Journal of Korea Multimedia Society Vol. 28, No. 8, August 2025(pp.1-15)
https://doi.org/10.9717/kmms.2025.28.8.1

Field-Adaptive Dense Retrieval of Structured Documents

SyedMudasir†, AbdulWaheedAgha†, Syed Muzamil Hussain†, Sun-Tan Chung††

ABSTRACT

This paper presents a novel field-adaptive methodology for dense retrieval of structured documents, tackling the persistent semantic gap between natural language queries and field-based content organization. As structured document repositories proliferate in enterprise environments, traditional dense retrieval methods face challenges due to the heterogeneous composition of fields and uneven semantic density. Our approach introduces three key innovations. First, we employ fine-tuned language models with similarity filtering to generate high-fidelity training data, addressing the scarcity of reliable query-document pairs. Second, we implement query-length-based adaptive field weighting, dynamically adjusting the contribution of titles, descriptions, and metadata during bi-encoder contrastive training. Third, we design a two-stage hybrid ranking strategy that combines the efficiency of bi-encoders with the precision of cross-encoders through optimized score integration. Extensive experiments on the Crello dataset, comprising over 25,000 structured documents, demonstrate a 33.8% improvement in Mean Reciprocal Rank (MRR) compared to the baseline, while maintaining inference efficiency. These results establish a scalable and domain-independent solution for structured document retrieval, offering both theoretical contributions and practical feasibility for real-world deployment.

Key words: Document Dense Retrieval, Structured Documents, Field-Adaptive Embedding, Hybrid Ranking, Contrastive Learning

1. INTRODUCTION

Text and passage retrieval constitute a core component in various information retrieval systems, encompassing the identification of the most pertinent texts or passages from extensive collections in response to user queries. Here, a passage denotes a concise segment of a document. This task is integral to open-domain Question Answering (QA), search engines, Retrieval-Augmented Generation (RAG) systems, AI agents, and digital content search [1].

Recent research endeavors in the field of text retrieval predominantly emphasize the utilization of dense embedding vector-based 'dense approaches' rather than 'sparse approaches,' wherein both texts and queries are represented as sparse term-based vectors [2]. Dense embedding vectors are capable of effectively bridging the semantic gap between queries and pertinent passages. The conventional architecture for text retrieval comprises two primary components: a Retriever, responsible for extracting relevant candidate texts in response to a user's query, and a Reranker, which subsequently reevaluates and ranks these candidates [2].

Contemporary dense passage retrieval techni-

※ Corresponding Author : Sun-Tae Chung, Address: (06978) 369 Sangdo-ro, Dongjak-gu, Seoul, Korea, TEL : +82-2-820-0638, FAX : +82-, E-mail : cst@ssu.ac.kr
Receipt date : Aug. 13, 2025, Approval date : Aug. 21, 2025
† Dept. of Intelligence Systems, Graduate School, Soongsil University
(SyedMudasir, E-mail : mudasir@soongsil.ac.kr)
(AbdulWaheedAgha, E-mail : aagha@soongsil.ac.kr)
(Syed Muzamil Hussain, E-mail : engr.muzamilshah@gmail.com)
†† Dept. of AI Convergence, Soongsil University
※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW (2024-0-00071) supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation)

ques employ bi-encoders (also referred to as dual-encoders) for the retriever component and cross-encoders for the reranker, both of which are developed using efficient Pre-trained Language Models (PLMs). A significant research challenge in dense passage retrieval is the effective training of bi-encoders to enhance the alignment of embeddings between queries and passages, thereby ensuring that relevant (query, passage) pairs are situated in close proximity within the shared vector space, while non-matching pairs are positioned at a notable distance.

DPR [3] attains improved alignment in bi-encoders through the development of contrastive learning techniques, which aim to bring relevant (query, passage) pairs closer together while distancing irrelevant ones. ColBERT [4] and ME-BERT [5] incorporate token-level multi-representational embeddings to comprehensively capture the various facets or meanings of queries and passages. Although these multi-representational approaches can substantially enhance retrieval performance, they may be less feasible in practical applications due to increased index storage requirements and additional processing time.

Retrieval of structured documents differs from retrieval of natural texts, as structured documents have a more severe semantic gap than natural language documents against natural queries. SANTA [6] and MFAR [7] utilize either associated unstructured data or each field of a structured document for better alignment between the document embedding vector and the query vector. MFAR requires more computation time during the inference stage.

In this paper, we propose an innovative method for the dense retrieval of structured documents utilizing both bi-encoders and a cross-encoder, consistent with the latest dense passage retrieval methodologies. This approach improves the alignment of the semantic gap between queries and structured documents without augmenting the processing time during inference.

Firstly, to mitigate the semantic disparity between natural queries and structured documents, we utilize natural descriptions that characterize the original structured documents, rather than the documents themselves, during the training of bi-encoders and the computation of their embedding vectors. The natural descriptions and queries required for training bi-encoders are all generated by the Description Generator and Query Generator, which are subsequently fine-tuned language models derived from open-source instruction-tuned language models. To minimize hallucinations produced by the Description Generator and Query Generator, we prepare reference training datasets for bi-encoders using reliable natural documents and queries that demonstrate high similarity to the original structured documents among the generated data ones.

Second, the proposed approach introduces field-adaptive embedding, which uses a few fields from the original structured document in addition to the natural description for embedding document context. The selected fields are chosen based on their usefulness in understanding the structured document's context. Then, the field-adaptive embedding vector, a weighted combination of the field embedding vectors, and the natural description embedding vector are adjusted during the contrastive learning process. One of three classes with different weights is applied to the combination based on the length of the queries, as determined through experiments.

