TRIAL: Token Relations and Importance Aware Late-interaction for Accurate Text Retrieval

Hyukkyu Kang
POSTECH
hkkang@dblab.postech.ac.kr

Injung Kim
Handong Global University
ijkim@handong.edu

Wook-Shin Han[*]
POSTECH
wshan@dblab.postech.ac.kr

Abstract

Late-interaction based multi-vector retrieval systems have greatly advanced the field of information retrieval by enabling fast and accurate search over millions of documents. However, these systems rely on a naive summation of token-level similarity scores, which often leads to inaccurate relevance estimation caused by the tokenization of semantic units (e.g., words and phrases) and the influence of low-content words (e.g., articles and prepositions). To address these challenges, we propose TRIAL: Token Relations and Importance Aware Late-interaction, which enhances late interaction by explicitly modeling token relations and token importance in relevance scoring. Extensive experiments on three widely used benchmarks show that TRIAL achieves state-of-the-art accuracy, with an nDCG@10 of 46.3 on MSMARCO (in-domain), and average nDCG@10 scores of 51.09 and 72.15 on BEIR and LoTTE Search (out-of-domain), respectively. With superior accuracy, TRIAL maintains competitive retrieval speed compared to existing late-interaction methods, making it a practical solution for large-scale text retrieval.

1 Introduction

Text retrieval is a long-standing problem in both information retrieval (IR) and natural language processing (NLP), with broad applications ranging from search engines to question-answering systems. The rise of Large Language Models (LLMs) has further underscored the importance of accurate retrieval, as retrieval models play a key role in Retrieval-Augmented Generation (RAG) by supplying relevant context for reliable responses (Lewis et al., 2020).

Text retrieval poses a significant challenge, as it demands both precise measurement of semantic similarity between queries and documents and the ability to maintain high retrieval speeds. In natural language, the same semantics can be expressed with different words and phrases, limiting the effectiveness of lexical matching. Thus, semantic retrieval must capture nuanced meanings, context, and relationships between words and phrases, which is complex and computationally demanding.

Recent advances in pre-trained language models (PLMs) have enabled neural retrievers to surpass traditional methods like BM25 (Robertson et al., 1995). Early approaches used cross-encoders to jointly encode queries and documents for semantic similarity estimation (Qiao et al., 2019; Reimers and Gurevych, 2019), but their high computational cost constrains scalability. This led to dual-encoder models like DPR and MDR (Karpukhin et al., 2020; Xiong et al., 2020), which encode queries and documents independently into single-vector embeddings, enabling faster retrieval. To further improve accuracy without sacrificing speed, late-interaction models such as ColBERT (Santhanam et al., 2022b), COIL (Gao et al., 2021), CITADEL (Li et al., 2023), and XTR (Lee et al., 2024) have been introduced. These models generate multi-vector token-level representations, striking a balance between accuracy and efficiency.

Despite achieving state-of-the-art accuracy, existing late-interaction methods have two major limitations caused by their naive summation of token-level similarities between query vectors E_q and document vectors E_d:

$$\text{score}(Q, D) = \sum_{i=1}^{|Q|} \max_{j=1}^{|D|} \left(E_{q_i} \cdot E_{d_j}\right) \qquad (1)$$

The first limitation is that the naive token-level summation can lead to incorrect relevance scores when words or phrases are fragmented into multiple tokens. For instance, "Scott Derrickson" may be tokenized as "scott", "derrick", and "##son", which are then matched independently. Even with contextualized embeddings (e.g., BERT), this can yield high scores for documents containing those

[*]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 16864–16877
November 4-9, 2025 ©2025 Association for Computational Linguistics

tokens in unrelated contexts. The second limitation is their inability to account for term importance, a concept emphasized in traditional methods like TF-IDF and BM25. To address these limitations, we propose **TRIAL**: **T**oken **R**elations and **I**mportance **A**ware **L**ate-interaction. TRIAL explicitly models token relations to capture semantic dependencies between related tokens, reducing noise from fragmented tokens by ensuring multi-token entities (e.g., noun phrases) are treated as cohesive semantic units during relevance computation. Additionally, TRIAL incorporates a weighting mechanism that predicts the importance of each query token, effectively prioritizing critical terms in relevance scoring.

Our main contributions are as follows:

- We propose TRIAL, a token relations and importance aware late-interaction method that improves the accuracy of relevance scoring for text retrieval.
- We perform an extensive evaluation of TRIAL on both in-domain and out-of-domain datasets (i.e., MSMARCO, BEIR, and LoTTE), demonstrating its effectiveness across diverse settings.
- We provide a detailed analysis of TRIAL, showing how it addresses the challenges of phrase fragmentation and term importance in existing late-interaction methods

2 Related Work

We categorize recent neural text retrieval methods into sparse and dense approaches. Then, we further divide dense retrieval systems based on their query-document interaction mechanisms: full-interaction, representation-based, and late-interaction.

Sparse Retrieval. Recent efforts to improve traditional retrieval methods like TF-IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 1995) have focused on integrating neural learning to produce better sparse representations. DeepCT (Dai and Callan, 2020) uses PLMs (e.g., BERT) to assign contextualized term weights, bridging conventional and neural term weighting for improved accuracy. SparTerm (Bai et al., 2020) enhances sparse encoding by learning term importance and applying activation gating, balancing semantic and exact matching. The SPLADE family (Formal et al., 2021b,a; Lassance and Clinchant, 2022) introduces sparsity regularization and term expansion. SPLADEv2 incorporates hard negatives and distillation for state-of-the-art BEIR performance, while SPLADE++ achieves BM25-level latency using FLOPS-regularized training and decoupled encoders.

Full-Interaction Dense Retrieval. Full interaction methods, also known as cross-encoders, jointly encode query–document pairs using PLMs to capture rich token-level interactions and achieve high accuracy (Qiao et al., 2019; Reimers and Gurevych, 2019). MonoBERT (Nogueira and Cho, 2019) fine-tunes BERT to produce a single relevance score for query–passage pair. CELI (Zhang et al., 2024) adds interaction layers to improve out-of-domain generalization. H-ERNIE (Chu et al., 2022) leverages hierarchical encoding and word-phrase alignment for Chinese-language retrieval, demonstrating the applicability beyond English. However, the high computational cost of full-interaction methods limits scalability, so they are typically used as rerankers following lightweight retrievers (e.g., BM25).

