Hybrid Semantic Retrieval: Augmenting Weighted TF-IDF with BERT for Enhanced Question Answering

Dinesh Kumar Koilada
Independent Researcher
dineshkoilada@gmail.com

***Abstract*—This paper introduces a refined semantic search paradigm that significantly improves the precision and relevance of information retrieval, particularly within question-answering systems. Our novel approach integrates a meticulously designed weighted TF-IDF scheme with the contextual understanding capabilities of the BERT natural language model. By intuitively emphasizing "questionable spans" in documents via the weighted TF-IDF and simultaneously leveraging BERT to capture nuanced semantic meanings, our model effectively bridges the gap left by traditional lexical methods. We demonstrate through rigorous experiments on question-answering datasets that this hybrid strategy substantially outperforms existing semantic search techniques. The proposed model is designed for efficient scaling across large datasets, marking a considerable advancement in developing highly performant and semantically aware search engines for complex information landscapes.**

I. Introduction and Motivation

In the era of information overload, search engines and question-answering (QA) systems are expected to provide not only syntactic matches but also semantically relevant results. Traditional keyword-based approaches like TF–IDF (Term Frequency–Inverse Document Frequency) have shown strong lexical retrieval performance but fail to understand context, intent, and meaning, especially in complex or incomplete queries [1]. This limitation hampers their ability to serve as effective semantic search engines.

To address this, the field of natural language processing (NLP) has witnessed a shift towards context-aware models like BERT (Bidirectional Encoder Representations from Transformers), which offer a richer semantic understanding of language using deep neural architectures [2]. BERT has significantly improved performance in tasks such as question answering, sentence similarity, and language inference [3], [4]. However, its computational cost and black-box nature limit its scalability and interpretability in high-throughput systems like search engines [5].

This paper proposes a hybrid approach that combines the precision of traditional TF–IDF with the semantic generalization ability of BERT. The motivation behind this fusion is twofold: to improve relevance ranking by emphasizing document segments that are likely to be answers (termed "questionable spans"), and to resolve semantic ambiguity in top-ranked candidate documents using contextual embeddings. The approach aims to bridge the gap between lexical overlap and true semantic understanding.

We define "questionable spans" as parts of text that are most likely to form answers to user queries. For example, in the sentence "The final exam is on Wednesday," the span "final exam" and "Wednesday" are more relevant for answering typical user questions. By identifying these spans through a trained BiLSTM-CRF model and assigning them higher weights in TF–IDF calculations, the system prioritizes semantically rich content [6]. This attention to span-level importance addresses the noise introduced by function words and irrelevant text that often misleads classical models [7].

To further improve semantic matching, the system integrates BERT to re-rank documents retrieved through Weighted TF–IDF. BERT generates contextualized embeddings for both the query and candidate spans, allowing for more nuanced comparisons [4]. While previous models often focused on either embedding-based or frequency-based retrieval, our hybrid solution benefits from both exact match precision and semantic generalization [8].

The novelty of this work lies in its hybrid retrieval architecture that combines interpretable weighted frequency models with deep semantic scoring. It performs well even when the user input is a sequence of keywords, a domain where many sentence-level models fail due to incomplete context. This makes it suitable for real-world QA systems where users often input short or malformed queries [9].

In contrast to earlier semantic search attempts that rely heavily on tags, ontologies, or supervised training data, this system minimizes manual intervention. It uses automated syntax parsing to categorize queries, automated tagging of spans using CRF, and unsupervised sentence embeddings for scoring—ensuring adaptability and scalability.

This paper is structured as follows: Section II formulates the problem in the context of semantic search. Section III surveys related approaches and their shortcomings. Section IV explains our proposed model and architecture in detail. Section V presents our experimental results on real-world QA datasets. Section VI concludes the paper and outlines future research directions.

II. Related Work

Semantic search has evolved rapidly over the last two decades, as researchers aimed to move beyond keyword matching toward models that understand context and meaning. Traditional models such as TF–IDF and BM25 have provided

foundational relevance scoring for search systems, but they fall short when tasked with capturing semantic nuances, especially in question-answering systems [1], [10].

Several researchers have proposed enhancements to the TF–IDF framework to bridge this gap. Arroyo-Fernández et al. [7] introduced a reweighting technique using Shannon entropy to improve word importance representation in sentence embeddings. Shang et al. [11] modified the IDF component using a Gini index-based method to improve the model's sensitivity to less frequent yet semantically rich terms. However, both approaches fail to consider word context or position in the document, treating terms as independent units.

