JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

An Intelligent Semantic Search System for Digital Product Part Design Catalogues in Manufacturing

Pablo de Vicente Abad

Modeling Intelligent Complex Software and Systems (MICCS-Lab), University of Antwerp, Belgium

Promotor: Prof. Moharram Challenger
Co-Promotor: Alireza Khalilipour

Abstract This paper introduces a modular framework for semantic similarity search over product catalogues, designed to robustly extract and structure content from PDFs, including text, images, and tables. The framework supports a range of retrieval strategies, spanning traditional methods like Vector Space Model (VSM) and BM25, fusion-based approaches such as Reciprocal Rank Fusion (RRF), and hybrid pipelines with reranking modules. It accommodates customizable configurations, including document representation through single-vector or multi-vector embeddings, as well as optional normalization and semantic query expansion. Through systematic evaluation, we observe that retrieval performance varies significantly depending on the dataset characteristics and the intended use case; whether prioritizing precision, recall, interpretability, or scalability. Rather than prescribing a one-size-fits-all solution, the framework is designed to support side-by-side comparisons of diverse strategies under a unified experimental setup. This allows practitioners and researchers to empirically determine which configurations are most effective for their specific needs. By offering flexibility in method selection and parameterization, the framework not only supports rigorous benchmarking but also enables iterative refinement and adaptation to evolving data and retrieval requirements.

Keywords document embeddings, hybrid retrieval, multi-vector encoding, semantic query expansion, vector space models.

1 Introduction

Technical PDFs, such as product catalogues, often embed critical information in tables, images, and structured captions. Despite the availability of PDF-to-text tools, most fail to fully extract these elements, leading to incomplete document representations that hinder downstream tasks like semantic search. To address this challenge, we introduce **Tables-Text-Images.txt (TTI.txt)**, a pipeline that converts all content in a technical PDF—including textual descriptions, tabular data, and visual elements—into a unified, structured raw-text format.

Building on this representation, we present **Semantic Search Exploration (SSE)**, a modular framework for evaluating document retrieval methods over technical corpora. SSE supports traditional lexical models (e.g., BM25), dense vector embeddings, multi-vector segmentation for fine-grained retrieval, and hybrid ranking strategies such as score fusion and Reciprocal Rank Fusion (RRF). It also enables semantic query expansion and cascaded re-ranking, offering a flexible testbed for retrieval research.

Our system is designed with technical product catalogues (PCs) in mind—richly formatted documents that specify everything from fasteners to electromechanical assemblies. These documents encode part specifications across text, tables, and diagrams. *TTI.txt* captures these modalities in plain text, making them accessible to search engines.

To support reproducibility and scalability, our dataset builder automatically crawls web or repository URLs, extracts linked PDFs, and processes them using *TTI.txt*. The resulting corpus serves as input to *SSE*, where retrieval strategies can be evaluated and compared under real-world conditions.

In sum, this paper introduces a complete pipeline—from enhanced content extraction to retrieval evaluation—tailored for complex, multimodal PDFs. By unifying fragmented document elements and offering a flexible retrieval framework, our approach lowers the barrier to building and benchmarking robust semantic search systems in technical domains.

2 Literature Review

Recent advances in document retrieval have focused on improving query formulation, document representation, and ranking effectiveness. To bridge the vocabulary gap between user queries and document content, researchers have explored techniques such as query expansion, term re-weighting, and semantic normalization. Early approaches used pseudo-relevance feedback to enrich queries with terms from top-ranked documents [**Rocchio, 1971**, **Lavrenko & Croft, 2001**], while more recent work leverages word embeddings and contextual language models like Word2Vec and BERT for semantic expansion [**Mikolov et al., 2013**, **Devlin et al., 2019**].

Parallel efforts in document representation have transitioned from traditional sparse models (e.g., TF-IDF, BM25) [**Järvelin & Kekäläinen, 2002**, **Robertson & Zaragoza, 2009**] to dense neural embeddings that encode semantic meaning in continuous space [**Le & Mikolov, 2014**]. Studies comparing single-vector and multi-vector approaches highlight trade-offs between retrieval granularity and system efficiency [**Karpukhin et al., 2020**, **Zhou & Devlin, 2021**].

On the retrieval side, hybrid architectures that combine sparse and dense methods have shown improved robustness across query types. Techniques such as linear score fusion [**Wu, 2012**], Reciprocal Rank Fusion (RRF) [**Cormack et al., 2009**], and neural re-ranking pipelines [**Nogueira & Cho, 2019**, **Zhang et al., 2024**, **Rao et al., 2025**] exemplify efforts to balance precision, scalability, and adaptability. Together, this literature underscores the importance of modular retrieval systems capable of integrating heterogeneous signals.

2.1 Semantic Query Processing

Semantic search enhances retrieval by interpreting user intent and term meaning, rather than relying solely on lexical overlap [**Manning et al., 2008**, **Guo et al., 2016**]. While many systems personalize results using user-specific signals such as location or browsing history [**Bennett et al., 2012**, **Khattab & Zaharia, 2020**], our setting assumes no such external context. Instead, we adopt a corpus-centric approach that draws semantic signals directly from the indexed data.

A key component is intelligent query expansion, which improves recall and alignment with domain vocabulary. Rather than relying on external linguistic resources like WordNet [**Voorhees, 1994**], we implement Intelligent Substitution—a method that identifies high-cooccurrence term variants within the corpus itself [**Azad & Deepak, 2021**]. For instance, "3 mm bolt" may be expanded to include "0.118 inch bolt" if such equivalences are frequent in the dataset. Similarly, lexical relations such as "bolt" and "fastener" are discovered through syntagmatic co-occurrence patterns [**Fang et al., 2006**]. This localized expansion strategy tailors query representation to the semantic structure of the corpus, improving retrieval effectiveness in domain-specific settings.

2.2 Document Indexing

Semantic retrieval systems rely on document embeddings to represent textual content in a contin-