Representation-Based Dense Retrieval. Representation based methods use separate encoders to map queries and documents into single dense vectors for fast retrieval. DPR and MDR (Karpukhin et al., 2020; Xiong et al., 2020) are foundational works that precompute document embeddings and facilitate real-time query-document similarity comparison over millions of documents. Dense X (Chen et al., 2024) introduces proposition-level retrieval to improve question-answering performance. ANCE (Xiong et al., 2021) enhances retrieval accuracy by leveraging global hard-negative sampling through an Approximate Nearest Neighbor (ANN) index. RocketQA (Qu et al., 2021) refines training with cross-batch negatives, denoised hard examples, and data augmentation. DensePhrases (Lee et al., 2021) reformulated open-domain question answering as a phrase retrieval task, learning dense phrase representations for efficient retrieval. GTR (Ni et al., 2022) scales dual encoders and shows that larger models improve generalization even with fixed embedding sizes.

Late-Interaction Dense Retrieval. Late-interaction methods use separate encoders to map queries and documents into multiple dense vectors, enabling efficient token-level similarity computation at inference. ColBERT (Khattab and Zaharia, 2020) introduces MaxSim operation (i.e., Chamfer similarity) for efficient token-to-token comparisons, striking a balance between speed and accuracy. COIL (Gao et al., 2021) optimizes retrieval by matching only identical tokens, while CITADEL (Li et al., 2023) uses lexical routing to relax this constraint. ColBERTv2 (Santhanam et al., 2022b) enhances ColBERT with cross-encoder distillation and quantization. XTR (Lee et al., 2024) simplifies scoring, and ColBERTer (Hofstätter et al., 2022) reduces vector counts with a Neural BoW. LITE (Ji et al., 2024) combines learnable scorers with dual encoders, significantly reducing storage and latency.

3 Method

3.1 Problem Formulation

Let $\mathbb{Q}$ denote the set of possible text queries and $\mathbb{D}$ the set of all text documents. Given a query $Q \in \mathbb{Q}$ and a corpus $C = \{D_1, \ldots, D_m\} \subseteq \mathbb{D}$, the goal of document retrieval is to rank all documents in C according to their relevance to Q. Relevance may be defined as providing the most appropriate answer for a question-type query or being the most contextually aligned for a keyword-based query. Formally, the task is to learn a scoring function $score(\cdot, \cdot) : \mathbb{Q} \times \mathbb{D} \rightarrow \mathbb{R}$ such that, for a given query Q, documents can be ranked by their relevance scores:

$$\text{rank}(C \mid Q) = \operatorname*{argsort}_{D \in C} \text{score}(Q, D) \tag{2}$$

Many neural methods approximate the relevance distribution over the corpus via a softmax transformation of the scores:

$$P(D \mid Q) = \operatorname*{softmax}_{D \in C}(\text{score}(Q, D)) \tag{3}$$

The objective of the retrieval model is to ensure that higher relevance scores are assigned to more relevant documents, enabling accurate ranking.

3.2 TRIAL

We propose an advanced scoring function for late-interaction retrieval systems that models *token relation score* and *token importance weight*. The scoring function is formulated as follows:

$$\begin{aligned}\text{score}(Q, D) := \sum_{i=1}^{|Q|} w_{q_i} \max_{j=1}^{|D|} \Big(& \text{sim}\big(E_{q_i}, E_{d_j}\big) \\ & + \lambda \, \text{sim}\big(\text{Rel}(q_i, q_{i-1}), \text{Rel}(d_j, d_{j_{i-1}})\big)\Big)\end{aligned} \tag{4}$$

Here, q_i and d_j denote the i^{th} and j^{th} tokens in the query Q and document D, respectively. E_{q_i} and E_{d_j} are contextualized token embeddings that capture semantic meaning, and w_{q_i} represents the importance weight of query token q_i. The document token d_{j_i} denotes the token that best aligns with query token q_i. For the initial document token d_{j_0}, we use the CLS token. The function $\text{Rel}(\cdot, \cdot)$ encodes the contextual relation between two tokens. For instance, the term $\text{Rel}(q_i, q_{i-1})$ represents the relation between consecutive query tokens. The hyperparameter λ controls the contribution of token relation scores to the overall score. Throughout our experiments, we set $\lambda = 0.5$, unless stated otherwise.

By explicitly modeling token-level relations and assigning weights to tokens based on their importance, TRIAL can calculate more accurate relevance scores for queries with multi-token words/phrases and functional words (e.g., articles and prepositions).

3.3 Token Relation Score

We introduce the token relation score to explicitly model the similarity between the token relations in queries and documents. The token relation scores enable the final query-document relevance score to consider the semantic dependencies between tokens, allowing for improved handling of multi-token words and phrases.

To capture dependencies between tokens with minimal overhead, we model relations using *query-side bigrams*. Specifically, for each token q_i, we compute a relation with its preceding token q_{i-1}, which acts as a contextual anchor. To compare with relations between document tokens, we compare this query relation to one formed by the two document tokens most similar to q_i and q_{i-1}. This bigram-based approach keeps the relation representation compact and computationally efficient, as the potential gains from higher-order relations were negligible.

Formally, to encode the relation between two tokens t_1 and t_2, we begin by concatenating their embedding vectors E_{t_1} and E_{t_2}. The concatenated vector is then processed through a two-layer Multilayer Perceptron (MLP) that consists of a linear layer, layer normalization, a Mish activation function, and a second linear layer:

$$\text{Rel}(t_1, t_2) := \text{LL}\Big(\text{Mish}\Big(\text{LN}\big(\text{LL}(E_{t_1}; E_{t_2})\big)\Big)\Big) \tag{5}$$

We then use the dot product as the similarity function to measure the similarity between two relation representation vectors R_1 and R_2:

$$\mathrm{sim}(R_1, R_2) := R_1 \cdot R_2, \quad R_1, R_2 \in \mathbb{R}^d \tag{6}$$

Following the lightweight computational paradigm of late-interaction, TRIAL efficiently integrates relation representations and their pairwise similarities into the scoring function. This design enables TRIAL to effectively distinguish between phrases with partially shared tokens (e.g., 'Ed Wood' vs. 'Robert Wood') while preserving computational efficiency. In Section 5.1, we conduct an ablation study to examine how token relation scores affect retrieval accuracy. Additionally, in Section 5.2, we provide a qualitative analysis demonstrating how TRIAL leverages token relation scores to assign more accurate query-document relevance scores.

3.4 Token Importance Weight

To address the varying significance of terms within a query, we incorporate token importance weights into the scoring function. These weights allow the model to prioritize key tokens that are critical for determining document relevance. Inspired by the *removal gate* mechanism in ColBERTer (Hofstätter et al., 2022), which identifies contextualized stopwords to prune document tokens and improve latency, our approach differs by focusing on query tokens. Specifically, we design a gating function that learns the significance of each query token, enabling the model to compute more accurate relevance scores.