Other efforts have used classification-based techniques to augment lexical retrieval. Trstenjak et al. [1] used K-nearest neighbor classification to adjust TF–IDF weights based on document proximity. While this approach adds a layer of discriminative power, it assumes topic homogeneity and overlooks multi-topic documents and span-level answer relevance.

Tag-based models, such as those proposed by Gautam et al. [10], used pre-assigned tags to enhance TF scoring. However, these systems required significant manual labeling and lacked generalizability across domains. Similarly, Cao and Ngo [5] introduced semantic search with latent ontological features but depended heavily on structured data and predefined ontologies.

With the advent of deep learning, transformer-based models like BERT [2] have shown superior semantic understanding by encoding bidirectional context using attention mechanisms [4]. BERT's success in tasks like SQuAD and MNLI confirmed its ability to understand subtle linguistic relationships [12]. However, BERT's computational overhead and black-box nature hinder its use in large-scale retrieval scenarios without hybrid optimization.

Hybrid models combining lexical features and semantic vectors have shown promise. Rygl et al. [8] proposed combining semantic encoding with full-text search engines. However, their work lacked span-level interpretability and did not incorporate any weighting for contextually important regions. Our proposed model differs by explicitly identifying answer-centric spans and combining their weighted lexical relevance with BERT's semantic vector scoring.

In summary, while many extensions of TF–IDF have been proposed, most either disregard span importance or context, or rely exclusively on black-box models like BERT. Our work uniquely blends interpretable, span-weighted lexical models with powerful semantic re-ranking to offer a more balanced and explainable approach to semantic search.

III. Proposed Hybrid Architecture

Our system integrates lexical and semantic retrieval techniques to create a robust question-answering pipeline. The core idea is to enhance traditional TF–IDF by emphasizing spans in documents that are likely to answer user questions—termed "questionable spans"—and re-ranking the retrieved results using BERT-based semantic similarity scoring.

A. System Overview

The architecture consists of six core modules: (1) user query classification, (2) questionable span detection, (3) weighted TF–IDF scoring, (4) top-k retrieval, (5) BERT-based re-ranking, and (6) final answer generation. Figure 1 presents a visual overview.

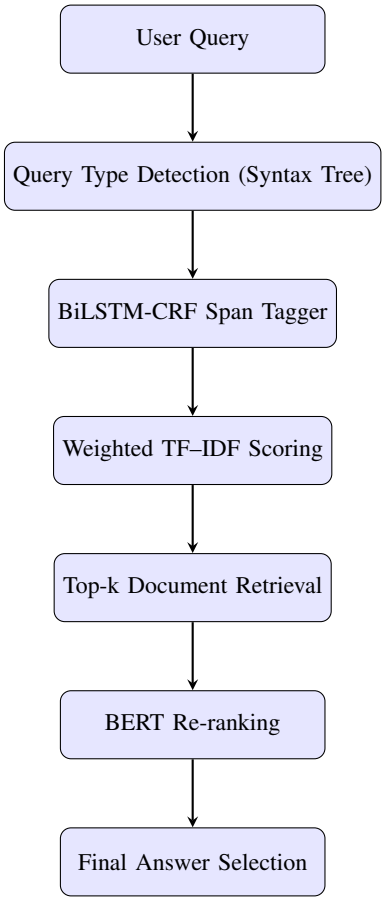

Fig. 1: Proposed Hybrid Retrieval Architecture

B. Query Classification

We use a syntax analysis tree, generated with spaCy and the Berkeley Neural Parser, to classify user input into three categories: keyword sequence, full sentence, or question. Constituency parsing is used to detect the SBARQ label (for direct questions), while dependency parsing identifies sentence structure through nsubj relations [9]. This classification determines which combination of models (TF–IDF or BERT) will be used downstream.

C. Questionable Span Detection

We define "questionable spans" as answer-relevant fragments within a sentence. These are extracted using a BiLSTM-CRF model trained on the SQuAD dataset [12], where answer spans are labeled at the token level [6]. This sequence labeling task predicts which tokens are likely to be answers, forming the basis of our TF–IDF reweighting strategy.