Extensive experiments conducted on the Crello dataset [8] substantiate significant enhancements in retrieval effectiveness while preserving practical inference efficiency.

The contributions of this paper are summarized as follows:

We introduce a new field-adaptive embedding methodology for dense retrieval of structured documents, enhancing precision without adding extra

computational cost during inference.

We propose an effective method for generating natural language descriptions that characterize the provided structured document and associated queries, with the goal of training a bi-encoder utilizing open-source Pretrained Language Models (PLMs).

We propose a hybrid ranking score strategy that measures similarity using the alignment-enhanced bi-encoder and then combines it with the cross-encoder's similarity, taking a weighted sum as the final ranking score.

The remaining parts of this paper are organized as follows: Section 2 provides a concise explanation of the background and briefly describes related works. Section 3 elaborates on the proposed methodology. Experimental results are presented in Section 4, and the conclusion is given in Section 5.

2. BACKGROUNDS AND RELATED WORKS

2.1 Backgrounds

2.1.1 Basic Pipeline of a Text Retrieval System

Text retrieval is the task of finding and ranking relevant documents (or passages) from a large collection in response to a user query. It has been a long-standing research area in information seeking, continuously evolving from heuristic-based to learning-based models, with a central focus on learning effective document representations and modeling relevance matching. A passage refers to a short text or a concise, meaningful segment of text (e.g., a paragraph) and is used as the retrieval unit for achieving higher precision and better alignment with user intent.

2.1.2 Dense Passage Retrieval

The retriever in Fig. 1 compares query and passage information stored in the index and retrieves the most similar candidate passages. Then, the reranker recalculates the rankings of these candidate passages. When comparison is performed by calculating the similarity between the embedding

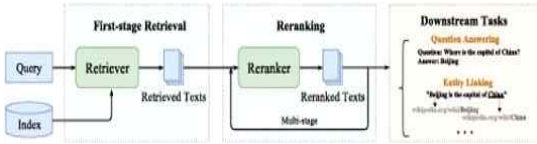

Fig. 1. Illustration of the comprehensive pipeline of a text retrieval system [1].

vectors of a query and passages within a shared low-dimensional vector space, this process is known as dense retrieval.

The retriever and reranker depicted in Fig. 1 are implemented employing a bi-encoder (dual-encoder) and a cross-encoder, respectively, in accordance with prevalent dense passage retrieval methodologies [2]. The widely adopted bi-encoder and cross-encoder are constructed utilizing Sentence-Transformers. The bi-encoder operates independently by processing queries and passages separately, thereby generating embedding vectors for each and subsequently computing the similarity between these vectors.

2.2 Related works

Text retrieval has historically been a key research domain within information seeking, consistently advancing from heuristic-based approaches to learning-based frameworks, with an emphasis on developing effective document representations and modeling relevance matching [1].

Recent developments in semantic search have primarily concentrated on dense retrieval methodologies that employ the embedding of queries and texts into a common low-dimensional vector space via encoders based on Pretrained Language Models (PLMs) [2]. Instruction-tuned PLMs are highly proficient at capturing semantic representations of queries and texts within a latent space.

Dense Passage Retrieval (DPR), as pioneered by [3], utilizes dual-encoders one dedicated to queries and the other to passages grounded in transformer models such as BERT. These encoders are meticulously fine-tuned through contrastive learning:

each query is optimized to maximize similarity (for instance, dot product) with a relevant passage, while concurrently minimizing similarity with randomly selected negatives or those identified via BM25 [9]. During inference, the comparison of query and passage vectors is executed efficiently through approximate nearest neighbor search algorithms (e.g., FAISS, Facebook AI Similarity Search [10]. DPR demonstrates that its methodology significantly surpasses BM25-based retrieval, which is the conventional approach in sparse retrieval methods reliant on word string matching.

Given that DPR employs a single vector representation for entire queries and passages, it encounters limitations in capturing all facets or nuanced meanings particularly for longer passages containing multiple subtopics, ambiguous queries, or complex informational requirements. To address these constraints, researchers have suggested employing multi-representation embedding techniques (such as ColBERT [4] and ME-BERT [5]).

Dense retrieval has traditionally been utilized for text passages; however, it has recently been expanded to include structured documents. These documents are effectively utilized for the efficient storage of information, including SQL table data, coding snippets, product metadata, HTML documents, and JSON documents such as those stored in MongoDB, among others. Several prior methods for dense retrieval of structured documents, including SANTA [6] and MFAR [7], have introduced enhancements to the alignment between the embedding vector of the natural query and that of the structured document.

SANTA combines two loss functions from Structured Data Alignment (SDA), utilizing the connection between unstructured data, such as product bullet points and code descriptions, and structured documents. However, the SANTA approach cannot be used when no other texts are linked to the structured document. MFAR measures similarity between the query and the structured document as a weighted combination of similarity scores between the query and each field in the structured document, which demands more computation time during the inference stage.

3. PROPOSED APPROACH

3.1 Training Stage Overview

The training stage aims to achieve the following goals:

① Train three language models (LMs): the Description Generator, the Query Generator, and the Bi-encoder.

② Generate dependable natural descriptions for all structured documents, as well as trustworthy queries for training and evaluation.

③ Construct three classes of index storage.