The importance weight for a token t_i is determined using a gating mechanism, which evaluates the contribution of each token to the overall query-document relevance score. Specifically, the importance weight is computed using a two-layer neural network that employs Mish (Misra, 2019) and ReLU (Nair and Hinton, 2010) activations. The process begins by transforming the token representation E_{t_i} using a learnable weight matrix W_1 and a bias term b_1. The intermediate result is passed through a Mish activation, followed by multiplication with a second weight matrix W_2, and the addition of another bias term b_2. Formally, this is expressed as:

$$\mathrm{Gate}(t_i) := \mathrm{ReLU}\left(\mathrm{Mish}\left(W_1 E_{t_i}^{\top} + b_1\right) W_2 + b_2\right) \tag{7}$$

Here, $W_1 \in \mathbb{R}^{\mathrm{dim}' \times \mathrm{dim}'}$, $W_2, E_{t_i} \in \mathbb{R}^{1 \times \mathrm{dim}'}$, $b_1 \in \mathbb{R}^{1 \times \mathrm{dim}'}$, and $b_2 \in \mathbb{R}^{1 \times 1}$. This gating mechanism highlights the most significant tokens in the query, enabling the model to focus on key components for relevance computation.

We focus exclusively on weighting query tokens for several practical reasons. Firstly, assigning a distinct weight to every query-document token pair is computationally prohibitive and incompatible with standard pre-indexing architectures. Secondly, we found that learning separate weights for document tokens slowed convergence without improving retrieval accuracy as shown in Table 6. This is likely because the MaxSim operation already suppresses low-information document tokens by selecting only the most similar one for each query term, making additional document weights redundant. In contrast, our query-only weighting approach adds negligible computational cost while improving interpretability by learning to emphasize rare and informative query terms, functioning similarly to an IDF-based mechanism.

By leveraging token importance weights, TRIAL effectively differentiates documents based on the prioritized tokens. In Section 5.1, we evaluate the effectiveness of importance weights through an ablation study. Furthermore, in Section 5.3, we analyze the importance weights across different token types based on POS tags and evaluate whether the learned gating function effectively captures meaningful patterns.

3.5 Training

The TRIAL model is trained using contrastive learning, following the prior dense retrieval models (Karpukhin et al., 2020; Khattab and Zaharia, 2020). Specifically, we use the Kullback-Leibler (KL) divergence loss L_{KL} and cross-entropy loss L_{CE}, which together train the model to distinguish relevant documents from irrelevant ones. To enable the model to focus on important tokens, we introduce a regularization loss L_{R} that optimizes the gating function responsible for computing token importance weights.

$$L_{\mathrm{total}} = L_{\mathrm{KL}} + L_{\mathrm{CE}} + L_{\mathrm{R}} \tag{8}$$

KL-Divergence Loss. The KL-divergence loss facilitates knowledge distillation by transferring knowledge from a powerful full-interaction based teacher model (i.e., MiniLM (Wang et al., 2020)) to the TRIAL model.

$$L_{\mathrm{KL}} := \sum_{D \in C} P_{\mathrm{teacher}}(D \mid Q) \log \left(\frac{P_{\mathrm{teacher}}(D \mid Q)}{P_{\mathrm{student}}(D \mid Q)} \right) \tag{9}$$

For the training efficiency, we use the positive D^+ and sampled negatives N instead of the entire corpus C.

$$P(D \mid Q) \approx \text{softmax}_{D \in \{D^+ \cup N\}}(\text{score}(Q, D)) \tag{10}$$

$$= \frac{e^{\text{score}(Q,D)}}{e^{\text{score}(Q,D^+)} + \sum_{D^- \in N} e^{\text{score}(Q,D^-)}} \tag{11}$$

Cross-Entropy Loss. The cross-entropy loss trains the model to distinguish the relevant document D^+ from irrelevant documents N within a mini-batch.

$$L_{\text{CE}} := -\log \left(\frac{e^{\text{score}(Q,D^+)}}{e^{\text{score}(Q,D^+)} + \sum_{D^- \in N} e^{\text{score}(Q,D^-)}} \right) \tag{12}$$

We use in-batch negatives (Henderson et al., 2017; Gillick et al., 2019) for efficiency.

Regularization Loss. To encourage sparse and meaningful token importance weights, we employ L_1 regularization to train the gating function (Hofstätter et al., 2020).

$$L_{\text{R}} := \lambda_q \|W_Q\|_1 \tag{13}$$

Here, $W_Q \in \mathbb{R}^{1 \times |Q|}$ represents the computed token importance weights for the query Q. The regularization coefficients λ_q is set to 10^{-2}. By encouraging sparsity in W_Q, the regularization loss ensures that only the most relevant tokens are assigned significant importance, while less informative tokens are suppressed.

3.6 End-to-End Retrieval

Late-interaction methods leverage approximation algorithms such as PLAID (Santhanam et al., 2022a), MuVERA (Jayaram et al., 2024), and DESSERT (Engels et al., 2024) to retrieve candidate documents efficiently, avoiding the cost of computing exact relevance scores over the full corpus. In this work, we adopt PLAID and modify its *centroid interaction* stage to incorporate token importance, enabling seamless integration with TRIAL. Detailed methodology and empirical analysis are provided in Appendix A.3 and Appendix A.2, respectively.

4 Experiments

4.1 Experimental Setting

We evaluate TRIAL on three benchmarks: MSMARCO (Craswell et al., 2021) for in-domain evaluation, and BEIR (Thakur et al., 2021) and LoTTE (Santhanam et al., 2022b) for out-of-domain evaluation. Following prior work (Lee et al., 2024; Khattab and Zaharia, 2020; Santhanam et al., 2022b), we report nDCG@10 for MSMARCO and BEIR, and Success@5 for LoTTE. TRIAL is compared against a range of sparse and dense retrieval baselines. Comprehensive details on datasets, evaluation metrics, baseline systems, and implementation are provided in Appendix A.1.

4.2 Retrieval Accuracy

We train the TRIAL model on the MSMARCO dataset and evaluate its retrieval accuracy for both in-domain and out-of-domain scenarios. The first column of Table 1 shows nDCG@10 scores on MSMARCO, where TRIAL achieves state-of-the-art performance across various retrieval methods. The subsequent columns of Table 1 present nDCG@10 scores on the BEIR benchmark datasets. TRIAL achieves state-of-the-art performance on four individual datasets and records the highest average accuracy across all BEIR datasets. Notably, TRIAL surpasses other late-interaction dense retrieval models and outperforms the previous state-of-the-art accuracy achieved by the sparse retrieval method SPLADE++. These results demonstrate the robustness and effectiveness of TRIAL in diverse retrieval settings, confirming its superior capabilities in both in-domain and out-of-domain retrieval.

