Efficient Multi-Vector Retrieval with Reduced Index Size

Lam Do
lamdo@illinois.edu

Aaditya Bodke
abodke2@illinois.edu

1 Research Question

We investigate how to significantly reduce the index size in multi-vector retrieval models while maintaining or improving retrieval effectiveness. Specifically, we aim to develop a new representation and scoring mechanism that preserves the core advantages of multi-vector models—namely, their capacity to capture nuanced token-level interactions—without incurring prohibitive storage costs.

2 Significance

While multi-vector retrieval approaches like ColBERT (Khattab and Zaharia, 2020) have delivered state-of-the-art results, they are rarely deployed in large-scale applications due to the high cost of storing multiple vectors per document token. This constraint becomes even more pronounced when dealing with massive corpora. By addressing this problem, we can potentially unlock the widespread adoption of these powerful models in settings where storage space is a critical limitation (e.g., commercial search engines, data-intensive research labs, or low-resource environments). A more efficient multi-vector model could have a transformative impact, enabling faster query responses, more economical hardware usage, and a broader range of application scenarios.

3 Novelty

There has been existing work that aim to address this question. However, they are not able to substantially reduce index size of the model while preserving the performance. In particular, existing work can reduce only up to 40% of the index size while preserving the performance. This is an observation we made empirically, and there are also previous work that agree with our observation (Formal et al., 2020).

4 Approach

Our approach is grounded in two key observations:

- Firstly, in ColBERT, key tokens often perform exact matching, suggesting that sparse (or even one-hot) vectors might be sufficient for these tokens. This could significantly reduce storage requirements.
- Secondly, ColBERT uses a *sum of max* ranking function, while SPLADE (Formal et al., 2021) employs a *sum of sum*. By reformulating SPLADE's ranking function as sum of max, we can potentially retain the benefits of multi-vector representations while reducing complexity.

We propose to treat SPLADE as a multi-vector model where each document token is represented by an extremely sparse vector (potentially one-hot). The model will be trained using a sum of max objective function to leverage the strengths of both ColBERT and SPLADE.

5 Evaluation

We will evaluate the proposed model on BEIR benchmarks [1], MS MARCO Dev [2], and potentially LoTTE dataset [3]. Evaluation metrics: nDCG@10 for BEIR benchmarks, MRR@10 for MS MARCO Dev and Success@5 for LoTTE.

6 Timeline

- Week 1: Complete literature review and refine the proposed hybrid model design.
- Week 2-3: Implement the core components and conduct preliminary experiments .

[1] https://github.com/beir-cellar/beir
[2] https://microsoft.github.io/msmarco/Datasets.html
[3] https://github.com/stanford-futuredata/ColBERT/blob/main/LoTTE.md

- Week 4: Run full-scale experiments on MS MARCO Dev, BEIR, and LoTTE.

7 Task Division

- Lam Do (Coordinator): Oversees progress, leads experimental design, ensures timely completion of evaluations, and consolidates the final report.

- Aaditya Bodke: Implements model components, manages data preprocessing, runs experiments, and assists with data analysis.

References

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2020. A white box analysis of colbert. *Preprint*, arXiv:2012.09650.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. *Preprint*, arXiv:2107.05720.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Preprint*, arXiv:2004.12832.