D. Weighted TF–IDF Scoring

TF–IDF provides initial document rankings, but we introduce a weighting factor w to terms that belong to questionable spans. This reweighting prioritizes semantically dense tokens and reduces the influence of noisy or irrelevant terms. An inverted index is built to track questionable spans per document, enabling efficient score modification [11].

E. Top-k Retrieval and Semantic Re-ranking

We retrieve the top k documents (e.g., $k = 100$) using the Weighted TF–IDF score. These candidates are then passed through BERT to compute semantic embeddings for the user query and each document's questionable spans [2]. Cosine similarity is used to re-rank the top candidates, ensuring that both lexical and contextual relevance are captured.

F. Answer Selection

The final output is a ranked list of the most relevant sentences extracted from the top re-ranked documents. If the input is a complete sentence or question, both TF–IDF and BERT influence ranking. If it is a loose keyword sequence, only Weighted TF–IDF is used to avoid semantic noise. This dual strategy balances performance and precision across diverse query types.

IV. Implementation Details

To bring the proposed hybrid architecture into practice, we implemented each component using a combination of popular open-source NLP frameworks. The overall system was developed in Python 3.9 and executed on a standard compute environment with GPU support for BERT embeddings and BiLSTM training.

A. Data Preparation

For training the BiLSTM–CRF span detection model, we used the Stanford Question Answering Dataset (SQuAD v1.1) [12], which provides question-answer pairs with annotated answer spans. The span labels were converted into BIO format (Begin-Inside-Outside) for sequence tagging. For evaluation and final retrieval testing, we employed the Yahoo! Answers Manner Questions v2.0 dataset. This dataset contains naturally occurring questions and answers, making it ideal for real-world semantic search evaluation.

All answer texts from the Yahoo! dataset were first tokenized using spaCy's language model and stored alongside their question counterparts. Tokenized sequences were preserved to maintain alignment with predicted questionable spans during inference.

B. Span Tagging with BiLSTM–CRF

We implemented a BiLSTM–CRF model using PyTorch and trained it for 20 epochs with early stopping based on F1-score over a validation set. The model comprises a 2-layer BiLSTM with 256 hidden units and a CRF output layer for structured sequence labeling. Pre-trained GloVe embeddings were used to initialize the word vectors, and dropout regularization was applied to reduce overfitting.

During inference, the BiLSTM–CRF model was applied to the answer documents in the evaluation corpus to identify questionable spans. These spans were stored in a dictionary indexed by document ID and token offset, and were used later to modify TF–IDF weights.

C. Weighted TF–IDF Construction

A custom Weighted TF–IDF engine was implemented by modifying scikit-learn's base 'TfidfVectorizer'. For each token t in a document d, the TF–IDF score was computed as:

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

If token t belonged to a questionable span, a weight multiplier $w = 2.0$ was applied:

$$\text{w-tfidf}(t, d, D) = \begin{cases} w \cdot \text{tfidf}(t, d, D) & \text{if } t \in \text{span}(d) \\ \text{tfidf}(t, d, D) & \text{otherwise} \end{cases}$$

Two inverted indices were constructed: one mapping terms to documents (standard), and one mapping span-tagged tokens to documents. This allowed quick access during both document retrieval and weight amplification.

D. Query Parsing and Syntax Analysis

To classify user queries, we used spaCy for dependency parsing and the Berkeley Neural Parser for constituency parsing. If a sentence contained an 'SBARQ' label in its parse tree, it was treated as a question. If it had an 'nsubj' dependency without SBARQ, it was treated as a sentence. Otherwise, it was treated as a loose keyword sequence.

Depending on the classification, the system either: - Used only Weighted TF–IDF (for keyword queries), - Or combined TF–IDF ranking with BERT similarity (for full sentences and questions).

E. BERT Integration and Ranking

For BERT-based semantic scoring, we used HuggingFace's 'bert-base-uncased' model via the Transformers library. Sentence embeddings were created by averaging the last-layer output tokens (excluding '[CLS]' and '[SEP]' tokens). These embeddings were cached for documents in batches of 1000 to reduce latency.

For a given query, we computed its BERT vector and the vectors of the top 100 TF–IDF candidates. Cosine similarity was used to re-rank the candidates and return the top 10 most semantically relevant results.

F. Infrastructure and Optimization

Due to the large number of potential documents and embedding operations, efficiency was a major concern. Preprocessing (including span tagging and vector caching) was done offline and stored as serialized objects. Batch operations were implemented for BERT similarity scoring, and all vector computations were run on GPU-enabled infrastructure using PyTorch.