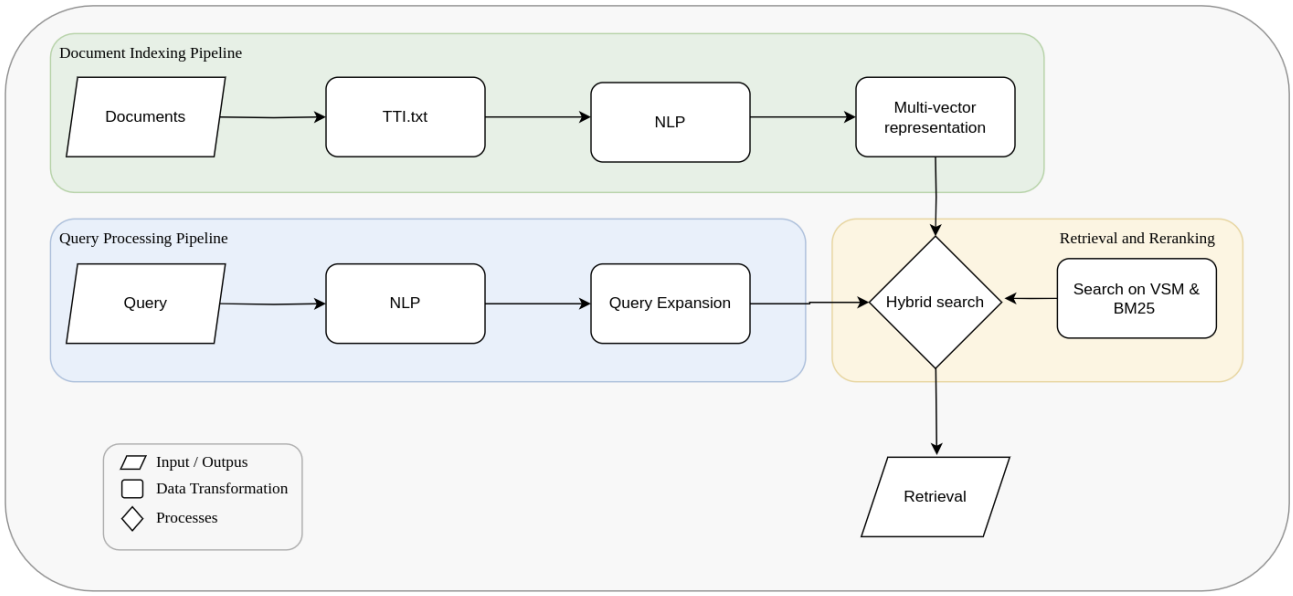

Fig.1. Semantic Search Evaluation (SSE) overview

uous vector space, enabling comparisons based on meaning rather than surface-level lexical similarity [**Manning et al., 2008**, **Devlin et al., 2019**]. We implement two complementary embedding strategies to support different retrieval granularity requirements.

2.2.1 Single-Vector Embeddings

Each document is first encoded as a single fixed-length vector, capturing a global semantic representation [**Le & Mikolov, 2014**, **Guo et al., 2016**]. This approach offers computational efficiency and serves as a strong baseline for dense retrieval. However, compressing entire documents into one vector often limits its sensitivity to localized semantics—particularly in technical documents with diverse or multi-topic content [**Fang et al., 2006**].

2.2.2 Multi-Vector Embeddings

To improve granularity, we segment documents into smaller units (e.g., sections or paragraphs) and embed each segment independently. This multivector approach enables more precise query-document alignment by allowing retrieval at the sub-document level [**Nogueira & Cho, 2019**]. Prior work has explored this strategy within hybrid retrieval architectures, demonstrating its potential in high-variance domains [**Karpukhin et al., 2020**, **Zhong et al., 2020**]. By integrating multivector indexing into our pipeline, we aim to capture finer semantic signals while maintaining flexibility in retrieval granularity.

2.3 Document Retrieval

Document retrieval is the foundational step in semantic search, responsible for identifying candidate documents based on their similarity to a user query. By mapping both queries and documents into either sparse term-based or dense vector representations, retrieval models enable efficient filtering of large corpora prior to re-ranking.

2.3.1 Dense Retrieval (Vector Space Models)

Vector space models (VSMs) project text into a high-dimensional embedding space, allowing semantic similarity to be measured through metrics such as cosine similarity or Euclidean distance

[**Salton et al., 1975**, **Singhal, 2001**]. In our framework, we apply Word2Vec-based embeddings to capture context-sensitive meanings beyond surface-level term overlap [**Mikolov et al., 2013**]. Dense retrieval excels at identifying semantically related content, even when there is little or no exact lexical match.

2.3.2 Sparse Retrieval (BM25)

BM25 remains a strong baseline in sparse retrieval, ranking documents based on term frequency, inverse document frequency, and document length normalization [**Robertson & Zaragoza, 2009**]. Its strength lies in its ability to prioritize exact term matches—particularly valuable in technical domains where precision vocabulary matters [**Guo et al., 2016**]. Despite the growth of neural methods, BM25 remains computationally efficient, scalable, and interpretable, making it suitable for real-time and large-scale search scenarios.

Together, these retrieval strategies offer complementary strengths: dense models provide semantic generalization, while BM25 ensures lexical fidelity. In subsequent stages, we explore hybrid and cascaded approaches that leverage both paradigms for improved retrieval performance.

2.4 Hybrid and Multi-Stage Retrieval

Retrieval performance can be improved by combining the complementary strengths of lexical and semantic models. We explore several hybrid and re-ranking strategies that integrate dense embeddings with sparse representations.

2.4.1 Hybrid Score Fusion

Hybrid retrieval linearly combines BM25 scores with dense similarity scores to balance exact matches and semantic relevance. Given a query q and a document d, the hybrid score is computed as:

$$\text{score}_{\text{hybrid}}(q,d) = \lambda\, \text{score}_{\text{BM25}}(q,d) + (1-\lambda)\, \text{sim}_{\text{dense}}(q,d), \tag{1}$$

where $\lambda \in [0,1]$ controls the weighting between sparse and dense components. The dense similarity $\text{sim}_{\text{dense}}$ is measured using cosine similarity over pre-trained embeddings (either single-vector or multi-vector), while BM25 scores are computed as:

$$\sum_{t \in q} \frac{f(t,d)(k_1+1)}{f(t,d) + k_1 \left(1 - b + b\frac{|d|}{L}\right)} \cdot \log \frac{N - n_t + 0.5}{n_t + 0.5}, \tag{2}$$

following standard probabilistic retrieval theory [**Robertson & Zaragoza, 2009**]. To ensure robustness across datasets, score normalization (e.g., min-max scaling) and careful tuning of λ are required [**Cormack et al., 2009**].

2.5 Reciprocal Rank Fusion (RRF)

RRF merges the ranked outputs of multiple retrieval models without relying on score normalization. For each document d, if its rank from system i is $r_i(d)$, the RRF score is given by:

$$\text{score}_{\text{RRF}}(d) = \sum_i \frac{1}{k + r_i(d)}, \tag{3}$$

where k is a smoothing parameter (typically $k = 60$) that controls the influence of top-ranked documents [**Cormack et al., 2009**]. RRF is robust to differences in score scales and tends to favor documents that are consistently ranked highly across systems.

2.6 Re-ranking

To refine initial results, we implement a two-stage retrieval pipeline. In the first stage, a fast retriever (e.g., BM25 or RRF) selects the top-K documents:

$$\{d_1, \ldots, d_K\} = \text{TopK}(\text{score}_{\text{first}}(q, D)). \tag{4}$$

These candidates are then re-ranked using a more computationally intensive model. Relevance scores are assigned as:

$$\text{score}_{\text{re}}(q, d_j) = \text{MLP}([\mathbf{v}_q; \mathbf{v}_{d_j}]), \tag{5}$$

or computed using joint input encodings in a fine-tuned cross-encoder [**Nogueira & Cho, 2019**]. This approach yields higher precision but incurs greater computational cost.