To further investigate out-of-domain retrieval accuracy, we follow prior works (Santhanam et al., 2022b; Lee et al., 2024) and conducted additional evaluations using the LoTTE benchmark. Tables 2 and 3 compare the retrieval accuracy of various models on the LoTTE search and forum benchmarks, respectively. In the LoTTE search benchmark, TRIAL achieves the highest average accuracy across the six datasets, setting a new state-of-the-art. However, in the LoTTE forum benchmark, TRIAL achieves the highest score for only one dataset (i.e., Technology), while ColBERTv2 shows the highest accuracy on the remaining five datasets. We hypothesize that this performance disparity between the search and forum datasets stems from differences in query patterns. Search queries are typically concise and knowledge-focused, whereas forum queries exhibit greater diversity in style and intent (Santhanam et al., 2022b). Consequently, the relation and weight terms learned from MSMARCO's search-oriented queries may not generalize well to forum datasets, which cover long-tail topics and have dis-

Table 1: Retrieval accuracy (nDCG@10) on MSMARCO (in-domain) and BEIR (out-of-domain) datasets. For a fair comparison, base-sized models are used for CELI, GTR, and XTR. Results for competitors are sourced from their respective papers. BM25 is used for candidate retrieval in full-interaction dense retrieval methods. "TRIAL w/o rel. & importance" is equivalent to our trained version of ColBERTv2.

	MS	**BEIR**													
	MARCO	**Arguana**	**Climate-fever**	**DBpedia-entity**	**Fever**	**Fiqa**	**HotpotQA**	**NFCorpus**	**NQ**	**Quora**	**SciDocs**	**SciFact**	**TREC-COVID**	**Touche2022**	**Avg.**
Sparse Retrieval															
BM25	22.8	31.5	21.3	31.3	75.3	23.6	60.3	32.5	32.9	78.9	15.8	66.5	65.6	**36.7**	44.02
SPLADEv2	43.3	47.9	23.5	43.5	78.6	33.6	68.4	33.4	52.1	83.8	15.8	69.3	71.0	27.2	49.85
SPLADE++	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.70
Full-interaction based Dense Retrieval															
MonoBERT	-	29.1	11.2	43.6	74.8	38.9	66.0	36.2	52.3	77.3	15.2	66.9	72.7	31.7	47.38
CELI	-	44.3	15.2	44.9	74.0	**40.1**	**71.4**	**36.8**	52.7	69	16.2	**71.5**	73.6	32.0	49.36
Representation based Dense Retrieval															
DPR	17.7	17.5	14.8	26.3	56.2	11.2	39.1	18.9	47.4	24.8	7.7	31.8	33.2	13.1	26.31
ANCE	38.8	41.5	19.8	28.1	66.9	29.5	45.6	23.7	44.6	85.2	12.2	50.7	65.4	24.0	41.32
GTR	42.0	51.1	**24.1**	34.7	66.0	34.9	53.5	30.8	49.5	**88.1**	14.9	60.0	53.9	20.5	44.77
Late-interaction based Dense Retrieval															
ColBERT	40.1	23.3	18.4	39.2	77.1	31.7	59.3	30.5	52.4	85.4	14.5	67.1	67.7	20.2	45.14
ColBERTv2	-	46.3	17.6	44.6	78.5	35.6	66.7	33.8	56.2	85.2	15.4	69.3	73.8	26.3	49.95
XTR	45.0	40.7	20.7	40.9	73.7	34.7	64.7	34.0	53.0	86.1	14.5	71.0	73.6	34.7	49.41
LITE	45.2	42.4	21.3	43.4	**78.8**	33.6	68.1	35.8	54	83.9	**16.4**	63.3	76.3	30.5	49.83
TRIAL	**46.3**	**53.9**	18.2	**50.7**	78.2	34.6	66.9	33.3	**57.2**	82.4	14.8	65.7	**77.3**	31.0	**51.09**
TRIAL w/o relations	46.1	50.4	18.5	49.8	77.2	34.0	63.3	32.2	57.0	80.9	13.8	63.9	77.3	29.8	49.85
TRIAL w/o rel. & importance	45.6	49.6	18.0	48.8	75.9	32.0	62.2	32.3	51.9	80.6	14.0	63.3	77.3	29.6	48.88

Table 2: Retrieval accuracy (Success@5) of various models on the LoTTE search dataset across different topics.

	Writing	**Recreation**	**Science**	**Technology**	**Lifestyle**	**Pooled**	**Avg.**
BM25	60.3	56.5	32.7	41.8	63.8	48.3	50.56
ColBERT	74.7	68.5	53.6	61.9	80.2	67.3	67.70
ColBERTv2	**80.1**	72.3	56.7	66.1	**84.7**	71.6	71.91
XTR	77.0	69.4	54.9	63.2	82.1	69.0	69.26
SPLADEv2	77.1	69.0	55.4	62.4	82.3	68.9	69.18
TRIAL	79.5	**73.2**	**57.6**	**67.2**	83.6	**71.8**	**72.15**

Table 3: Retrieval accuracy (Success@5) of various models on the LoTTE forum dataset across different topics.

	Writing	**Recreation**	**Science**	**Technology**	**Lifestyle**	**Pooled**	**Avg.**
BM25	64.0	55.4	37.1	39.4	60.6	47.2	50.61
ColBERT	71.0	65.6	41.8	48.5	73.0	58.2	59.68
ColBERTv2	**76.3**	**70.8**	**46.1**	53.6	**76.9**	**63.4**	**64.51**
XTR	73.9	68.7	42.2	51.9	74.4	60.1	61.86
SPLADEv2	73.0	67.1	43.7	50.8	74.0	60.1	61.45
TRIAL	75.5	69.8	45.3	**54.0**	75.8	62.3	63.78

tinct annotation patterns. This finding suggests that adapting the model to account for such variations could further enhance its performance across diverse contexts.

4.3 Retrieval Speed

Table 4 compares the computational complexity and estimated floating-point operations (FLOPs) of TRIAL and the baseline late-interaction model ColBERTv2. TRIAL introduces additional complexity due to the computation of similarity between token relations, leading to an estimated FLOP count that is two orders of magnitude higher than ColBERTv2. However, we analyze the execution time of TRIAL on the MSMARCO dataset and demonstrate its efficiency and practicality for large-scale retrieval tasks.

Table 5 provides a detailed breakdown of the average execution times for each stage of the retrieval process: query encoding, candidate retrieval, and relevance scoring. TRIAL exhibits a modest increase in total query processing time, averaging 603.98 ms compared to 523.34 ms for ColBERTv2. In the candidate retrieval stage, TRIAL averages 519.15 ms compared to 470.73 ms for ColBERTv2, with the additional time attributed to incorporating query importance weights into the centroid interaction mechanism via element-wise multiplication. In the relevance scoring stage, TRIAL requires 48.68 ms compared to 18.90 ms for ColBERTv2, driven by operations such as applying token importance weights and computing token relation similarities. Nonetheless, TRIAL maintains comparable performance in query encoding and candidate generation, demonstrating its efficiency in processing a million-scale corpus.

