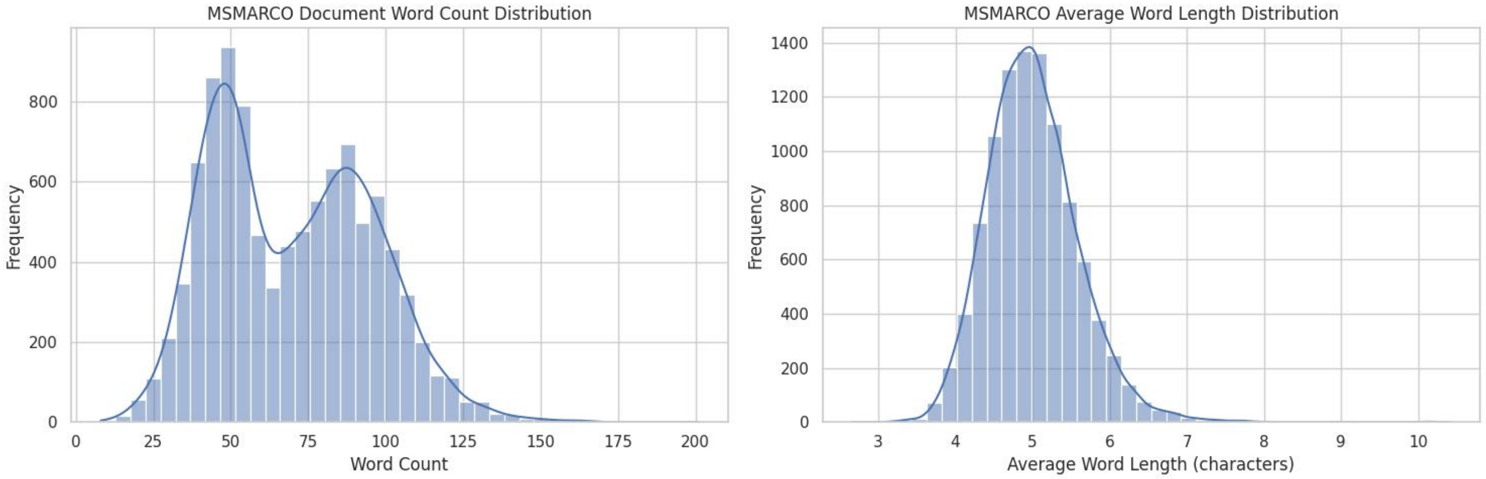

Figure 2

3. Sentence Structure

- On average, each passage includes about 4–5 sentences. These sentence lengths are moderate and generally consistent, facilitating sentence-level semantic encoding by transformer-based models (e.g., SBERT).
- Figure 3: Demonstrates the distribution of sentence counts per document, reflecting the typical structural features in MS MARCO passages.