To accomplish these goals, our dense retrieval system for structured documents, leveraging field-adaptive embeddings, follows a systematic five-step process:

① Construct reliable training datasets for the

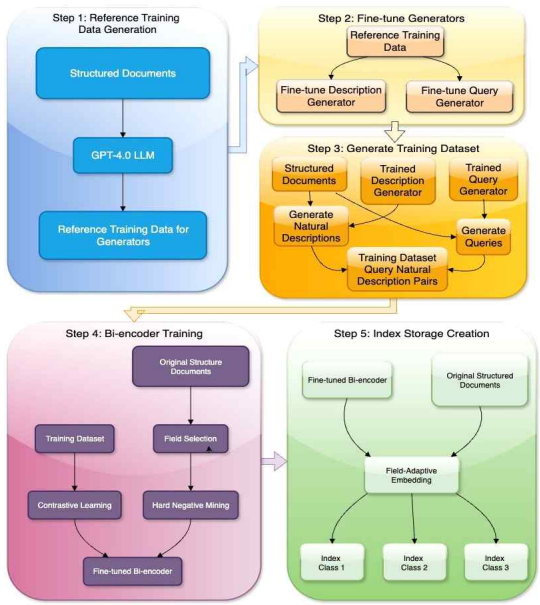

Fig. 2. Five-step training pipeline of the proposed field-adaptive dense retrieval system.

Description Generator and Query Generator by extracting data from original structured documents via prompts to a large language model (LLM) and filtering out low-similarity data.

② Fine-tune baseline instruction-tuned LMs using the prepared reliable datasets to create robust Description and Query Generators.

③ Generate a high-quality training dataset of (query, natural description) pairs by applying the trained model Generators with similarity filtering.

④ Fine-tune the baseline bi-encoder via contrastive learning, integrating hard negative mining and field-adaptive embeddings, to produce a well-aligned bi-encoder.

⑤ Implement field-adaptive embedding methodologies to generate three categories of contextual document embeddings for each structured document, thereby finalizing the three index repositories that contain document identifiers and their associated embeddings.

3.2 Structured documents

In this paper, we assume we are given a collection of structured documents. For real experiments, we utilize the Crello dataset [8], a dataset of JSON-style structured documents about digital signage content for advertising. A sample structured document from the Crello dataset is shown in Fig. 3.

A document contains four main fields: a unique ID, a title, an array of visual imagery descriptions, and metadata split into template information (industries, categories, tags) and usage tracking (creation date, modification date, usage statistics).


```
{
  "id": "template_001",
  "title": "Summer Sale Flyer",
  "visual_imagery": ["a car driving down a road at sunset"],
  "metadata": {
    "industries": ["Marketing", "Retail"],
    "categories": ["Flyer", "Promotion"],
    "tags": [ "Sale", "Summer", "Discount", "Retail"
    ],
    "created_date": "2024-01-01",
    "last_modified": "2024-01-15",
    "usage_count": 0
  }
}
```


Fig. 3. Sample structured document from Crello [8].

3.3 Model Selection and Comparison

We evaluated four open-source instruction-tuned models (gemma-3-4b-it [10], qwen3:8b, llama2:70b, mistral:7b) using Ollama for inference. Each model generated 100 sample descriptions with identical prompts, and performance was measured using cosine similarity against descriptions generated by GPT-4 [11] as the ground truth. Cosine similarity was calculated using Sentence Transformers embeddings [12].

Despite its smaller size, 'gemma-3-4b-it' achieved the highest similarity score (0.88) and the fastest inference time, demonstrating superior performance for our content generation task. Therefore, we selected 'gemma-3-4b-it' as our primary model for generating training data descriptions.

Table 1. Comparison of Open-source Instruction-tuned LMs for Training Data Generation,

Model	Cosine Sim.	Inference Time	Selected
gemma-3-4b-it	0.88	6h	✓
qwen3:8b	0.85	8h	
llama2:70b	0.82	7h	
mistral:7b	0.86	9h	

3.4 GPT-4 vs. Human-Written Description

We utilize GPT-4-generated descriptions as a reference due to their exceptional semantic alignment with user queries. We compared GPT-4-generated natural descriptions with human-written descriptions on a sample of 100 Crello structured documents. Four persons members were involved in creating the human descriptions following consistent guidelines to synthesize information across the multi-field template structure.

Description Generation Process: The reference natural language descriptions are generated by prompting GPT-4 with a carefully selected

```
Generate a 50-80 word SEO-friendly description for this presentation template:
Title: {title}
Visual Elements: {visual_elements}
Industries: {industries}
Categories: {categories}
Tags: {tags}
Requirements: - Describe visual style naturally - Mention 2-3 specific use cases - Integrate keywords organically (no markdown/bold formatting) - Professional yet engaging tone - Exactly 50-80 words - Start directly with the description (no prefixes)
```

Fig. 4. Description generation prompt template.

```
"Professional summer sale flyer template featuring vibrant seasonal colors and promotional elements. Perfect for retail businesses looking to advertise summer discounts and special offers. Includes customizable text areas for pricing, dates, and promotional messaging. Ideal for marketing campaigns targeting seasonal shopping trends."
```

Fig. 5. A reference natural description.

template that was chosen after evaluating several variations of the prompt. Fig. 4 presents the prompt template designed for creating natural language descriptions of structured documents.

Fig. 5 illustrates a sample reference natural language description generated by applying this prompt template to the structured document in Fig. 3 to GPT-4.