While TRIAL's complexity is quadratic with respect to the number of document tokens, this computation can be efficiently parallelized using GPUs. The execution time is more influenced by the number of query tokens, as relation scores must be computed sequentially as shown in Equation 4. To assess scalability, we appended dummy tokens to 1,000 sampled queries from the MSMARCO dataset. Figure 1 shows that TRIAL processes queries with up to 200 tokens in under 1200 ms, demonstrating scalability for long inputs. In practice, user queries are typically short, with an average length of 30 tokens in the MSMARCO and BEIR datasets. Under these conditions, TRIAL's execution time closely aligns with ColBERTv2, confirming its efficiency for real-world retrieval scenarios.

Table 4: Comparison of complexity and estimated FLOPs per query for the scoring functions of ColBERTv2 and TRIAL. FLOPs are calculated with n = 32 (query tokens), m = 300 (document tokens), and d = 128 (vector dimension).

	Complexity	**Estimated FLOPs/query**
$f_{\text{ColBERTv2}}$	nmd	1.22×10^6
f_{TRIAL}	$nmd + nm^2d$	3.69×10^8

Table 5: Execution time (ms) for each stage of the retrieval process: query encoding, candidate retrieval, relevance scoring, and total time. The time is measured over 1,000 sampled queries from MSMARCO.

	Query Encoding	**Candidate Retrieval**			**Relevance Scoring**	**Total Time**
		Stage 1	**Stage 2**	**Stage 3**		
ColBERTv2	16.95	3.65	470.73	13.11	18.90	523.34
TRIAL	18.37	4.12	519.15	13.65	48.68	603.98

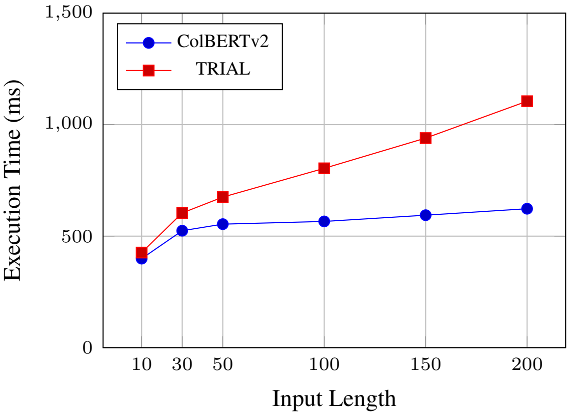

Figure 1: Speed comparison of two models across different input lengths.

5 Analysis

5.1 Ablation Study

We perform an ablation study on both the MSMARCO and BEIR benchmarks to evaluate the individual contributions of TRIAL's components: token relations and importance weights.

In Table 1, the last two rows (highlighted in gray) present the ablation results. Across 13 BEIR datasets, removing token relations reduces the average nDCG@10 from 51.09 to 49.85. Further removing importance weights lowers the score to 48.88. These consistent declines demonstrate the effectiveness of both components across diverse retrieval tasks. Token relations enhance TRIAL's ability to capture semantic dependencies and mitigate noise from multi-token entities. Simultaneously, token importance weights prioritize critical terms, leading to improved relevance scoring. Together, these components enable TRIAL to consistently outperform its ablated variants across a range of retrieval scenarios.

Table 6: Ablation study comparing TRIAL against a variant that also learns document token weights.

	MSMARCO
TRIAL	46.3
TRIAL w/ doc. weight	46.1

5.2 Token Relations

We perform a qualitative analysis using examples from the BEIR benchmark datasets to examine how incorporating token relations enhances relevance scoring. Table 7 presents five representative examples where the inclusion of relation similarity scores improved the ranking of gold documents. Queries and documents are shown in tokenized form, with important phrases that are split into multiple tokens highlighted in red and blue. Their corresponding matches in the documents (i.e., with the maximum similarity) are highlighted in the same colors.

In the first example, the query contains the phrase “movie beyond the sea”, while the gold documents are titled “beyond the sea film". Initially, the document “beyond the sea song” is ranked higher due to partial match of the phrase “beyond the sea”, along with the matching token “sang”. However, incorporating relation similarity allows the model to account for the cohesive phrase "movie beyond the sea" and to increase the ranking of the gold documents with relation similarity scores.

The second and third examples illustrate a similar problem, where a partial match with a noun phrase in the query leads to irrelevant documents being ranked higher. In the second query, the document “localeats” is ranked above the gold document “for against” due to the high similarity score of the token “local”. By incorporating relation similarity scores, TRIAL prioritizes the gold document that matches the full phrase, correctly ranking it

Table 7: Examples from BEIR showing how token relation similarity improves ranking when phrases are split into multiple tokens. Gold documents are ranked higher after adding the relation similarity in the scoring function.

Document Tokens	Relation Sim.	Gold
Query 1: ***[CLS] [QXQ] who sang the songs in the movie beyond the sea [SEP]***		
[CLS] [DXD] beyond the sea film [SEP] beyond the sea is a 2004 american musical drama film based on the life of singer actor bobby dar ##in [SEP]	↑	O
[CLS] [DXD] beyond the sea film [SEP] the soundtrack album features 18 tracks performed by Kevin space ##y [SEP] starring in the lead role ...	↑	O
[CLS] [DXD] beyond the sea song [SEP] robbie Williams sang a version in the closing credits of finding ne ##mo [SEP]	↓	X
Query 2: ***[CLS] [QXQ] are local h and for against both from the united states ? [SEP]***		
[CLS] [DXD] local h [SEP] local h is an American rock band originally formed by guitarist and vocalist scott lucas basists matt garcia ...	↑	O
[CLS] [DXD] for against [SEP] for against is a united states post punk dream pop band from lincoln nebraska [SEP] despite numerous ...	↑	O
[CLS] [DXD] local #ea ##ts [SEP] local ##ea ##ts is a venture backed restaurant information ... operates in the united states and in 50 ...	↓	X
Query 3: ***[CLS] [QXQ] were scott derrick ##son and ed wood of the same nationality ? [SEP]***		
[CLS] [DXD] ed wood [SEP] edward davis wood jr October 10 1924 December 10 1978 was an American filmmaker actor writer ...	↑	O
[CLS] [DXD] scott derrick ##son [SEP] scott derrick ##son born July 16 1966 is an American director screenwriter and producer [SEP] he lives ...	↑	O
[CLS] [DXD] robert wood Williamson [SEP] robert wood Williamson 1856 12 January 1932 was a British solicitor and anthropologist [SEP]	↓	X
Query 4: ***[CLS] [QXQ] what is the meaning of x girl friend [SEP]***		
[CLS] [DXD] ex relationship [SEP] in social relationship ... for example one might refer to a music group s ex guitarist or someone s ex friend ...	↑	O
[CLS] [DXD] x ##sca ##pe album [SEP] blue gangs ##ta was written and recorded in ... the remix titled gangs ##ta no friend of mine featured ...	↓	X
Query 5: ***[CLS] [QXQ] is martin freeman in season 2 of fargo [SEP]***		
[CLS] [DXD] fargo tv series [SEP] the second season in 1979 and starring kirsten dunst Patrick Wilson jesse plemons jean smart and ted danson ...	-	O
[CLS] [DXD] Halloween 4 the return of ... producer paul freeman a friend of akkad with ... under the pseudonym jack martin had written ...	↓	X

higher. The fourth example follows a similar pattern, though it involves only one noun phrase in the query.