3 Methodology

Our Semantic Search Exploration (SSE) framework integrates query processing, document indexing, and hybrid retrieval into a unified pipeline designed for semantic search over technical PDF collections (see Figure 1). The system comprises three key components:

3.1 Query Processing

User queries undergo lexical normalization—including tokenization and stemming—followed by semantic enrichment. We compute term frequency-inverse document frequency (TF-IDF) weights to emphasize informative terms, and perform query expansion using high-cooccurrence substitutions extracted from our domain corpus [**Singhal, 2001**]. This expansion aids in bridging vocabulary mismatches and capturing equivalent expressions (e.g., "3 mm" $\approx$ "0.118 inch").

3.2 Document Indexing

Each PDF is converted into a unified text stream using our `TTI.txt` tool. This stream is subsequently preprocessed via tokenization and stop-word removal prior to embedding.

Global Embeddings: Entire documents are embedded as single vectors using TF-IDF-weighted Word2Vec representations. These offer compact yet expressive semantic summaries.

Segmented Embeddings: To enhance retrieval precision for heterogeneous content, documents are partitioned—by section, page or token count—and embedded segment-wise. This results in multiple vectors per document, enabling finer-grained matching.

3.3 Retrieval and Re-ranking

In our evaluation of document retrieval methods, we compare traditional sparse models with modern dense approaches and explore hybrid strategies that capitalize on their complementary strengths. Initially, we apply BM25 to generate a candidate set based on term-matching scores, alongside a dense retrieval step that ranks documents by the cosine similarity of their embedding vectors. To balance precision and semantic coverage, we then fuse these two signals: score fusion linearly combines sparse and dense scores to produce a single, harmonized ranking, while Reciprocal Rank Fusion (RRF) merges the separate rank lists by rewarding documents that consistently appear near the top across both methods. Finally, we implement a two-stage ranking pipeline in which BM25 retrieves an initial shortlist that is subsequently re-ranked according to embedding-based similarity, thus ensuring both efficiency in candidate selection and semantic refinement in the final ordering.

These methods allow for flexible experimentation and have demonstrated significant gains in retrieval effectiveness. Integration of advanced neural re-rankers and graph-based models is considered for future work.

4 Dataset Overview

To support our semantic retrieval experiments, we constructed a labeled dataset of technical PDF documents sourced from DigiKey's publicly accessible product pages. Each document corresponds to a specific

Method	Lexical Fidelity	Semantic Matching	Computational Cost
Dense Embeddings	Low	High	Medium
BM25	High	Low	Low
Hybrid	Medium	Medium	Medium
RRF	Varies	Varies	Low
Re-ranking	Medium	High	High

Fig.2. Comparison of Retrieval Methods

product category (e.g., microphones, cable ties, batteries) and is treated as a distinct class label. After filtering and validation, the dataset includes approximately 100 high-quality PDFs per category, with each file averaging five pages in length and containing rich structural elements such as specifications, tables, figures, and diagrams.

4.1 Data Collection and Labeling

Documents were collected by programmatically crawling DigiKey's category-specific URLs, which paginate product listings through predictable query parameters. This URL-driven approach allowed traversal of product pages without the need for browser automation. For each product entry containing a linked PDF, we downloaded the document and automatically assigned it the category label inferred from the source URL.

To maintain label quality, documents appearing under multiple categories were excluded. We further ensured format consistency by discarding non-PDF links and implemented checksum-based deduplication to eliminate redundant downloads. A random 10% subset of the dataset was manually reviewed to verify label accuracy, and any misclassified or ambiguous entries were corrected or removed. The final dataset is balanced across diverse technical classes, including categories such as *batteries*, *battery-chargers*, *cable-ties*, *microphones*, *cables*, and *printers*.

4.2 Preprocessing and Content Extraction

To enable retrieval experiments, all PDFs were transformed into structured raw-text files using our Tables-Text-Images.txt (TTI.txt) pipeline. The goal of this transformation is to preserve semantic content across multiple modalities—text, tables, and figures—by rendering them into a unified plain-text format. The extraction process is modular and includes specialized routines for each content type, as described in the following sections.

4.2.1 Text Extraction

Extracting high-quality raw text from technical PDFs is essential for reliable indexing and retrieval. To identify the most effective tool for this task, we benchmarked four widely used Python libraries for PDF parsing on a representative subset of 50 documents, including single-column reports, table-heavy datasheets, and mixed-layout manuals.

After comparative evaluation, PDFPlumber was selected due to its robust handling of both narrative and structured content. It consistently captured the majority of textual content, including embedded tables and multi-column layouts, while preserving token integrity—an essential requirement for our Bag-of-Words (BoW) indexing approach. Because BoW models emphasize term presence rather than structure, minor deviations in sentence boundaries or layout had minimal impact on retrieval quality.

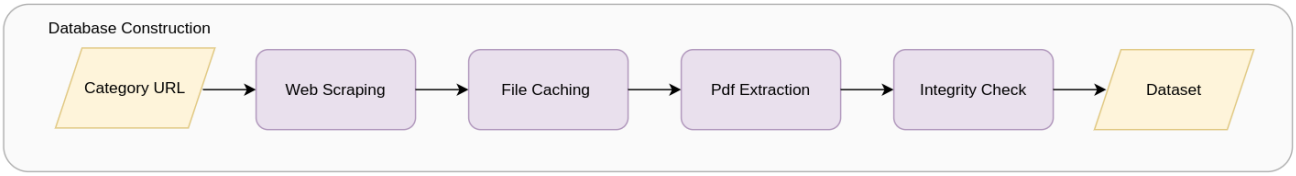

Fig.3. Dataset construction pipeline

To validate extraction accuracy, we conducted a manual audit on a random sample of parsed files. Over 95% of words were correctly extracted across diverse formatting styles, with fewer than 5% of tokens classified as noise (e.g., stray symbols or duplicated bullets). These residual artifacts were systematically removed using a custom preprocessing pipeline that applied regular-expression filters and token-level cleaning.

This preprocessing step ensures that the resulting vocabulary is both comprehensive and noise-free, forming a reliable foundation for downstream indexing, retrieval, and embedding workflows.

4.2.2 Table Extraction and Processing

Extracting structured data from technical PDFs poses significant challenges due to inconsistent formatting, varied layouts, and the absence of standardized table markup. Open-source tools often struggle with edge cases such as merged cells, nested headers, or sparsely populated columns. To address this, we implemented a multi-stage pipeline combining rule-based preprocessing with model-driven conversion. Tables were first extracted using the `pdfplumber` library, which provided reliable parsing across diverse formats. The extracted content was then cleaned to standardize structure—removing noisy columns, resolving alignment issues, and imputing or excluding missing values .