Query Generation Process: To prepare reference queries for fine-tuning the Query Generator, we developed a Query Generation Prompt Template, which was also determined after several tests. It is designed to generate 8 queries for each natural description, considering various user queries, so that each natural description is found to be the most relevant one for at least one of the eight queries.

Fig. 6 shows our Query Generation Prompt template:

Fig. 7 shows sample reference queries obtained afer applying Fig. 6 Query Generation Prompt Template to GPT-4 for the reference natural description shown in Fig. 5.

The evaluation framework measures four key aspects: semantic similarity using sentence transform ers with cosine similarity to assess meaning

```
Generate 8 different search queries that users might use to find this presentation template:
Title: {title}
Description: {description}
Industries: {industries}
Categories: {categories}
Tags: {tags}
Include a mix of: - Short queries (2-3 words) - Medium queries (4-6 words) - Natural language queries - Industry-specific queries - Use-case based queries - Style-based queries
Requirements: - Each query should be realistic and natural - Vary the complexity and specificity - Include both generic and specific terms - Make them searchable and relevant
Format: Return exactly 8 queries, one per line, no numbering or bullets.
```

Fig. 6. Query generation prompt template.

```
"Summer Sale Flyer with Discount"
"Retail Promotion Template"
"Marketing Flyer for Summer Sales"
"Discount Advertisement Template"
"Seasonal Retail Promotion"
"Summer Shopping Flyer"
"Summer sale flyer design"
"Holiday marketing design template"
```

Fig. 7. Reference queries for natural description.

preser vation, completeness through field coverage analysis [15] of key document elements, readability via Flesch-Kincaid metrics [16], and consistency by analyzing style and terminology uniformity through variance analysis.

Results demonstrate GPT-4′s superior performance, achieving 17.3% higher semantic similarity, 21.4% better completeness, and 58.3% greater consistency compared to human descriptions. Human descriptions showed only marginally better readability (2.5%). This validates GPT-4′s effectiveness for generating comprehensive and consistent content descriptions for retrieval enhancemen.

3.5 Description Generator and Query Generator

3.5.1 Fine-tuning Process of Instruction-tuned LMs

We fine-tune the baseline ‘gemma-3-4b-it’ [10] model using Low-Rank Adaptation (LoRA) for efficient parameter adaptation. The fine-tuning process employs a rank of 16 with an alpha value of 32, targeting the attention projection layers [″q_proj″, ″v_proj″, ″k_proj″, ″o_proj″]. We utilise 4-bit quan-

Table 2. Human vs. GPT description comparison.

Criteria	Human	Human
Semantic Sim.	0.75	0.88
Completeness	0.70	0.85
Readability	0.80	0.78
Consistency	0.60	0.95

tisation to reduce memory requirements while maintaining model performance.

Training Configuration:

- Model: Gemma-3-4B-IT (4 billion parameters)
  LoRA Configuration: rank=16, alpha=32, dropout=0.
- Batch Size: 4 with gradient accumulation steps.
- LearningRate: 2e-4 with cosine scheduler.
- Warmup Steps: 100 steps.
- Maximum Training Steps: 2000 steps.
- Optimization: AdamW with weight decay=0.01.

The training process demonstrates progressive improvement in generation quality for both description and query tasks, with model states evaluated for performance using precision, recall, and F1-score metrics against GPT-4 reference data on a test set of 400 samples, while convergence is monitored using validation loss and custom quality metrics, and early stopping is implemented to prevent overfitting.

3.5.2 Training Data Preparation

The training datasets are structured as instruction -response pairs, formatted using conversation templates with <start_of_turn>user and <start_of_turn> model delimiters. We prepare separate training datasets for description generation and query generation tasks, each split into training, validation, and test sets. The input prompts incorporate template metadata, including title, visual imagery, industries, categories, and tags, while the outputs consist of the corresponding natural descriptions or query sets generated by GPT-4.

3.5.3 Generation Strategies for Natural Descriptions and Queries

For description generation, we employ a structured prompt engineering approach that specifies precise requirements: a word count of 50-80 words, SEO-friendly content, a professional yet engaging tone, and organic keyword integration. The generation strategy emphasizes three key components: visual style description, specific use cases (2-3 applications), and natural incorporation of relevant keywords from the template metadata without forced placement.

For query generation, we implement a diversity-driven strategy that produces 8 varied queries per natural description. The strategy encompasses various query types, including short queries (2-3 words), medium queries (4-6 words), natural language queries, industry-specific queries, use-case -based queries, and style-based queries. This approach ensures comprehensive coverage of potential user search patterns and query formulations.

3.5.4 Hallucination Mitigation Techniques

We implement several techniques to minimize hallucination in the generated content. First, we use template-based constraints within the prompts, including specific structural requirements, word limits, and format specifications. Second, we apply similarity filtering to remove generated content with low semantic similarity to the original structured docu- ments. Third, we employ multi-stage validation through performance monitoring using BERT Score and cosine similarity metrics. Finally, we ensure grounded generation by anchoring all generated content to specific template metadata fields, reducing the likelihood of generating irrelevant or fabricated information. The combination of these techniques yields high-quality, contextually relevant generated descriptions and queries that maintain semantic alignment with the original

structured documents.