The fifth example presents a distinct scenario where the gold document does not explicitly mention the noun phrase from the query. Instead, an irrelevant document containing partial matches with the noun phrase is ranked higher. Specifically, the irrelevant document contains names like 'paul freeman' and 'jack martin', which partially match with "martin freeman". However, by incorporating relation similarity, TRIAL assigns lower scores to the partially matched tokens, enabling the gold document to be correctly ranked higher.

In summary, TRIAL leverages relation similarity scores to treat multi-token entities as cohesive semantic units, reducing noise from fragmented token matches and enhancing relevance computation. As demonstrated in Table 7, TRIAL effectively distinguishes gold documents from irrelevant ones by incorporating relation similarity scores.

5.3 Token Importance

To examine the role of token importance in computing query-document relevance, we analyze the learned importance weights and their average scores across different part-of-speech (POS) tags. Using 1,000 queries sampled from the MSMARCO dataset, we apply the spaCy (Honnibal and Montani, 2017) parser to perform POS tagging and compute average weights and similarity scores for each POS tag.

Figure 2 illustrates the average token weights assigned by TRIAL to various POS tags. Tokens with significant semantic information, such as interjections (INTJ), adjectives (ADJ), nouns (NOUN), and proper nouns (PROPN), are assigned the highest weights, demonstrating TRIAL's ability to prioritize meaningful terms. In contrast, functional tokens such as auxiliary verbs (AUX), determiners (DET), and separators ([SEP]), are assigned lower weights, minimizing their influence on relevance scoring. The low standard deviations for functional tokens indicate TRIAL's consistent handling of less critical terms, while the slightly higher variance for content-rich tokens reflects an adaptive weighting mechanism that depends on contextual relevance.

Figure 3 further highlights the difference in query token scores between TRIAL and ColBERTv2. TRIAL's weighted scoring mechanism amplifies the relevance of semantically important tokens, such as nouns and adjectives, while suppressing less relevant ones. TRIAL assigns scores ranging from 0.27 to 9.74, whereas ColBERTv2 relies solely on token similarity and shows minimal variation in average scores, remaining around 0.8 across all POS tags.

This analysis underscores the difference between TRIAL's importance weighted scoring mechanism and ColBERTv2's unweighted approach. By selectively prioritizing critical tokens and minimizing the influence of less relevant ones, TRIAL achieves more accurate and effective query-document relevance computation across diverse retrieval tasks.

6 Conclusion

In this paper, we introduced TRIAL, a token relations and importance-aware late-interaction model for accurate text retrieval. By explicitly modeling token relations and adaptively weighting token importance, TRIAL advances the state of the art

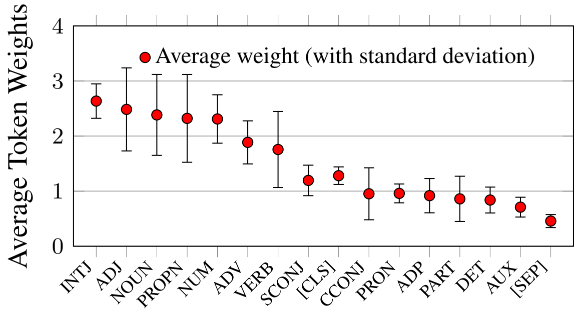

Figure 2: Average token weights for different POS tags in the TRIAL model. Content-rich tokens (e.g., INTJ, ADJ, NOUN) receive higher weights, while functional tokens (e.g., DET, AUX) are assigned lower weights.

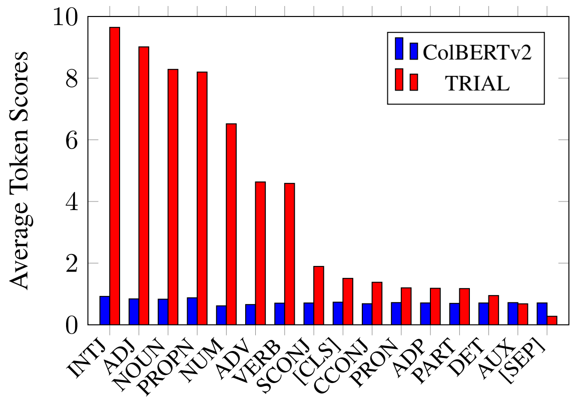

Figure 3: Comparison of scores between ColBERTv2 and TRIAL for different tags, sorted by TRIAL's scores.

in text retrieval, achieving superior accuracy on both in-domain and out-of-domain datasets (i.e., MSMARCO, BEIR, and LoTTE Search). The results show that TRIAL not only enhances semantic understanding and precision but also maintains computational efficiency, making it practical for large-scale retrieval. Ablation studies highlight the individual contributions of token relations and importance weights in boosting retrieval accuracy. Additionally, analysis shows that token relations improve relevance scoring for multi-token phrases, while token importance effectively emphasizes content-rich tokens (e.g., INTJ, ADJ, NOUN).

Limitations

While TRIAL demonstrates strong performance across multiple benchmarks, several limitations remain. First, TRIAL introduces additional computational overhead due to relation modeling. Although our analysis shows it remains practical for real-world use, the quadratic complexity with respect to document tokens may limit scalability for extremely long queries. Second, despite strong results on MSMARCO, BEIR, and LoTTE Search, TRIAL's gains diminish on datasets with long-tail, user-generated content such as LoTTE-Forum. This suggests that token-level relation and importance modeling may be less effective under significant domain shifts. Enhancing robustness to diverse linguistic styles and informal discourse remains an open direction for future work. Third, our evaluation is limited to English-language datasets. The effectiveness of TRIAL on morphologically rich or non-English languages remains unexplored and could be addressed in future research.

Acknowledgements

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00517736, 50%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00509258, Global AI Frontier Lab, 30%) (No. RS-2018-II181398, Development of a Conversational, Self-tuning DBMS, 10%) (No. RS-2024-00454666, Developing a Vector DB for Long-Term Memory Storage of Hyperscale AI Models, 5%), and Basic Science Research Program through the National Research Foundation of Korea Ministry of Education (No. RS-2024-00415602, 5%).

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 15159–15177. Association for Computational Linguistics.