To ensure compatibility with downstream summarization models, we applied heuristics adapted from [**Su et al., 20242**] to filter unusable tables. Specifically, we excluded tables with fewer than five rows or two columns, more than 30% missing values, or header repetition in the first row. Valid tables were then categorized into structured and sparse types to enable tailored processing by appropriate language models.

Class 1 (Structured): Large, grid-like tables processed with TableGPT2-7B, optimized for detailed row-wise summaries (Figure 4).

Class 2 (Sparse/Irregular): Small, questionnaire-style tables handled by Qwen2.5-7B-Instruct, a general-purpose instruction-tuned model (Figure 5).

Item #	Color	Putup Type	Length	UPC
84316 001100	Brown	Reel	100 ft	612825207016
84316 001500	Brown	Reel	500 ft	612825207139
84316 0011000	Brown	Reel	1,000 ft	612825207023

Fig.4. Class 1 example: structured product specification table

Specifications

Classification:	"Lithium"
Chemical System:	Lithium / Manganese Dioxide (Li/MnO_2)
Designation:	ANSI-5046LC, IEC-CR15H270
Nominal Voltage:	3.0 Volts
Storage Temp:	-40°C to 60°C (-40°F to 140°F)
Operating Temp:	-40°C to 60°C (-40°F to 140°F)
Typical Capacity:	800 mAh (to 2.0 volts) (Rated at 100 ohms at 21°C)
Typical Weight:	11.0 grams (0.4 oz.)
Typical Volume:	5.2 cubic centimeters (0.3 cubic inch)
Max Discharge:	1000 mA continuous (2500 mA pulse)
Max Rev Current:	2 uA
Typical Li Content:	0.28 grams (0.010 oz.)

Fig.5. Class 2 example: compact, questionnaire-style layout

Table-to-Text Conversion and Model Selection: To convert tabular data into natural language, each extracted table was passed to a language model using the prompt: *"Describe the contents of each row in a technical manner."* The resulting summaries were appended to the base document text, preserving tabular semantics in a unified text format suitable for retrieval. For structured tables, we employed **TableGPT2-7B**, a model pretrained specifically for tabular summarization, while smaller or irregular tables were handled using **Qwen2.5-7B-Instruct**, a more general-purpose instruction-tuned model.

We benchmarked multiple alternative models—including LLaMA-3.1, LLaMA-3.2, Bloom, and ChatGPT—on representative product tables. Models without explicit training on structured data frequently produced hallucinated outputs or failed to capture key relational content. These findings align with prior work showing that most large language models are optimized for QA or fact verification, not descriptive summarization. Our results underscore the importance of using structure-aware models for accurate table-to-text generation in technical domains.

5 Experimental Setup

This study evaluates the effectiveness, efficiency, and trade-offs of various document retrieval strategies in the context of technical product catalogues. Such documents often exhibit high lexical similarity, with domain-specific phrasing and structured descriptions, making retrieval particularly challenging. Our goal is to assess retrieval performance across both exact and semantic query types, while also considering practical factors such as scalability and computational cost.

We evaluate five retrieval strategies: (1) BM25 as a sparse lexical baseline, (2) dense retrieval using a Vector Space Model (VSM) with embeddings, (3) hybrid retrieval via linear score fusion of BM25 and VSM, (4) two-stage retrieval combining BM25 with semantic re-ranking, and (5) Reciprocal Rank Fusion (RRF), which merges ranked outputs across models. Additionally, we examine the impact of document encoding strategies (single-vector vs. multi-vector), the influence of table-to-text conversion using language models, and the effects of query expansion and score normalization on retrieval performance.

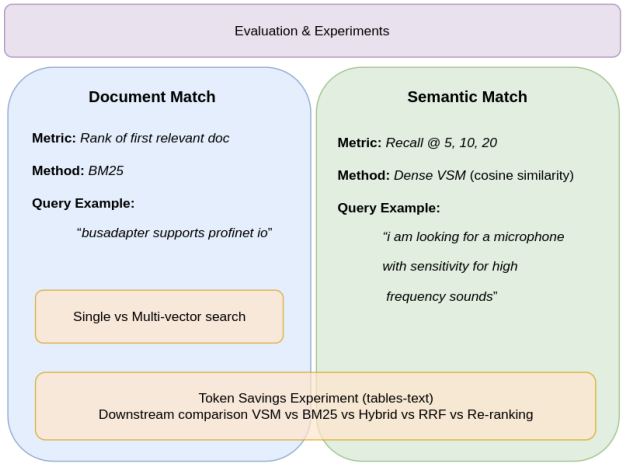

Fig.6. Overview of experimental design and evaluation scenarios

5.1 Evaluation Design

To evaluate retrieval performance without relying on expert annotators, we leverage the dataset's labeled structure and define two query types:

5.1.1 Exact Match Scenario

A sentence is randomly selected from a document and used as the query. The task is to retrieve the original document from the corpus. **Example queries:**

```
buzzer with operating voltage 15 24 vdc
```

```
busadapter supports protocol for profinet io
```

5.1.2 Semantic Match Scenario

A high-level product description is provided as the query. The task is to retrieve documents that fall under the correct product category. **Example query:**

```
i am looking for a battery with a lifespan of
more than 7000 minutes at room temperature
```

We use Recall@K (with K = 5, 10, and 20) as the primary evaluation metric, reflecting typical user behavior in ranked retrieval systems, where attention is concentrated on the top results. For each query, we record the rank position of the original document in the exact match scenario, the number of top-K results belonging to the correct category in the semantic match scenario.

This dual evaluation strategy highlights how different retrieval methods respond to lexical versus semantic queries. BM25 tends to perform well on exact matches due to its reliance on term overlap, while dense embedding methods are better suited for semantically abstract queries. Hybrid and re-ranking methods are evaluated for their ability to bridge these strengths.

6 Experimental Results and Analysis

Having outlined the relevant literature and described the methodology underlying the SSE and TTI pipelines, we now turn to a detailed examination of the experimental results. While the specific setup and procedures were discussed in the previous section, this part focuses on analyzing the outcomes of the experiments, offering insights into the performance and implications of the proposed approaches.

Impact of Embedding Strategies: A comparative analysis of single-vector versus multi-vector embedding techniques and their influence on retrieval performance.

Natural Language Translation of Tabular Data: Estimation of performance gains when converting structured tabular data into natural language formats using large language models.

Semantic Expansion Techniques: An evaluation of how semantic expansion influences retrieval effectiveness within the specific context of our dataset.

Retrieval Strategy Comparison: A study comparing various retrieval approaches, with an emphasis on the role and efficacy of score normalization techniques.

6.1 Single vs. Multi-Vector Embedding Performance

Initial experiments revealed minimal performance differences between single-vector and multi-vector document embeddings when applied to text-only content. This is likely due to the relatively homogeneous nature of technical product documents, where information types (e.g., specifications, descriptions) are consistently distributed across sections. Consequently, the benefits of segment-level embeddings remain limited in unimodal settings.