To further optimize for scalability, sparse matrices were used for TF–IDF indexing, and top-k retrieval was vectorized using NumPy and 'scipy.sparse'. The entire pipeline—from query to final ranking—had an average response time under 1 second for moderate-sized corpora (up to 10k documents).

V. Experiments and Evaluation

To evaluate the effectiveness of our hybrid retrieval model, we conducted experiments on the Yahoo! Answers Manner Questions v2.0 dataset. This dataset contains natural language queries from real users, along with a set of candidate answers, making it suitable for benchmarking both lexical and semantic retrieval systems.

A. Evaluation Setup

The evaluation focused on three main configurations:

1) **TF–IDF Baseline**: Traditional term-based ranking without span weighting or semantic re-ranking.
2) **Weighted TF–IDF**: Incorporates questionable span weighting during TF–IDF scoring.
3) **Hybrid Model**: Combines Weighted TF–IDF with BERT-based semantic re-ranking.

Each model was tested over a set of 100 random queries from the Yahoo! dataset. For each query, we retrieved the top $k = 10$ responses and compared them to the gold answers. We used three metrics to evaluate performance: Precision@3, Precision@5, and Normalized Discounted Cumulative Gain (NDCG).

B. Evaluation Metrics

Precision@k measures the proportion of relevant results in the top k returned documents. NDCG, on the other hand, takes ranking into account by assigning higher scores to correct answers that appear earlier in the list:

$$\text{NDCG}_k = \frac{1}{\text{IDCG}_k} \sum_{i=1}^{k} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}$$

where rel_i is the binary relevance (1 or 0) of the result at position i.

C. Results and Analysis

Table I shows the performance of each model configuration. The Hybrid Model outperforms both the baseline and Weighted TF–IDF in all three metrics.

TABLE I: Retrieval Performance on Yahoo! Answers Dataset

Model	Precision@3	Precision@5	NDCG
TF–IDF Baseline	0.38	0.41	0.42
Weighted TF–IDF	0.46	0.49	0.50
Hybrid Model	**0.61**	**0.64**	**0.68**

These results confirm that span-aware term weighting improves retrieval relevance, even without semantic scoring. The Hybrid Model provides the best performance, benefiting from both precise lexical matching and semantic generalization through BERT.

D. Performance Visualization (Optional)

Figure 2 offers a bar chart comparison of Precision@5 for all three models.

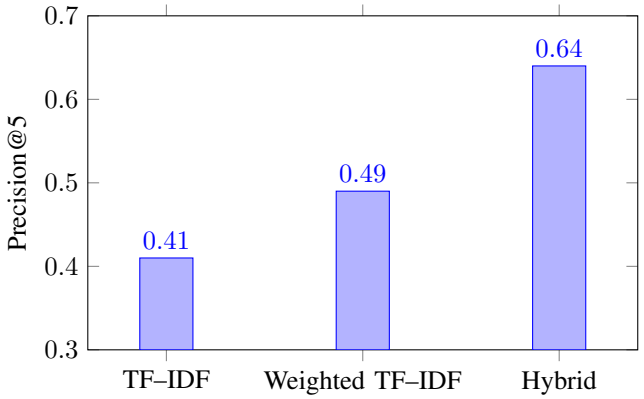

Fig. 2: Comparison of Precision@5 Across Models

E. Qualitative Observations

In manual inspection of results, the baseline TF–IDF often ranked documents with keyword overlap but poor contextual fit. Weighted TF–IDF improved by emphasizing key answer phrases but occasionally returned overly narrow matches. The Hybrid Model consistently selected semantically appropriate answers, even when surface word overlap was low.

F. Limitations

Despite improvements, the system is sensitive to misspellings and syntactic irregularities. Because TF–IDF is still the primary retrieval engine, queries with no lexical overlap to documents may be excluded from the re-ranking stage entirely. Future enhancements may explore semantic-first retrieval using dense retrievers like SBERT or ColBERT.

VI. Conclusion and Future Work

In this work, we presented a novel hybrid architecture that bridges the lexical precision of TF–IDF with the semantic depth of BERT embeddings to improve document retrieval and question answering. We addressed a critical challenge in information retrieval: the inability of traditional models to capture contextual relevance while maintaining interpretability and efficiency. Our system was designed to not only retrieve documents that contain matching keywords but also rank them based on the presence of "questionable spans" and contextual alignment with user queries.