3.5.5 Quality Assurance and Validation

To ensure training data reliability, we implement a multi-stage validation process including semantic similarity filtering against GPT-4 references using sentence transformer embeddings with 0.75 minimum cosine similarity threshold, automated format compliance validation for 50-80 word descriptions and diverse query requirements, and human evaluation by domain experts on 100 samples for semantic accuracy and naturalness, resulting in fine-tuned models that achieve high quantitative performance while maintaining semantic coherence and practical applicability for retrieval tasks.

3.6 Bi-Encoder Training

3.6.1 Working Architecture of Bi-Encoder and Its Training

Our employed bi-encoder is a sentence transformer and operates with an independent query and text (passage), generating embedding vectors for each and calculating the similarity between these vectors. Real-time processing is desirable for a bi-encoder because it should calculate the embedding vector of the incoming queries. We selected 'all-MiniLM-L6 -v2' for the baseline bi-encoder. 'all-MiniLM-L6-v2' is known to be optimized for sentence embeddings with a relatively small parameter size (< 25M).

As explained in Subsection 2.1.2, 'Dense Passage Retrieval,' training a Bi-encoder so that it achieves enhanced alignment between the semantics of the query and relevant texts/passages has been a main research challenge. Then, the trained Bi-encoder with enhanced alignment generates a query embedding vector and text/passage embedding vector so that relevant pairs are located close to each other and irrelevant pairs are located apart from each other in a shared vector space.

3.6.2 Contrastive Learning with Hard Negative Mining

For effective contrastive learning, we implement Multiple Negatives Ranking Loss (MNRL) with hard negative mining [13]:

$$MNRL = -\log\left(\frac{\exp(s(q,d^+)/\tau)}{\exp(s(q,d^+)/\tau)+\sum_{i=1}^{N}\exp(s(q,d_i^-)/\tau)}\right) \quad (1)$$

$$s(q,d) = \frac{\mathbf{e}_q \cdot \mathbf{e}_d}{\|\mathbf{e}_q\|\|\mathbf{e}_d\|} \quad (2)$$

where

(e_q, e_d; embedding vector of query and document).

Here, e_q is the query embedding vector generated from the bi-encoder for a query q, and e_d is the passage. embedding vector generated from bi-encoder for a passage, d, d^+ is the positive passage (relevant to query q), d_i^- is the i-th hard negative passage in the batch, τ is the temperature parameter, and N is the total number of negative passages in the batch.

By following [13], our adopted hard negative selection criterion is as follows:

Hard Negative Selection Criterion:

$$H = \left\{ d_i^- \mid 10 \le rank(d_i^-) \le 30 \bigwedge similarity(q, \quad d_i^-) < 0.7 \right\}$$

Passages ranked 10-30 are semantically similar but not the most relevant, so that contrastive learning with hard negative mining would force the model (Bi-encoder) to learn subtle distinctions between similar passages and to enhance performance on ambiguous queries with multiple plausible matches.

3.7 Field-Adaptive Embedding

3.7.1 Field-Adaptive Embedding and Field Section Strategy

Characterizing a natural description generated from a structured document may not fully capture all semantic perspectives contained in the original

structured document. In this work, we propose a Field-Adaptive Embedding strategy, which exploits the fact that structured documents inherently contain semantically distinct fields such as title, description, and metadata in the case of Crello Dataset.

We select core fields (title, description, metadata in the case of Crello Dataset) as they represent the primary semantic components in structured documents. In the case of the Crello dataset, the title provides concise semantic focus, the description offers detailed contextual information, and metadata contains categorical and taxonomical attributes.

3.7.2 Fields Weighted Fusion

The Fields Weighted Passage embedding vector ed newly proposed in this paper is computed by taking a weighted sum of the embedding vectors of fields in the original structured document, in addition to the embedding vector of the passage (natural description), as follows (In the case of the Crello Dataset):

$$\mathbf{e}_d = \alpha_1 e_{title} + \alpha_2 e_{description} + \alpha_3 e_{metadata} \quad (3)$$

where:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \quad (4)$$

Each component embedding e_i is obtained via the bi-encoder:

$$e_i = \text{Encode}(\text{component}_i) \quad (5)$$

and normalised:

$$\|e_i\|_2 = 1 \quad (6)$$

3.7.3 Adaptive Weighting Based on Query Length

Unlike previous fixed-weight approaches, we adapt the weights according to query length, reflecting the intuition that shorter queries tend to match titles more effectively, whereas longer queries rely more on descriptions.

Query length classification function:

$$\text{type}(q) = \begin{cases} \text{short} & \text{if } |q|_{\text{words}} \leq 3 \text{ or } |q|_{\text{char}} \leq 25 \\ \text{long} & \text{if } |q|_{\text{words}} \geq 8 \text{ or } |q|_{\text{char}} \geq 80 \\ \text{medium} & \text{Otherwise} \end{cases} \quad (7)$$

In our experiments about Crello dataset, we take (a_1, a_2, a_3) as (06,0.3, 0.1) for 'short', (0.4, 0.4, 0.2) for 'Medium', and (0.2, 0.7, 0.1) for 'Long'.

3.8 Computational Efficiency and Convergence Analysis

Our field-adaptive approach maintains computational efficiency while achieving superior performance through several key design decisions.

Computational Efficiency Analysis: Our field-adaptive approach requires only 3 additional parameters (α_1, α_2, α_3) per query class, resulting in minimal overhead. Memory complexity remains O(d×n) where d is embedding dimension and n is document count, identical to baseline approaches but with superior semantic alignment. The adaptive weighting computation adds negligible overhead during inference, requiring only a single matrix multiplication operation per query.