Xiaokai Chu, Jiashu Zhao, Lixin Zou, and Dawei Yin. 2022. H-ernie: A multi-granularity pre-trained language model for web search. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 1478–1489.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Msmarco: Benchmarking ranking models in the large-data regime. In

Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1566–1576.

Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*, pages 1897–1907.

Joshua Engels, Benjamin Coleman, Vihan Lakshman, and Anshumali Shrivastava. 2024. Dessert: An efficient algorithm for vector set search with vector set queries. *Advances in Neural Information Processing Systems*, 36.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. 2022. Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 737–747.

Sebastian Hofstätter, Aldo Lipani, Markus Zlabinger, and Allan Hanbury. 2020. Learning to re-rank with contextualized stopwords. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2057–2060.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Rajesh Jayaram, Laxman Dhulipala, Majid Hadian, Jason Lee, and Vahab Mirrokni. 2024. Muvera: Multi-vector retrieval via fixed dimensional encoding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ziwei Ji, Himanshu Jain, Andreas Veit, Sashank J Reddi, Sadeep Jayasumana, Ankit Singh Rawat, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. 2024. Efficient document ranking with learnable late interactions. *arXiv preprint arXiv:2406.17968*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for splade models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2220–2226.

Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Zhao. 2024. Rethinking the role of token retrieval in multi-vector retrieval. *Advances in Neural Information Processing Systems*, 36.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. Citadel: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11891–11907.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Diganta Misra. 2019. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and 1 others. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1995. Large test collection experiments on an operational, interactive system: Okapi at TREC. *Inf. Process. Manag.*, 31(3):345–360.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and 1 others. 2020. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Crystina Zhang, Minghan Li, and Jimmy Lin. 2024. Celi: Simple yet effective approach to enhance out-of-domain generalization of cross-encoders. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 188–196.

A Appendix

A.1 Experimental Setting

Datasets. To evaluate the effectiveness of TRIAL, we utilize three benchmark datasets: MSMARCO[1] (Craswell et al., 2021), BEIR[2] (Thakur et al., 2021), and LoTTE[3] (Santhanam et al., 2022b). For in-domain retrieval accuracy, we use the MSMARCO dataset, which contains 8.8 million passages from the web and approximately 7,000 real-world queries sourced from Bing's search engine. To assess out-of-domain retrieval performance, we employ the BEIR and LoTTE benchmarks. BEIR includes over 13 datasets spanning nine diverse retrieval tasks (e.g., fact-checking, question answering, and biomedical IR), providing a standardized framework for zero-shot and out-of-distribution evaluations. LoTTE focuses on long-tail topics and natural search queries, comprising 12 domain-specific datasets derived from StackExchange. By covering diverse topics like science, technology, and lifestyle, LoTTE offers datasets for evaluating model performance on underrepresented, real-world retrieval scenarios.

Metrics. We follow existing works (Lee et al., 2024; Khattab and Zaharia, 2020; Santhanam et al., 2022b) to evaluate our system on MSMARCO and BEIR datasets using normalized discounted cumulative gain at rank 10 (nDCG@10), a metric that measures ranking performance by comparing a ranked list to the ideal ranking (Järvelin and Kekäläinen, 2002). Formally, it is defined as:

$$nDCG_N = \frac{DCG_N}{IDCG_N} \tag{14}$$

$$DCG_N = \sum_{i=1}^{N} \frac{G_i}{\log_2(i+1)} \tag{15}$$

$$IDCG_N = \sum_{i=1}^{N} \frac{G_i^{\text{ideal}}}{\log_2(i+1)} \tag{16}$$

Here, G_i is the relevance score at position i, and G_i^{ideal} represents the relevance score in the ideal ranking. nDCG@10 ranges from 0 to 1, with higher values indicating better ranking quality.

For the LoTTE datasets, we use Success@5, which is the standard metric for this benchmark (Santhanam et al., 2022b). This metric evaluates whether a relevant document appears within the top five results. It is defined as:

$$\text{Success@5} = \begin{cases} 1, & \text{if a relevant document is in the top 5,} \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

This metric highlights the importance of retrieving at least one relevant document within the top-ranked results, ensuring user satisfaction.

Competitors. We evaluate TRIAL against a diverse set of neural retrieval systems, covering both sparse and dense retrieval methods.

- *Sparse retrieval:* BM25 (Robertson et al., 1995), SPLADEv2 (Formal et al., 2021a), and SPLADE++ (Lassance and Clinchant, 2022) are widely recognized sparse retrieval methods.

- *Full-interaction dense retrieval:* MonoBERT (Nogueira and Cho, 2019) and CELI (Zhang et al., 2024) are reranking models that achieve high retrieval accuracy by leveraging full interaction between queries and documents.

- *Representation-based dense retrieval:* DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021) and GTR (Ni et al., 2022) are efficient neural retrieval models designed for scalable end-to-end retrieval with fixed representations.

- *Late-interaction dense retrieval:* ColBERT (Khattab and Zaharia, 2020), ColBERTv2 (Santhanam et al., 2022b), XTR (Lee et al., 2024), and LITE (Ji et al., 2024) are representative neural methods that are both scalable and accurate, leveraging token-level representations.

This diverse set of competitors ensures a comprehensive evaluation of TRIAL across various retrieval paradigms.

Implementation. We implemented TRIAL using Python 3, leveraging PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020) to build the neural network architecture. Experiments were conducted on an Ubuntu server with an Intel Xeon Gold 6230 CPU (80 logical threads), 1TB RAM, 1TB storage, and four NVIDIA RTX A6000 GPUs. To enhance training efficiency, we initialized the model parame-

[1] https://microsoft.github.io/msmarco/

[2] https://github.com/beir-cellar/beir

[3] https://github.com/stanford-futuredata/ColBERT/blob/main/LoTTE.md

Table 8: Candidate retrieval accuracy (Recall@256) and end-to-end retrieval accuracy (nDCG@10) for TRIAL on MSMARCO (in-domain) and BEIR (out-of-domain) datasets. PLAID† represents the version that incorporates importance weights during centroid interaction, as detailed in Section A.2. Oracle candidates include annotated gold documents in the candidate set if they are missing in the set.