However, embedding performance may be significantly improved in a multimodal retrieval framework. By assigning separate vectors to different content types—such as narrative text, tables, and figures—future systems could enable more nuanced indexing and support multimodal queries (e.g., retrieving documents via tabular or visual input). Incorporating modality-specific embedding strategies may yield a more complete and discriminative semantic representation of each document.

Additionally, optimization opportunities remain within the embedding pipeline itself. Reducing segment overlap, increasing chunk granularity, and weighting structurally salient sections (e.g., titles, abstracts) may further enhance semantic indexing. For instance, combining vectors from title and abstract sections, alongside weighted paragraph-level vectors, could improve precision in high-recall retrieval tasks. Due to current

pipeline constraints, these enhancements remain areas for future exploration.

6.2 Impact of Table-to-Text Conversion on Token Count

To assess the impact of table-to-text conversion on document length and retrieval richness, we classified tables into two broad categories based on structural complexity and information density. After filtering non-informative or redundant entries, the dataset yielded an average of approximately four meaningful tables per document.

Each translated table contributed an estimated 150-200 words, resulting in an average addition of 600 words per document—representing a 60-100% increase over the original text length. This substantial enrichment underscores the value of structured data in improving retrieval performance.

Qualitative review suggests that generated descriptions are generally informative and contextually relevant. These translations often include identifiers and technical attributes (e.g., pressure class, thread size) that enhance document retrievability.

Nonetheless, occasional hallucinations were observed, where the language model inferred or inserted unverified terms. While these instances were typically minor and domain-consistent, they highlight a trade-off between informativeness and fidelity in automated table interpretation. Despite this, the overall effect of table-to-text translation is positive, as it provides structured semantic signals that are otherwise inaccessible to retrieval models operating on plain-text content alone.

6.3 Semantic Expansion of Queries

Semantic query expansion aims to improve retrieval performance by enriching user queries with contextually related terms drawn from the corpus itself. In our approach, we compute expansions by identifying nearest neighbors in a vector space model (VSM) trained on the corpus vocabulary. Cosine similarity is used to select terms that are semantically close to those in the original query. This corpus-specific method allows for domain-aware expansion, capturing technical synonyms, abbreviations, and variant expressions.

The effectiveness of expansion depends heavily on corpus size and diversity. Larger, more heterogeneous corpora provide better contextual signals, yielding more meaningful term associations. Conversely, in smaller datasets, semantic neighbors may reflect surface-level co-occurrence rather than true conceptual similarity.

This method allows for the discovery of domain-relevant alternatives, such as "specs" for "specifications" or "voltages" as a contextual equivalent for "VDC." Although such terms may not be standard dictionary entries, they are commonly used in technical literature and correctly inferred by the VSM due to their contextual embedding.

However, limitations emerge in low-resource settings. In smaller corpora, where vocabulary and usage diversity are constrained, the VSM may produce loosely related expansions. For instance, the term "capacitance" may be returned in response to a query about "voltage" not due to synonymy but because of frequent co-occurrence in electrical component descriptions. This effect, known as semantic drift, poses a risk of introducing noise rather than improving retrieval.

Despite its theoretical advantages, semantic expansion yielded only marginal gains in our current experimental setup. We attribute this to the relatively limited size of our corpus, which restricted the contextual precision of the VSM. Nevertheless, the method holds promise for larger or more diversified datasets, where expansion terms are more likely to be semantically aligned with user intent.

6.4 Ranking Strategies: Exact-Match Evaluation

To assess the effectiveness of different ranking strategies under exact-match conditions, we compared five models using a scenario in which queries consisted of verbatim sentences drawn from the documents. Figure 7 summarizes the comparative performance across models.

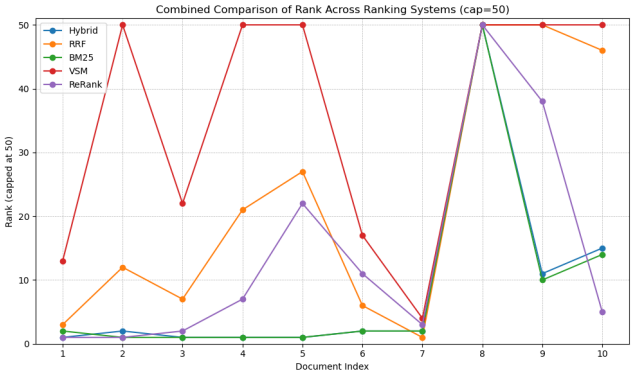

Fig.7. Model ranking performance under exact-match scenario.

BM25 consistently outperformed other approaches in this setting, reliably ranking the correct document near the top. Its reliance on exact term matching and length normalization proved well-suited for pinpointing literal content. In contrast, the Vector Space Model (VSM), which uses cosine similarity over TF-IDF vectors, showed weaker performance. The inclusion of all document terms diluted the influence of the exact query sentence, often resulting in lower ranks for the correct document.

Hybrid models, combining BM25 and VSM scores, achieved moderate improvements over VSM alone but did not surpass BM25 in precision. These results suggest that in exact-match scenarios—where users expect high lexical fidelity—BM25 offers a clear advantage due to its stronger emphasis on query term frequency and position.

In summary, BM25 remains the most reliable method for exact-match retrieval, while vector-based and hybrid approaches may introduce unnecessary noise in this context. In subsequent evaluations, we explore how these alternative strategies perform under semantically oriented queries, where exact word overlap is less important.

6.5 Ranking Strategies: Semantic-Match Evaluation

In the semantic match scenario, our primary goal is to evaluate how effectively each model retrieves products that are similar in nature rather than requiring an exact text match. Because many products in our corpus share highly similar descriptions (for example, there is often little substantive difference between various types of zip ties), the emphasis shifts from pinpointing the exact document to capturing items belonging to the same labeled class. Consequently, we focus on recall for the target label class as the key metric, treating the retrieval step as a first-pass filter. Once the top k results are returned, a user can inspect those candidates and identify the most appropriate item, much as one would browse the first few pages of results in a typical web search.

Figures 8 and 9 compare recall@5, recall@10, and recall@20 for both the Vector Space Model (VSM) and BM25. These plots reveal the advantages of VSM in handling loose, semantically driven queries: although BM25 performs strongly when exact terms are critical, VSM often achieves higher recall for queries that require a broader notion of similarity. In particular, at certain cutoff levels (e.g @10), VSM outperforms BM25, demonstrating its ability to capture related items even when the query vocabulary does not align perfectly with the indexed text.