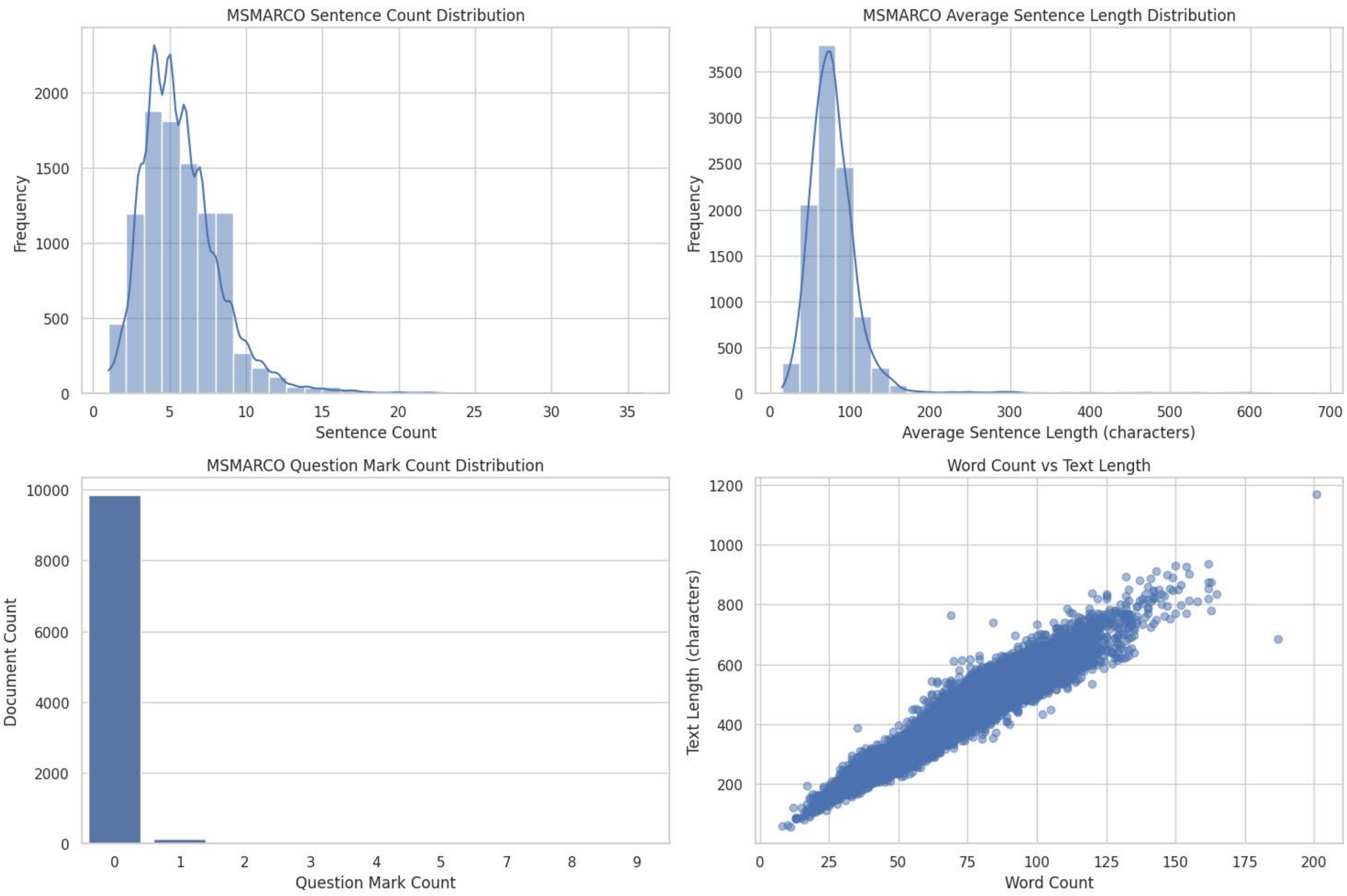

Figure 3

Overall, these data confirm that MS MARCO mostly consists of short to moderately long texts, aligning well with our intended hybrid retrieval approach (BM25 + SBERT). Furthermore, MS MARCO's queries (not shown here) tend to be short (often <10 words).

- Passage length distribution (avg. ~70 words).
- Query length distribution (short, often under 10 words).
- Basic statistics (min/max length, average word length, etc.).

3.1.2 STS-B Dataset Characteristics

In addition to MS MARCO, we also examined the STS-B (Semantic Textual Similarity Benchmark) dataset, which provides sentence pairs labeled with similarity scores. This dataset plays a key role in fine-tuning and evaluating sentence-level semantic models. We focused on two major aspects:

1. Similarity Score Distribution
    - STS-B assigns scores ranging from 0 (completely unrelated) to 5 (highly similar/equivalent). The distribution is relatively uniform across this range, providing a comprehensive set of examples for the model to learn different levels of semantic similarity.
    - Figure 4: Depicts how the 0–5 score range is covered, from very low similarity to near-equivalence.