	MS	**BEIR**													
	MARCO	**Arguana**	**Climate-fever**	**DBpedia-entity**	**Fever**	**Fiqa**	**HotpotQA**	**NFCorpus**	**NQ**	**Quora**	**SciDocs**	**SciFact**	**TREC-COVID**	**Touche2022**	**Avg.**
Candidate Retrieval Accuracy (Recall@256)															
PLAID†	92.8	98.9	49.0	61.6	93.6	70.1	81.9	35.3	93.8	98.8	43.3	93.9	40.8	58.5	72.30
PLAID	81.5	98.2	7.9	22.0	31.0	19.9	40.0	26.5	32.8	81.3	8.2	60.8	6.4	17.9	38.17
End-to-end Retrieval Accuracy (nDCG@10)															
TRIAL w/ Oracle Candidates	46.4	53.9	18.3	51.4	78.4	34.8	67.1	33.4	57.3	82.5	14.8	65.7	81.5	31.1	51.18
TRIAL w/ PLAID†	46.3	53.9	18.2	50.7	78.2	34.6	66.9	33.3	57.2	82.4	14.8	65.7	77.3	31.0	51.09
TRIAL w/ PLAID	43.8	53.8	6.7	32.0	30.0	15.2	41.3	30.5	25.7	71.3	6.1	50.2	53.3	19.3	33.49

ters with the ColBERTv2 checkpoint[4]. Hard negatives were pre-collected using ColBERTv2 for all training queries. The model was trained using the AdamW optimizer (Loshchilov and Hutter, 2019), with a linearly decaying learning rate schedule. The learning rate was initialized to 1×10^{-4} for the pre-trained language model and 1×10^{-5} for other parameters, with a warm-up phase of 2000 steps. Training was conducted for three epochs with a batch size of 32. To balance the regularization and contrastive losses, we applied a regularization loss with $\lambda = 0.01$. The token representation dimension was set to 128, and the maximum token lengths were configured to 40 for queries and 320 for documents. For evaluation, we used the BEIR benchmark scripts[5] to assess retrieval performance using nDCG@10 and Success@5. Additional implementation details, including source code, are available in our open-source repository[6].

A.2 End-to-End Retrieval

We follow ColBERTv2 (Santhanam et al., 2022b) and use a two-stage pipeline. In the first stage, we adapt the PLAID framework to retrieve an initial set of candidates. Then, in the second stage, we use the scoring function of TRIAL to find more accurate exact relevance scores.

We adopt PLAID and modify its *centroid interaction* stage by incorporating token importance weights, enabling seamless integration with the trained TRIAL model for efficient and accurate candidate retrieval. The majority of our candidate retrieval process adheres to the original PLAID algorithm. During the indexing stage, we use K-means clustering to group document tokens into clusters. At retrieval time, centroid interaction approximates relevance scores and extracts the top-k most relevant candidate documents. To begin with, the similarity between centroids and query tokens is computed as a matrix multiplication:

$$S_{c,q} = E_C \cdot E_Q \tag{18}$$

where $E_C \in \mathbb{R}^{|C| \times d}$ represents the centroid embeddings, and $E_Q \in \mathbb{R}^{|Q| \times d}$ represents the query token embeddings. For each query token, the top-k' closest centroids are identified, and approximate relevance scores are calculated for all documents associated with these centroids. Document relevance is estimated using precomputed centroid-to-query similarity scores:

$$\tilde{D} \cdot Q^T = \begin{bmatrix} S_{c,q}[c_1] \\ S_{c,q}[c_2] \\ \vdots \\ S_{c,q}[c_{|d|}] \end{bmatrix} \tag{19}$$

where c_i denotes the centroid index corresponding to the i^{th} document token, and $S_{c,q}[c_i]$ refers to the c_i^{th} row of $S_{c,q}$.

Our modification focuses on the final aggregation step by incorporating token importance weights. While PLAID uses Chamfer similarity to calculate the final approximated relevance scores, we refine the computation by incorporating query token importance. The final relevance score for a document is calculated as:

$$\tilde{s}_{d,q} = \sum_{j=1}^{|Q|} W_j \max_{i=1}^{|d|} \tilde{D}_i \cdot Q_j^T \tag{20}$$

To maintain computational efficiency, we incorporate token importance but omit token relation modeling in the relevance scoring.

[4] https://downloads.cs.stanford.edu/nlp/data/colbert/colbertv2/colbertv2.0.tar.gz

[5] https://github.com/beir-cellar/beir/blob/main/beir/retrieval/evaluation.py

[6] https://github.com/postechdblab/trial

This modification to the centroid interaction mechanism improves the effectiveness of candidate document retrieval when combined with the model trained on the TRIAL scoring function. In Section A.3, we evaluate the impact of this modification on retrieval performance.

A.3 Candidate Retrieval

Candidate retrieval is a critical step in late-interaction dense retrieval systems, as its recall rate directly affects the downstream end-to-end retrieval accuracy. In this section, we evaluate the impact of applying importance weights to the centroid interaction of the PLAID algorithm when using the embedding model trained with TRIAL. Additionally, we analyze how the quality of candidate retrieval influences final retrieval accuracy.

PLAID's candidate retrieval algorithm requires embedding models to generate representation vectors for query and document tokens. The second row of Table 8 shows that directly applying PLAID to the TRIAL model results in a very low recall rate, averaging 38.17 on BEIR datasets. This significantly reduces end-to-end retrieval accuracy to 43.8 nDCG@10 on MSMARCO and 33.49 nDCG@10 on BEIR. To address this limitation, we incorporate token importance weights into the centroid interaction stage of PLAID. As shown in the first row of Table 8, this adjustment substantially improves the recall rate, allowing TRIAL to achieve state-of-the-art accuracy on both MSMARCO and BEIR benchmarks.

To further evaluate whether candidate retrieval constrains end-to-end retrieval accuracy, we use oracle candidates that guarantee the inclusion of gold documents in the candidate set. As shown in Table 8, oracle candidates improve end-to-end retrieval accuracy by only 0.1 nDCG@10, indicating that further improvements in candidate recall would have limited impact. These results emphasize the importance of refining the scoring function to capture semantic relationships more effectively and achieve further improvements in retrieval performance.

A.4 Rationale for Activations, Regularization, and Losses

We provide additional details on our design choices for activation functions, regularization techniques, and loss components, as motivated by empirical ablations.

Activation Function: ReLU vs. Sigmoid. We selected ReLU as the activation function in our model due to its advantages in facilitating faster convergence during training. Unlike Sigmoid, ReLU allows for more efficient gradient flow. In our experiments, using ReLU resulted in a nDCG@10 score of 46.3, compared to 45.0 when using Sigmoid. This improvement underscores ReLU's effectiveness in optimizing the model's performance on ranking tasks.

Regularization: L1 vs. L2. For regularization, we opted for L1 over L2 to promote sparsity in the token weights, which helps in identifying and emphasizing the most salient features while reducing overfitting. L1 regularization encourages many weights to become near zero, leading to a more interpretable and efficient model. Ablation studies showed that L1 regularization achieved a nDCG@10 of 46.3, outperforming L2 regularization's 46.0. This slight but consistent gain highlights the benefits of sparsity in our token weighting scheme.

Ablations on Loss Terms. We conducted ablation study to validate the contribution of KL-divergence loss term. Removing the KL divergence loss led to a performance drop to 45.5 nDCG@10.