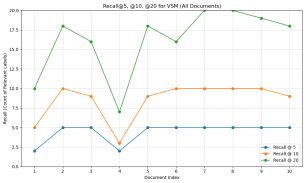

Fig.8. VSM Recall

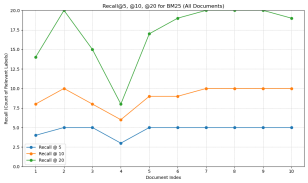

Fig.9. BM25 Recall

6.6 Ranking Strategies: Hybrid Retrieval

To leverage the complementary strengths of lexical and semantic models, we implemented a hybrid retrieval strategy combining BM25 and VSM scores through weighted linear fusion. In our experiments, we used weights of 0.7 for BM25 and 0.3 for VSM. These values were selected heuristically and can be adjusted based on the characteristics of the target domain. For instance, in domains requiring precise term matching (e.g., legal documents), a higher weight on BM25 is likely preferable, while semantically rich tasks may benefit from emphasizing VSM.

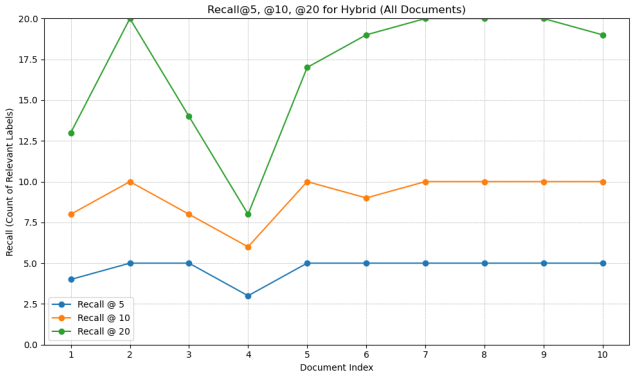

Fig.10. Retrieval performance of hybrid model (BM25 + VSM), measured by Recall@K.

The hybrid model achieved the highest overall retrieval performance across both exact and semantic match scenarios. As illustrated in Figure 10, hybrid retrieval consistently outperformed either BM25 or VSM alone, particularly in cases where partial semantic overlap complemented exact lexical matches. For example, document-level recall improved for several queries under hybrid scoring, with notable gains at top-K ranks.

These results support the use of hybrid ranking as a balanced strategy that adapts well to mixed query types, offering improved recall while preserving interpretable scoring contributions from both lexical and semantic features.

Appendix: Score Normalization in Hybrid Retrieval

Combining BM25 and VSM scores in hybrid retrieval requires normalization to align their disparate scales—BM25 scores being unbounded and VSM similarities bounded in $[0, 1]$. We evaluated two standard normalization techniques: Min–Max scaling, which linearly maps scores to $[0, 1]$, and Z-score normalization, which standardizes scores to zero mean and unit variance. These methods were tested in various pairwise combinations on both components.

As shown in Figure 11, normalization choice significantly affects hybrid retrieval performance. The best results were achieved by applying Min–Max followed by Z-score normalization, while the weakest performance arose when Z-score was followed by Min–Max. These findings highlight the sensitivity of hybrid scoring to normalization pipelines and suggest that consistent performance may require tuning based on corpus-specific score distributions.

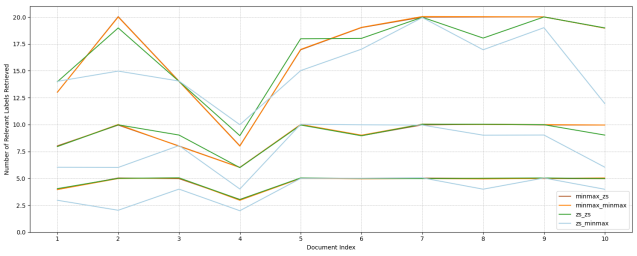

Fig.11. Recall@K performance across normalization configurations for BM25 and VSM.

Although not central to our main findings, these results underscore the need for careful normalization when integrating heterogeneous retrieval signals.

6.7 Rank Reciprocal Fusion and Re-ranking

We also evaluated two lightweight alternatives to hybrid scoring: Reciprocal Rank Fusion (RRF) and two-stage re-ranking. RRF, implemented with the standard parameter $k = 60$, merges ranked outputs without combining raw scores, prioritizing documents that consistently appear near the top of individual rank lists. Its simplicity and robustness make it a practical choice when score normalization is undesirable.

The re-ranking pipeline, in contrast, uses BM25 for initial retrieval followed by dense re-ranking using vector space model (VSM) similarity on the top k candidates. While computationally efficient and scalable, this approach showed only modest improvements over BM25 alone in our tests.

6.8 Overall Retrieval Performance

Figure 12 summarizes the average performance across all ranking strategies. As expected, the hybrid model outperformed others by combining the lexical precision of BM25 with the semantic coverage of VSM. VSM alone consistently underperformed due to its weaker handling of exact matches. Interestingly, both RRF and re-ranking offered measurable gains over their base models and may serve as effective, lower-complexity alternatives in resource-constrained settings.

6.9 Evaluation Summary and Limitations

While our findings highlight the effectiveness of hybrid retrieval, they are bounded by the size and specificity of our evaluation dataset. Manual relevance annotation limited the scale of testing, and further large-scale experiments are needed to generalize these results. Moreover, the observed similarity in performance between single and multi-vector embeddings likely stems from the short, structurally uniform nature of the documents in our corpus. Future work will explore more nuanced multi-vector strategies, such as segment-specific weighting and modality-aware representations, to better leverage document heterogeneity.

7 Conclusion

Modern industrial product catalogs pose unique challenges for semantic retrieval due to their complex structure, integrating textual descriptions, structured tables, and visual components. Existing approaches often treat these modalities in isolation, limiting retrieval performance in real-world systems. This work addresses that gap by introducing a unified pipeline for content extraction and evaluation, enabling more accurate search over richly formatted technical documents.

We developed and evaluated two complementary components: **TTI.txt**, a preprocessing pipeline that linearizes text, tables, and image annotations into a unified textual stream; and **SSE**, a modular framework for benchmarking document retrieval strategies. Together, these tools enabled us to investigate four key research questions:

RQ1: Unified Representation. We demonstrated that a plain-text format combining text, table content, and image annotations (via TTI.txt) can retain semantic completeness without the need for proprietary formats. This unified representation proved sufficient for effective downstream retrieval, especially when coupled with modular extraction and cleaning.

RQ2: Embedding Granularity. Our comparison of single-vector versus multi-vector embeddings showed no consistent recall advantage under default chunking settings. However, multi-vector methods remain promising, particularly when paired with adaptive chunk sizing, differentiated segment weighting (e.g., by document section), or future multi-modal embeddings.