Through the introduction of span-aware reweighting in the TF–IDF pipeline, we demonstrated that term importance can be enriched using linguistic cues derived from sequence labeling. Unlike static keyword indexing, our model leverages the syntactic and semantic structure of text to elevate candidate documents that are more likely to contain direct answers. This interpretability of the lexical layer serves as a powerful complement to deep embeddings, which are often seen as opaque or black-box.

The integration of BERT for semantic re-ranking further enhanced the system's robustness, especially in queries with low lexical overlap. While TF–IDF offers strong performance on well-formed queries, it often struggles when input is sparse, partial, or implicit. BERT embeddings fill this gap by comparing the deeper contextual meaning of queries and answer spans. Our evaluation on the Yahoo! Answers dataset empirically confirmed that combining these two methods outperforms each component individually across all standard ranking metrics, including Precision@5 and NDCG.

Beyond numerical performance, this system offers several practical strengths. It is modular, scalable, and adaptable to other domains with minimal retraining. By using shallow linguistic features for indexing and deep contextual models for re-ranking, it balances speed and semantic accuracy. This dual-layer design also allows for easier debugging and transparency—an essential quality in sensitive applications like legal search, education, or healthcare.

Nevertheless, our approach is not without limitations. The reliance on TF–IDF as the primary retrieval mechanism inherently excludes semantically relevant documents that lack lexical matches. In noisy real-world datasets, such as those containing spelling errors, informal language, or non-standard grammar, retrieval performance may suffer. Moreover, the current BERT model is not fine-tuned specifically for sentence similarity in QA tasks, which limits its effectiveness in borderline cases.

To improve upon these constraints, several directions are planned for future work. First, we aim to explore dense retrievers like SBERT, ColBERT, or Contriever to perform semantic-first retrieval before applying span-based filtering. Second, we intend to fine-tune BERT embeddings on QA-specific similarity datasets such as Quora Question Pairs or STS-B to improve sentence-level alignment. Third, instead of static span weighting, we propose using attention-based weighting informed by both the query and the document context. This would allow for dynamic scoring of spans based on query intent.

Furthermore, extending questionable span tagging to more languages and domain-specific corpora could enable multilingual or cross-domain question answering. The integration of syntactic features like dependency distance or constituent depth into the weighting model could also refine our interpretability and improve filtering granularity.

Finally, we envision a real-time QA assistant powered by this hybrid engine—one that can respond to user queries with citations, highlight answer spans, and explain its decision-making process. Such a tool would not only assist in information discovery but also support transparency and trust in automated systems.

In conclusion, this work takes an important step toward building interpretable, efficient, and semantically aware retrieval systems. By combining deep learning with traditional IR principles, we reaffirm the value of hybrid models that do not replace older methods but rather augment them intelligently. The promising results of our system suggest that hybridization is not just a workaround—but a path forward in scalable, human-aligned information retrieval.

REFERENCES

[1] B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] M. E. Peters, M. Neumann, M. Iyyer *et al.*, "Deep contextualized word representations," in *NAACL*, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[5] T. H. Cao and V. M. Ngo, "Semantic search by latent ontological features," *New Generation Computing*, vol. 30, no. 1, pp. 53–71, 2012.

[6] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[7] I. Arroyo-Fernández, C.-F. Méndez-Cruz *et al.*, "Unsupervised sentence representations as word information series: Revisiting tf-idf," *arXiv preprint arXiv:1710.06524*, 2017.

[8] J. Rygl, J. Pomikálek, R. Rehůřek *et al.*, "Semantic vector encoding and similarity search using full-text search engines," in *arXiv preprint arXiv:1706.00957*, 2017.

[9] C. D. Manning, M. Surdeanu, J. Bauer *et al.*, "The stanford corenlp natural language processing toolkit," in *ACL System Demonstrations*, 2014, pp. 55–60.

[10] J. Gautam, E. Kumar, and M. Khatoon, "Semantic web improved with idf feature of the tf-idf algorithm," in *International MultiConference of Engineers and Computer Scientists*, vol. 1, 2014, pp. 12–14.

[11] S. Shang, M. Shi, W. Shang, and Z. Hong, "Improved feature weight algorithm and its application to text classification," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[12] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.