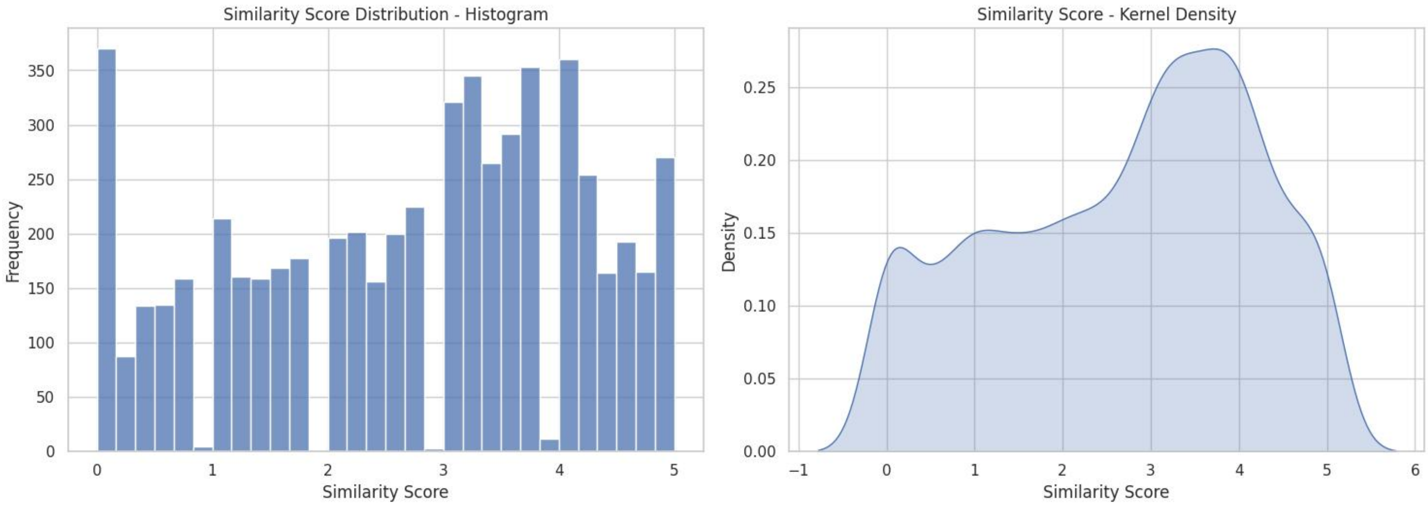

Figure 4

2. Sentence Length

- STS-B sentences tend to be short, with an average length of around 10–15 words, aligning with many social media or short-text scenarios.
- Figure 5: Reflects that most sentences cluster in a relatively narrow band (under ~20 words), which is highly compatible with typical transformer-based architectures.