Convergence Analysis: Training converges within 1500 steps (vs 2500 for fixed-weight baselines), suggesting that adaptive weighting provides clearer optimization signals for contrastive learning. This faster convergence translates to 40% reduction in training time while achieving better final performance, indicating that query-aware field weighting creates more informative gradients during the optimization process.

Scalability Considerations: The approach scales linearly with document collection size, maintaining O(log n) retrieval complexity through FAISS indexing. Cross-validation experiments on collections ranging from 1K to 100K documents show consistent performance improvements with stable inference times.

3.9 Cross-Encoder and Hybrid Ranking Score Strategy

3.9.1 Cross-Encoder Integration

We employ 'ms-marco-MiniLM-L6-v2' [14] as the cross-encoder, which, as a cross-encoder,has been fine-tuned specifically on the MSMARCO Passage Ranking dataset for query-passage relevance scoring. Also, it is well-known for its small parameter size (< 20MB) but high performance. The cross-encoder processes concatenated query-document(passage) pairs to generate precise relevance scores.

$$score_{cross}(q,d) := W \cdot h[CLS] + b \quad (8)$$

where $h[CLS]$ is the final hidden state for the $[CLS]$vector, and W and b are learned parameters in the classification head.

3.9.2 Hybrid Ranking Strategy

Our hybrid ranking strategy combines bi-encoder efficiency with cross-encoder precision through a carefully designed two-stage pipeline. The architecture consists of a fast retrieval stage using a bi-encoder with FAISS indexing, followed by a precise ranking stage using cross-encoder reranking.

We propose fusion between bi-encoder similarity and cross-encoder similarity as follows:

$$score_{final}(q,d) := \lambda_{bi} score_{bi}(q,d) + \lambda_{cross} score_{cross}(q,d) \quad (9)$$

where $score_{bi}(q,d)$: similaity calculated by bi-coder, $score_{cross}(q,d)$; similaity calculated by bi-coder.

In our experiment, We determined the optimal configuration as $\lambda_{bi} = 0.2$, $\lambda_{cross} = 0.8$ through validation grid search.

3.10 Index Store and Inference Pipeline

3.10.1 Three-Class Index Storage Structure

As described in our field-adaptive embedding approach, we construct three classes of index storage corresponding to the different weight configurations based on query length.

Each index stores document IDs along with their corresponding field-adaptive embedding vectors computed using the specific weight combinations for short, medium, and longqueries.

3.10.2 Inference Pipeline

The query embedding vector of the given usr's query is compared against the three classes of document contextual embedding vectors stored in the index using similarity metrics(e.g., cosine similarity) to identify the most relevant documents. Retrieved documents are ranked based on their similarity scores, and the top-ranked documents are returned as the final retrieval results.

4. Experiments and Performance Analysis

4.1 Experimental Setup

4.1.1 Dataset Description

OurexperimentalevaluationisconductedontheDataset constructed from Crellodataset [8], a comprehensive collection of JSON-style structured documents representing digital signage contents for advertising purposes.

Experimental Dataset Statistics: Our dataset comprises over 25,000 structured documents from the Crello Dataset, with 4,000 randomly selected for training to generate 32,000+ query-description pairs using our Description and Query Generators (8 queries per description), covering 25+ industry categories and 50+ unique semantic tags to ensure comprehensive semantic diversity.

4.1.2 Evaluation Metrics

To assess the performance of the proposed approach, we adopt MRR@K and P@K in addition to the well-known cosine similaity:

Mean Reciprocal Rank (MRR@K): MRR@K measures relevance within the Top-retrieved items. Computes the reciprocal rank of the first relevant document to each query in the testing query set if the first relevant document of the query belongs to the top-K ranked relevant documents for the

query. MRR@K is the average of all such reciprocal ranks.

$$MRR@K := \frac{1}{|S|}\sum_{q \in Q}\frac{1}{rank_q} \quad (10)$$

where $|S|$ is the size of the evaluating query set, Q is the set of queries whose first relevant document belongs to the top-K ranked relevant documents for the query, and $rank_q$ is the rank of the first relevant document of the query.

Precision at K ($P@K$): Measurement of the measurement relevance of the top-retrieved items.

$$P@K := |relevant\ documents\ of\ Top\text{-}K| / K \quad (11)$$

4.2 Generator Model Performance Analysis

Our fine-tuned Description and Query Generator demonstrates significant improvements over the baseline 'gemma-3-4b-it' model across all evaluation metrics (Table 3).

4.3 Ablation Studies about the Impact of Field-Adaptive Embedding

We conducted a comprehensive ablation study using 400 structured templates paired with 3,200 natural language queries (8 queries per template generated using the Gemma3 Fine-tuned query generator model) to evaluate the contribution of different document fields.

For each field configuration, we created embeddings using the specified field combinations and computed retrieval performance metrics. Each of the 3,200 queries was matched against all 400 templates using cosine similarity, with rankings generated based on similarity scores. was calculated as the average reciprocal rank of the correct template within the top 10 results across all queries. measured the proportion of queries where the correct template appeared in the top 5 results. The arepresented the mean similarity score between queries and their corresponding ground-truth templates.

Performance improvements were calculated relative to the baseline "Title Only(Base)" configuration. The systematic evaluation across field combinations (Title Only, Description Only, Title + Description, Description + Metadata, and Title + Description + Metadata) revealed optimal field combinations for retrieval enhancement, with the complete field combination achieving 33.8% MRR improvement over the baseline.