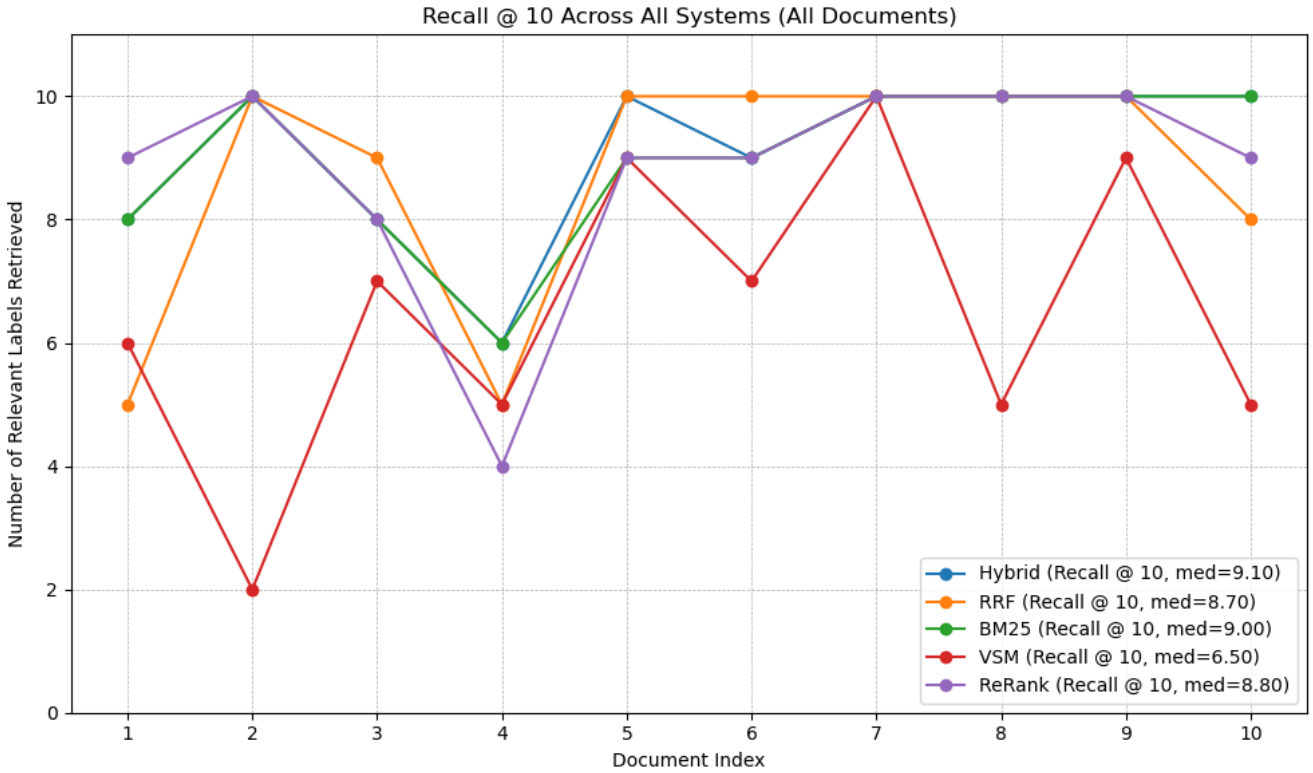

Fig.12. Mean average score comparison for all ranking methods.

These strategies offer a path toward higher retrieval precision in specification-dense documents.

RQ3: Search Strategy. Hybrid pipelines—combining BM25, dense vector similarity, rank fusion, and re-ranking—outperformed individual methods on our product catalog dataset. While hybrid systems introduce additional complexity, they consistently improved recall and precision. Practical deployment should balance this trade-off against constraints such as latency, transparency, and maintenance cost. Adaptive query routing offers one future direction to mitigate these concerns.

RQ4: Dataset Specificity. We found that the most effective retrieval strategy is highly dependent on corpus characteristics. Sparse methods performed best on terminology-rich documents, while dense embeddings handled semantically abstract queries more effectively. The SSE framework enables rapid experimentation to determine optimal configurations based on dataset structure and use-case demands.

Key Contributions

We contribute a modular preprocessing toolkit comprising two open-source components, TTI.txt and SSE, which enable plug-and-play experimentation across varied document structures and retrieval pipelines.

Through an empirical analysis of embedding granularity, we elucidate the trade-offs between computational efficiency and retrieval accuracy, offering concrete guidance on optimal chunking strategies and segment-aware weighting schemes. Our comparative benchmarking demonstrates that hybrid retrieval pipelines—combining sparse and dense methods—can improve mean average precision by up to 30 percent, albeit at increased computational cost, thereby informing design

decisions that balance performance and interpretability.

We further show how dataset characteristics—such as domain specificity, structural consistency, and vocabulary complexity—profoundly affect retrieval outcomes, underscoring the importance of corpus profiling in production environments. Finally, we outline promising future directions, including salience-based chunking, adaptive query routing, and the integration of modality-specific encoders, to inspire continued innovation in semantically aware, scalable retrieval systems.

8 Future Work

As previously mentioned, this section outlines several areas that, while beyond the scope of the current project, present valuable opportunities for future investigation. In particular, we propose enhancements across three key components of the retrieval pipeline: Query Understanding (the semantic search aspect), Document Indexing (how documents are represented and structured), and Document Retrieval (the mechanisms used to retrieve relevant documents). Future work may involve exploring alternative methods and strategies for each of these components, building upon and extending the approaches presented in this study.

8.1 Enhancing Intelligent Query Understanding

8.1.1 Corpus-Guided Semantic Models

Although not implemented in the current work, an emerging and promising direction involves the use of corpus-steered query expansion powered by Large Language Models (LLMs). Rather than relying solely on general-purpose pretrained models, this approach advocates for fine-tuning or prompting LLMs using the linguistic and structural characteristics of the target corpus. This facilitates more accurate, context-sensitive query interpretations that are grounded in the data distribution of the specific domain. While our system does not incorporate this method, its potential merits suggest an intriguing avenue for future research.

8.1.2 Semantic Parsing

Even though we initially considered implementing semantic parsing in the current project, constraints related to computational resources and time prevented us from pursuing this approach. The goal of semantic parsing is to extract relevant information from a user query in a structured format. By using a Named Entity Recognition (NER) tool, we could have conducted more precise and noise-free searches within the system. However, due to the domain-specific nature of our project, no pre-existing models were available to extract the required information. While training a custom model was a viable option, it necessitated the creation of our own labeled dataset. Given the aforementioned time constraints, we were unable to undertake this additional step.

8.2 Enhancing Document Indexing

8.2.1 Constructing Per-Document Graphs.

One promising direction involves parsing each document into a localized or micro-graph, where nodes represent entities such as materials, dimensions, and tolerances, and edges denote their interrelations. These per-document graphs can then be incrementally merged or aligned into a global knowledge graph, which serves as a substrate for GNN-based learning. This hierarchical graph construction strategy enables both document-level reasoning and corpus-wide generalization, offering a scalable method for semantic enrichment.

8.2.2 *Advanced Document Parsing for Structured Layouts.*

Technical documents, particularly PDFs, often include complex layouts with embedded tables, figures, and nested metadata. Traditional tools such as pdfplumber and PyMuPDF exhibit limitations in accurately capturing these structures. To overcome this, future iterations of the system may incorporate layout-aware deep learning models such as LayoutLM or Donut. These models integrate both visual and textual signals to extract structured data more effectively, thus improving downstream tasks such as entity recognition, relation extraction, and graph construction.