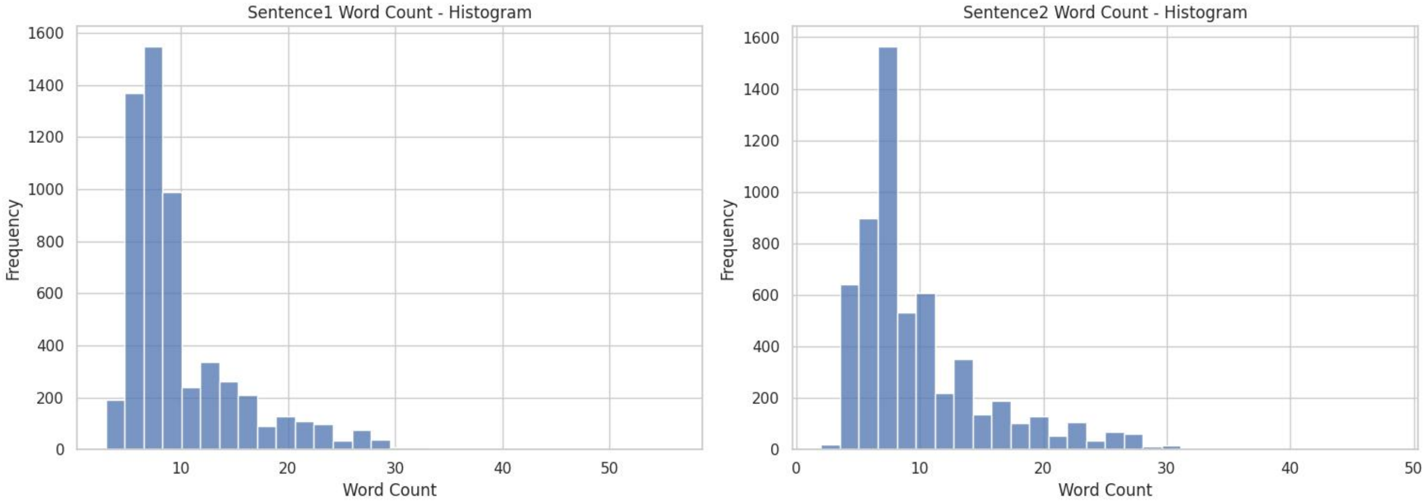

Figure 5

These insights from STS-B are critical for our retrieval system design, especially when training or fine-tuning SBERT to capture fine-grained semantic similarities. The balanced score distribution helps the model generalize across a wide range of semantic relatedness cases, while the relatively short sentence lengths ease computational overhead and tokenization requirements.

3.2 SBERT Fine-tuning

3.2.1 Pretrained Model Selection

We adopted Sentence-BERT (bert-base-nli-mean-tokens) as our base model for sentence-level representations, inspired by Reimers & Gurevych (2019) [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks]. This pre-trained checkpoint is already fine-tuned on Natural Language Inference (NLI) and has proven effective for tasks involving semantic similarity and sentence-level understanding.

3.2.2 Fine-tuning Process

Our fine-tuning process involved two phases:

1. Pre-training / Phase 1:
   - Dataset: STS-B (Semantic Textual Similarity Benchmark)
   - Loss Function: CosineSimilarityLoss
   - Motivation: STS-B provides gold similarity scores (0–5), which we normalize to [0, 1]. The CosineSimilarityLoss aligns the model's embedding-space distances with the human-annotated semantic similarity. Essentially, two sentences with higher similarity labels are pushed closer in the vector space, while dissimilar sentences are pushed apart.
   - Implementation: We used the standard approach from the SentenceTransformers library, training for several epochs to ensure the model captures fine-grained similarity signals.
2. Retrieval Fine-tuning / Phase 2:
   - Dataset: MS MARCO (queries and positive passages)
   - Loss Function: MultipleNegativesRankingLoss
   - Motivation: For retrieval tasks, the model must learn to rank a positive passage higher than all other negative passages for a given query. With MultipleNegativesRankingLoss, each batch contains one query and multiple candidate passages (one positive, others treated as negatives). The objective encourages the correct query–passage pair to have higher similarity than all negative pairs, making it well-suited for large-scale IR settings like MS MARCO.

3.2.3 Model Evaluation

After each fine-tuning step, we evaluated our model using SentEval tasks, notably:

- MRPC (Microsoft Research Paraphrase Corpus): Consists of sentence pairs from parallel news sources, labeled as paraphrase or not. We included MRPC to test how well our embeddings capture paraphrastic relationships—vital for retrieving passages that convey the same or similar meaning.
- TREC (Question Classification): A fine-grained question-type classification task involving broad question categories (e.g., location, numerical, etc.). We used TREC to test whether the embeddings can handle short, query-like inputs.

Implementation details and code appear in 2_sbert_finetuning.ipynb.

3.3 Hybrid Retrieval Pipeline

Finally, in our main pipeline (see Hybrid Retrieval & Evaluation.ipynb):

1. BM25 Index: We pre-processed and tokenized passages, then built a BM25 index.

2. SBERT + HNSW Index: We encoded all passages into dense vectors and stored them in an HNSW index for fast approximate nearest-neighbor search (cosine distance). The HNSW index, which supports approximate nearest neighbor search. This allows us to balance high retrieval accuracy with fast query response times, even over substantial corpora.