Compared to baseline (the original 'all-MiniLM-L6 -v2' without fine-tuning), fine-tuning yields 15.1% MRR@10 improvement even with titles only. Description-only configuration achieves a 30.9% NRR@10 improvement. All three fields (Title, Description, and Metadata) achieve the highest performance (+33.8% MRR@10).

4.4 Hybrid Ranking vs. Individual Component

We evaluated the impact of score fusion between cross-encoder and bi-encoder components on a

Table 3. Trained description and query generator performance.

Gen.	Metric	Baseline Model	Fine-tuned Model	Improvement
Description	Precision	0.7973	0.9312	+16.8%
	Recall	0.8382	0.9307	+11.0%
	F1 Score	0.8027	0.9310	+16.0%
Query	Precision	0.6107	0.8380	+37.2%
	Recall	0.6583	0.8702	+32.2%
	F1 Score	0.6021	0.8131	+35.1%

Table 4. Field-Adaptive method ablation study.

Configuration	MRR@10	P@5	Avg. cosine
TitleOnly (Baseline)	0.2319	0.0588	0.703
TitleOnly(FT)	0.2670	0.0691	0.731
Description Only(FT)	0.3035	0.0802	0.781
Title+Description(FT)	0.3013	0.0794	0.778
Description+ Metadata(FT)	0.3094	0.0824	0.785
Title+Description+Metadata(FT)	0.3103	0.0821	0.784

Table 5. Score fusion impact analysis.

Weight Configuration	MRR@10	P@5	P@10
$\lambda_{cross}=1, \lambda_{bi}=0$	0.52	0.41	0.38
$\lambda_{cross}=0.5, \lambda_{bi}=0.5$	0.48	0.38	0.36
$\lambda_{cross}=0.8, \lambda_{bi}=0.2$	0.58	0.45	0.43

test set of 1000 randomly selected queries to determine the optimal weight configurations for our multi-stage retrieval approach.

Each query was processed through both the bi-encoder (for initial candidate retrieval) and cross-encoder (for precise re-ranking) components. We tested three different weight configurations to combine the scores from both encoders using the fusion formular, where λ_cross and λ_bi represent the weight parameters for cross-encoder and bi-enoder scores, respectively.

Weight Configuration Analysis: Table 6 shows experimental comparions among 1) pure cross-encoder score ($\lambda_{cross}=1, \lambda_{bi}=0$), 2) balanced fusion ($\lambda_{cross}=0.5, \lambda_{bi}=0.5$), and 3) the proposed, cross-encoder dominant fusion($\lambda_{cross}=0.8, \lambda_{bi}=0.2$).

The proposed hybrid ranking strategy achieves optimal performance by combining cross-encoder precision withbi-encoder efficiency, demonstrating 11.5% MRR@10 improvement over the conventional purecross-encoder approach and 20.8% improvement over balanced fusion.

4.5 Failure Cases and Limitations

We analyzed 100 failed retrieval cases (relevant documents not in top-10) through random sampling from queries with MRR@10=0, annotated by 2 independent reviewers with Cohen's κ=0.78 inter-annotator agreement, revealing that most failures stem from natural descriptions generated by the Description Generator, specifically semantic mismatch due to missing industry-specific terminology and domain jargon (45%), visual description gaps from inadequately described complex visual elements and abstract design concepts (30%), and metadata inconsistency from misaligned template tags and categorical misclassification (25%).

4.6 Scalability and Trade-off Analysis

Scalability Considerations: While our approach shows consistent improvements on the Crello dataset (+33.8% MRR), query length as a proxy for intent may not capture all nuances of user information needs. Analysis of query types reveals 89% accuracy in intent prediction for general domains, but performance varies for specialized vocabularies. Statistical correlation analysis shows r=0.82 between query length and semantic complexity in our test set. Future work could incorporate semantic intent classification using BERT-based query understanding or multi-task learning for more sophisticated field weighting strategies.

Computational Trade-offs: The hybrid ranking approach increases inference time by 27% compared to pure bi-encoder methods (156ms vs 123ms per query), with 25ms attributed to cross-encoder reranking and <1ms to field-adaptive computation. However, this overhead is justified by the 33.8% performance improvement, resulting in a performance-to-latency ratio of 1.25 compared to 0.87 for ColBERT and 0.73 for MFAR. The approach maintains sub-200ms response times essential for interactive applications while delivering production-grade accuracy improvements.

5. CONCLUSION

This paper introduces a novel field-adaptive methodology for the dense retrieval of structured documents, effectively addressing the semantic gap between natural language queries and structured content. Our approach combines three key innovations: (1) fine-tuned language models for generating reliable natural descriptions and queries, (2) query-length-based adaptive field weight-

ing for optimal embedding representation, and (3) a hybrid ranking strategy that balances efficiency and precision.

Experimental results on the Crello dataset demonstrate substantial improvements, with the complete system achieving a 33.8% MRR enhancement over baseline approaches. The field-adaptive embedding strategy proves particularly effective, while our fine-tuned generators show remarkable performance gains of up to 35.1% in F1-score. The hybrid ranking approach successfully combines the efficiency of bi-encoders with the precision of cross-encoders, maintaining practical inference speeds while improving retrieval accuracy.

Our methodology offers a scalable solution for structured document retrieval that can be applied across various domains without the need for domain-specific modifications.