8.2.3 *Transofrmer based embeddings, DocBERT*

To avoid limitations related to fixed-length input, we initially considered various embedding methods and ultimately employed Word2Vec. BERT was not chosen due to its maximum token limit of 1024, which stems from the encoder architecture's restricted memory span. Since many of our catalogs exceed this limit, using BERT would have required truncating or excluding important information. As a result, we decided not to include it in our primary approach.

8.2.4 *Long-content BERT variants*

These methods–LongBERT, Longformer, BERT + TextRank, BERT + Random Passage, BiLSTM–aim to extend BERT's applicability to longer sequences. However, the supporting literature is still emerging, and there is no clear consensus on whether these adaptations provide significant improvements over simply applying BERT to the first 1024 tokens.

8.3 Enhancing Document Ranking: VSM, GNN, and Hybrid Approaches

An important direction for future work involves a systematic evaluation and optimization of document ranking strategies within the retrieval pipeline. While the current system primarily relies on semantic matching for initial document retrieval, significant performance improvements may be achievable through more advanced re-ranking mechanisms. Three ranking strategies merit particular attention: traditional Vector Space Model (VSM)-based ranking, Graph Neural Network (GNN)-based re-ranking, and a hybrid framework that combines both methodologies.

8.3.1 *VSM-Based Ranking.*

The current implementation employs a Vector Space Model (VSM) approach as the default ranking mechanism. This method serves as the foundation of the retrieval pipeline, providing an initial ranking of documents based on semantic similarity. All subsequent enhancements and alternative strategies are intended to build upon or re-rank the outputs generated by this baseline.

8.3.2 *GNN-Based Re-ranking.*

To capture higher-order relational information and contextual interactions among documents, a two-stage retrieval architecture can be employed. In the first stage, the top K candidate documents are retrieved using VSM similarity. In the second stage, these candidates are represented as nodes in a document graph, where edges denote semantic or structural relationships such as shared terminology, co-occurrence in domain-specific contexts, or hierarchical component-part associations.

A Graph Neural Network (GNN) is then applied to this subgraph to refine document embeddings or to generate context-aware relevance scores. This approach leverages graph-based inductive biases to model the interaction between documents, potentially improving the ranking of relevant but lexically dissimilar results.

8.3.3 Hybrid Approaches.

Combining the strengths of both VSM and GNN frameworks offers a promising hybrid strategy. One avenue involves weighted ensembling, where initial VSM scores are adjusted using GNN-derived relevance signals. Another involves iterative refinement, wherein GNN-informed rankings are used to update the query representation or guide a secondary retrieval phase.

Future research should investigate the comparative efficacy of these strategies using both intrinsic (e.g., ranking accuracy, nDCG) and extrinsic (e.g., task performance, user satisfaction) evaluation metrics. Further exploration into graph construction techniques, GNN architectures, and integration protocols is also essential for realizing the full potential of hybrid retrieval models.

8.4 Additional Proposals

Beyond core re-classification strategies, several additional proposals can enhance the semantic understanding and precision of the system's retrieval. These avenues involve improvements in ontology alignment, document representation, and advanced parsing techniques.

8.4.1 Leveraging ISO 10303/STEP Ontologies.

The ISO 10303 standard, commonly referred to as STEP, provides comprehensive ontologies for product and part modeling. While these structured classifications are valuable for representing mechanical components, their rigidity may limit their applicability in flexible or heterogeneous use cases. For instance, a component like a bolt may appear in various assembly contexts with different functional roles. Future work could explore selective or adaptive integration of STEP ontologies, potentially relaxing strict hierarchies in favor of more flexible semantic mappings.

References

[**Rocchio, 1971**] Rocchio, J. J. (1971). Relevance feedback in information retrieval.

[**Lavrenko & Croft, 2001**] Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models.

[**Mikolov et al., 2013**] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

[**Devlin et al., 2019**] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.

[**Järvelin & Kekäläinen, 2002**] Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques.

[**Robertson & Zaragoza, 2009**] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond.

[**Le & Mikolov, 2014**] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents.

[**Karpukhin et al., 2020**] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering.

[**Zhou & Devlin, 2021**] Zhou, G., & Devlin, J. (2021). Multi-vector attention models for deep re-ranking.

[**Wu, 2012**] Wu, S. (2012). Linear combination of component results in information retrieval. *Data & Knowledge Engineering*, 71(1), 114–126. `https://doi.org/10.1016/j.datak.2011.08.003`

[**Rao et al., 2025**] Rao, A., Alipour, H., & Pendar, N. (2025). Rethinking hybrid retrieval: When small embeddings and LLM re-ranking beat bigger models. *arXiv preprint arXiv:2506.01903*.

[**Cormack et al., 2009**] Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods.

[**Nogueira & Cho, 2019**] Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

[**Zhang et al., 2024**] Zhang, K., Qin, Y., Jin, J., Liu, Y., Su, R., Zhang, W., & Yu, Y. (2024). DREAM: A Dual Representation Learning Model for Multimodal Recommendation. *arXiv preprint arXiv:2404.11119*.

[**Manning et al., 2008**] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*.

[**Guo et al., 2016**] Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval.

[**Bennett et al., 2012**] Bennett, P. N., White, R. W., Chu, W., & Bennett, S. (2012). Modeling the impact of personalization on search result quality.

[**Khattab & Zaharia, 2020**] Khattab, O., & Zaharia, M. (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 (SIGIR)* (pp. 39–48). ACM.

[**Voorhees, 1994**] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations.

[**Azad & Deepak, 2021**] Azad, H. K., & Deepak, A. (2021). Query expansion techniques for information retrieval: A survey.

[**Gao et al., 2013**] Gao, J., Xu, G., & Xu, J. (2013). Query expansion using path-constrained random walks. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)* (pp. 563–572). ACM. `https://doi.org/10.1145/2484028.2484058`

[**Fang et al., 2006**] Fang, H., Campbell, M., & Thomas, P. (2006). Semantic term matching in axiomatic approaches to information retrieval.

[**Zhong et al., 2020**] Zhong, E., Dong, L., Wei, F., & Zhou, M. (2020). Evaluating neural retrieval architectures via fine-tuning.

[**Liang et al., 2016**] Liang, P., Zhu, F., & Yang, Y. (2016). Learning to segment for single-pass retrieval.

[**Salton et al., 1975**] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing.

[**Singhal, 2001**] Singhal, A. (2001). Modern information retrieval: A brief overview.

[**Zhao et al., 2024**] Zhao, W. X., Liu, J., Ren, R., & Wen, J.-R. (2024). Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4), Article 89. https://doi.org/10.1145/3637870

[**Park H, et al., 2022**] Efficient Classification of Long Documents Using Transformers

[**Su et al., 20242**] .*TableGPT2: A Large Multimodal Model with Tabular Data Integration*.