3. Fusion: For a given query, we retrieved top candidates from both BM25 and SBERT, normalized their scores, and combined them as:

$$final\_score = \alpha \times (BM25\ score) + (1 - \alpha) \times (SBERT\ score)$$

   where α is a hyperparameter.

4. Evaluation: We used the following metrics on a validation subset of MS MARCO:

   - Precision@k: The fraction of retrieved items (among the top k) that are relevant. This reflects how "precise" the system is when focusing on the highest-scoring results.

- Recall@k: The fraction of relevant items successfully retrieved within the top k. This helps assess how thoroughly the system uncovers relevant documents.
- nDCG (Normalized Discounted Cumulative Gain): A position-aware metric that rewards higher-ranked relevant documents more heavily than lower-ranked ones. It captures not only whether documents are relevant but also how they are ordered.
- MRR@100 (Mean Reciprocal Rank): Evaluates how quickly the first relevant document appears within the top 100 results. It is the average of the reciprocal ranks of the first relevant hit across all queries. If the first relevant document is ranked r, its contribution is $1/r$; if no relevant document appears in the top 100, the contribution is 0.

4. Literature Review

While we focused on the practical implementation, we also examined relevant literature:

1. Lexical-based Retrieval (BM25)

   Traditional approaches rely on exact keyword matching and document-term frequencies (Robertson & Zaragoza, 2009). Despite simplicity, BM25 remains a strong baseline in many IR tasks.

2. Dense Retrieval with Neural Embeddings

   Works like DPR (Karpukhin et al., 2020) and SBERT (Reimers & Gurevych, 2019) have demonstrated the advantages of learned embeddings for semantic matching.

3. Hybrid Approaches

   Research suggests combining lexical and semantic scores can address each method's shortcomings (e.g., Chen et al., 2022). Lexical matches capture domain-specific keywords, while dense embeddings capture context and paraphrases.

Across these studies, SBERT stands out for sentence-level semantic matching, and MS MARCO has long been a benchmark for IR tasks. Recent findings confirm that fine-tuning SBERT on retrieval data (like MS MARCO) further improves performance.

5. Critical Analysis

Our hybrid method leverages both exact-match (BM25) and semantic (SBERT) retrieval techniques, combined through a fixed-weight fusion parameter (α). The final results on a validation subset of MS MARCO are summarized in the table below:

Method	nDCG	MAP	MRR@100	Recall@100
stsb_finetuned_sbert_model + HNSW	0.2308	0.2308	0.2389	0.7379
msmarco_stsb_finetuned_sbert_model + HNSW	0.5589	0.4312	0.4387	0.9744
msmarco_finetuned_sbert_model + HNSW	0.5583	0.4305	0.4380	0.9739
msmarco_stsb_finetuned + Hybrid	0.5769	0.4490	0.4562	0.9891
msmarco_finetuned + Hybrid	0.5768	0.4488	0.4561	0.9891

5.1 Analysis of the Results

1. Fine-Tuning Benefits:
   The results indicate that fine-tuning SBERT on MS MARCO (and in combination with STS-B data) significantly improves retrieval performance compared to models fine-tuned solely on STS-B. For example, the msmarco_stsb_finetuned_sbert_model + HNSW yields higher nDCG, MAP, and MRR@100 than the STS-B-only fine-tuned model.

2. Hybrid Fusion Advantages:
   The hybrid method, which combines BM25 and SBERT scores, further boosts performance. The

msmarco_stsb_finetuned + Hybrid configuration reaches an nDCG of 0.5767 and a Recall@100 of 0.9891, outperforming both the BM25 baselines and the dense retrieval setups alone. This demonstrates that merging lexical and semantic signals can lead to improved ranking quality and broader coverage.

3. Metric Significance:

   - nDCG captures both the relevance and position of the retrieved items, rewarding systems that rank highly relevant documents at the top.
   - MAP (Mean Average Precision) reflects the overall precision across different recall levels.
   - MRR@100 (Mean Reciprocal Rank) indicates how quickly the first relevant document is found within the top 100 results.
   - Recall@100 measures the system's ability to retrieve the vast majority of relevant documents in the top 100 candidates.