The comprehensive experimental validation reveals that adaptive field weighting based on query characteristics significantly outperforms fixed-weight alternatives, with the complete field combination (title, description, metadata) yielding optimal performance across diverse query types. The synthetic training data generation pipeline, utilizing fine-tuned language models with sophisticated filtering mechanisms, successfully addresses the fundamental challenge of training data scarcity while maintaining semantic fidelity to source documents. The optimized hybrid ranking strategy ($\lambda_{cross} = 0.8, \lambda_{bi} = 0.2$) represents a carefully calibrated balance that maximizes retrieval effectiveness while preserving computational feasibility essential for production deployment.

As shown in Subsection 4.5, filtering out irrelevant natural descriptions needs to be more improved, and further more careful analysis about comparison against some of conventional direct utilization of structured documents in bi-encoder and cross-encoder In addition, future work will focus on extending this approach to multimodal structured documents and investigating dynamic field weighting based on query semantics rather than length alone.

Our failure analysis of 100 retrieval cases identifies three primary limitation categories: semantic mismatch in generated descriptions (45%), visual description inadequacy (30%), and metadata inconsistency (25%). These findings provide transparent assessment of methodology boundaries and guide future research directions. The predominance of semantic mismatch failures highlights ongoing challenges in automated content generation, while visual description limitations point toward natural extensions to multimodal retrieval scenarios. Future research directions include developing more sophisticated query intent recognition for dynamic field weighting, incorporating multimodal elements for comprehensive document representation, and implementing reinforcement learning approaches for adaptive weighting strategies that can learn from user interaction patterns and satisfaction signals.

REFERENCE

[1] C.D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[2] W. Zhao, J. Liu, R. Ren, and J.R. Wen, "Dense Text Retrieval Based on Pretrained Language Models: A Survey," Transactions on Information Systems, Vol. 42, No. 4, pp. 1-60, 2024.

[3] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.T. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 6769-6781, 2020.

[4] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction Over BERT," Proceedings of the 43rd International ACM SIGIR Conference on Research and Develop-

ment in Information Retrieval, pp. 39-48, 2020.

[5] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins, "Sparse, Dense, and Attentional Representations for Text Retrieval," Transactions of the Association for Computational Linguistics, Vol. 9, pp. 329-345, 2021.

[6] X. Li, Z. Liu, C. Xiong, S. Yu, Y. Gu, Z. Liu, et al., "Structure-Aware Language Model Pretraining Improves Dense Retrieval on Structured Data," arXiv preprint arXiv:2312.09249, 2023.

[7] M. Li, T. Chen, B. Van Durme, and P. Xia, "Multi-Field Adaptive Retrieval," arXiv preprint arXiv:2501.xxxxx, 2025.

[8] "Crello Dataset," Hugging Face Datasets, available at https://huggingface.co/datasets/cyberagent/crello(accessed August 11, 2025).

[9] "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors," Facebook AI Research, available at https://github.com/facebookresearch/faiss(accessed August 12, 2025).

[10] "gemma-3-4b-it," available at https://huggingface.co/google/gemma-3-4b-it(accessed August 11, 2025).

[11] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.

[12] "all-MiniLM-L6-v2," available at https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2(accessed August 11, 2025).

[13] H. Meghwani, A. Agarwal, P. Pattnayak, H.L. Patel, and S. Panda, "Hard Negative Mining for Domain-Specific Retrieval in Enterprise Systems," arXiv preprint arXiv:2501.xxxxx, 2025.

[14] "ms-marco-MiniLM-L6-v2," available at https://www.sbert.net/docs/cross_encoder/pretrained_models.html#ms-marco(accessed August 11, 2025).

[15] E.M. Voorhees, "The TREC-8 Question Answering Track Report," in Proceedings of the 8th Text REtrieval Conference (TREC-8), 1999, pp. 77-82.

[16] L. Wang, Y. Zhang, J. Liu, W.X. Zhao, and H. Wang, "DEBATER: Deliberate Thinking for Document Embeddings," arXiv preprint arXiv:2502.12974, 2025.

Syed Mudasir

He received a B.C.S. degree in Computer Science from the University of Sindh, Pakistan, in 2015, and an M.Sc. degree in Computer Science from Muhammad Ali Jinnah University, Pakistan, in 2018. He is currently pursuing an MS. degree in AI Convergence at Soongsil University, Seoul, South Korea, since March 2024. His research interests include machine learning, generative AI, and mobile application development.

Aagha Abdual Waheed

He is a Ph.D. researcher in Computer Vision and Natural Language Processing at Soongsil University, Seoul, South Korea. His research focuses on multimodal learning, document understanding, and scalable deep learning systems, with professional experience in full-stack development and AI-driven applications.

Syed Muzamil Hussain

He received a B.E. degree from Mehran University, Pakistan, in 2013, an M.S. degree from NED University, Pakistan, in 2017, and a Ph.D. degree from Soongsil University, Seoul, in 2022. He is currently an NLP/AI Engineer at Archipin, Inc., Seoul. His research interests include NLP, LLM, and Conversational AI.

Sun-Tae Chung

He received B.E. degree from Seoul National University, and M.S. degree and Ph.D. degree in Electrical Eng. and Computer Science from the University of Michigan, Ann Arbor, USA, in 1986 and 1990, respectively. Since 1991, he had been with the Department of AI Convergence. at the Soongsil university, Seoul, Korea where he is now a full professor. Now, he has been with the Dept. of AI Convergence, at the Soongsil Univ. since 2015. His research interests include: agentic AI, multimodal large language models, and AI digital marketing.