Evaluation Report on Embedding Models for Passage Retrieval

May 14, 2025

1 Introduction

The goal of this evaluation is to assess the performance of two embedding models in retrieving relevant passages given a query, a critical task in information retrieval systems such as question-answering or semantic search. Two models were evaluated: a baseline model (`l3cube-pune/bengali-sentence-similarity-sbert`) and a fine-tuned model (`d3_version`). The evaluation measures the models' ability to rank relevant passages correctly across various similarity thresholds and top-k retrieval settings, using a dataset of passages and associated questions stored in a JSON file.

2 Evaluation Setup

The evaluation process involves creating temporary vector stores using the Chroma vector database for each embedding model, embedding passages from the JSON dataset, and querying the vector store with questions to retrieve relevant passages. The following metrics were computed to assess performance:

- **Recall@k**: The proportion of queries where the relevant passage is retrieved within the top-k results.
- **Precision@k**: The average precision for queries where the relevant passage is within the top-k results, calculated as $\frac{1}{\text{rank}}$ if the relevant passage is found, else 0.
- **F1@k**: The harmonic mean of precision and recall, calculated as $F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.
- **MRR@k**: The Mean Reciprocal Rank, which measures the inverse of the rank of the first relevant passage within the top-k results.
- **Average Top-1 Similarity Score**: The mean cosine similarity score of the top-ranked passage for each query.

The evaluation was conducted across similarity thresholds ($T = 0.5, 0.6, 0.7, 0.8, 0.9$) to filter retrieved passages and top-k values ($k = 1, 3, 5$) to assess ranking perfor-

mance. A Precision-Recall curve with F1 annotations was generated, and metrics were saved to a JSON file for further analysis.

3 Evaluation Results

The evaluation results for both models are summarized below, highlighting performance across thresholds and top-k settings.

3.1 Baseline Model (`l3cube-pune/bengali-sentence-similarity-sbe`

The baseline model achieved near-perfect performance for thresholds $T = 0.5$ to $T = 0.8$, with the following metrics for all $k = 1, 3, 5$:

- Recall@k: 1.0000
- Precision@k: 1.0000
- MRR@k: 1.0000
- F1@k: 1.0000
- Avg Top-1 Similarity: 0.9319

At $T = 0.9$, performance slightly declined:

- Recall@1,3,5: 0.9434
- Precision@1,3,5: 0.9434
- MRR@1,3,5: 0.9434
- F1@1,3,5: 0.9434
- Avg Top-1 Similarity: 0.9346

This indicates that the baseline model struggles to maintain high performance when stricter similarity thresholds are applied, likely due to lower embedding quality for certain passages.

3.2 Fine-Tuned Model (`d3_version`)

The fine-tuned model (`d3_version`) also achieved perfect scores for thresholds $T = 0.5$ to $T = 0.8$, matching the baseline:

- Recall@k: 1.0000
- Precision@k: 1.0000
- MRR@k: 1.0000
- F1@k: 1.0000
- Avg Top-1 Similarity: 0.9389

At $T = 0.9$, the fine-tuned model outperformed the baseline:

- Recall@1,3,5: 0.9671
- Precision@1,3,5: 0.9671
- MRR@1,3,5: 0.9671
- F1@1,3,5: 0.9671
- Avg Top-1 Similarity: 0.9407

The fine-tuned model demonstrates a significant improvement at the highest threshold ($T = 0.9$), with a 2.37% increase in recall, precision, MRR, and F1 scores compared to the baseline (0.9671 vs. 0.9434).

3.3 Comparison and Improvements

The fine-tuned model (`d3_version`) shows clear improvements over the baseline, particularly at the strictest similarity threshold ($T = 0.9$):

- **Higher Robustness**: The fine-tuned model maintains higher performance (0.9671 vs. 0.9434 across all metrics) when only passages with high similarity scores are considered, indicating better embedding quality and relevance ranking.
- **Improved Similarity Scores**: The average top-1 similarity score is consistently higher for the fine-tuned model (0.9389–0.9407 vs. 0.9319–0.9346), suggesting that the fine-tuning process enhanced the model's ability to produce more discriminative embeddings.
- **Error Handling**: The fine-tuned model encountered errors during vector store creation for specific entries (e.g., annotation ID `d5f877c8-963e-4a96-925d-6e2c`), which may indicate dataset inconsistencies but did not prevent overall superior performance.

4 Discussion

Both models perform exceptionally well at lower thresholds ($T \leq 0.8$), achieving perfect scores across all metrics. However, the fine-tuned model's advantage becomes evident at $T = 0.9$, where it retrieves relevant passages more effectively. This improvement is likely due to fine-tuning on domain-specific data, which enhances the model's ability to capture semantic nuances in the Bengali language dataset. The higher average top-1 similarity scores further confirm that the fine-tuned model produces embeddings that better align queries with relevant passages.

The evaluation highlights the importance of threshold selection in retrieval systems. Lower thresholds ($T \leq 0.8$) ensure high recall but may include less relevant passages, while higher thresholds ($T = 0.9$) prioritize precision at the cost of missing some relevant passages. The fine-tuned model's ability to maintain high performance at $T = 0.9$ makes it more suitable for applications requiring precise retrieval.

5 Conclusion

The evaluation demonstrates that both the baseline and fine-tuned models are highly effective for passage retrieval, but the fine-tuned model (`d3_version`) offers significant improvements, particularly at high similarity thresholds. Its superior performance at $T = 0.9$, with a 2.37% increase in key metrics and higher similarity scores, makes it a better choice for applications demanding precise and robust retrieval. Future work could explore further fine-tuning or address dataset inconsistencies to eliminate errors during vector store creation.