4. Together, these metrics provide a comprehensive assessment of both ranking quality and retrieval coverage.

5.2 Limitations & Future Directions

1. Fixed Fusion Parameter (α):
   In our current approach, the fusion parameter α is set as a fixed hyperparameter. This static weighting may not be optimal across different queries or domains. Future work could explore dynamic weighting mechanisms that adjust α on a per-query basis, possibly leveraging query features or adaptive learning strategies.

2. Hyperparameter Tuning:
   Beyond α, more extensive hyperparameter tuning (e.g., BM25 parameters, HNSW configuration such as ef_search and M) could further refine performance, especially on large-scale corpora.

3. Domain Adaptation:
   Our experiments focused on general web-text; however, performance might vary in specialized domains (e.g., legal or medical texts). Future studies could investigate domain-specific fine-tuning and retrieval strategies.

4. State-of-the-Art Methods:
   Recent advancements like ColBERTv2 offer promising alternatives for dense retrieval and could be compared or integrated with our current hybrid method to push performance further.

5. Scalability: While we examined standard IR metrics, a deeper analysis of memory and computational costsunder truly massive datasets (hundreds of millions of documents) would be beneficial. HNSW and SBERT can scale well but require careful resource planning.

6. Conclusion

In summary, our experiments show that:

1. Hybrid retrieval consistently outperforms standalone BM25 or standalone SBERT in top-k retrieval metrics on a subset of MS MARCO.
2. Fine-tuning SBERT on STS-B before adapting to MS MARCO helps the model achieve better semantic understanding.
3. HNSW indexing is a practical approach to large-scale dense retrieval, though parameter tuning (e.g., ef_construction) is necessary for optimal recall.

Future Directions:

- Hyperparameter Optimization: Expand the search for the best αα (BM25 vs. SBERT weighting) and refine the HNSW index parameters.
- Domain-specific Extensions: Explore how the pipeline performs on specialized corpora.
- Explainability: Investigate methods to interpret and explain retrieval results to end-users.

7. References

1. MS MARCO Dataset: https://github.com/microsoft/MSMARCO
2. SentEval Toolkit: https://github.com/facebookresearch/SentEval
3. STSb_multi_mt Dataset: https://huggingface.co/datasets/PhilipMay/stsb_multi_mt
4. Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval.
5. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (EMNLP).
6. Svore, K. M., & Burges, C. J. C. (2009). A Machine Learning Approach for Improved BM25 Retrieval. Microsoft Research Technical Report MSR-TR-2009-92.13
7. Anserini. (2020). Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. SIGIR.
8. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).
9. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense Passage Retrieval for Open-Domain Question Answering. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).
10. Mandikal, P., Balachandran, V., Maas, A. L., & Shah, D. J. (2024). A Hybrid Approach to Enhance Scientific Document Retrieval. Proceedings of the AAAI-SDU Workshop.
11. OpenSourceConnections. (2023). Hybrid vigor: a winning way to use hybrid search. Technical Blog, February 27.12
12. Pinecone. (2025). Hybrid Search and Learning-to-Rank with Metarank. Technical Documentation, March 5.
13. Qdrant. (2021). Reranking in Hybrid Search. Technical Documentation.
14. Chen, Y., Johnson, T., Amitay, E., & Smith, N. A. (2022). Hybrid Approaches to Information Retrieval. Proceedings of the Association for Computational Linguistics (ACL).
15. Trautmann, D., Diallo, M., & Ferguson, S. (2022). Semantic Search with Sentence-BERT for Design Information Retrieval. NASA Technical Reports Server.
16. Ji, Z., Lu, Z., & Li, H. (2014). An Information Retrieval Approach to Short Text Conversation. arXiv preprint arXiv:1408.6988.
17. Meilisearch. (2024). Hybrid search: Definition, how it works, benefits and more. Technical Blog, March 13.