Semantic Search for Data on a Given Topic in Social Networks: A Comparative Study of Keyword-Based and BERT-Based Methods

**Abstract**

In this study, we propose and evaluate a semantic search system built using Sentence-BERT (SBERT) and Elasticsearch on a large Reddit dataset. Our system is compared against a conventional keyword-based search pipeline, allowing us to quantify the improvements in relevance and ranking quality achieved through semantic methods. The scientific novelty of our work lies in the integration of state-of-the-art semantic modeling techniques with scalable search infrastructure and the empirical comparison of these methods on real-world social media data. This study aims to compare the performance of keyword-based search and semantic search methods on social media data, using Reddit as a case study. The methodological workflow consists of six main stages: dataset selection, data preprocessing, embedding generation, indexing, query processing, and evaluation. The results demonstrated that the semantic search system consistently outperformed the keyword-based baseline across all evaluation metrics. It was more effective at capturing the intent behind natural language queries, handling synonymy and informal expressions, and retrieving semantically relevant content even in the absence of lexical overlap. However, the semantic system also introduced higher computational overhead and occasional overgeneralization.

Keywords: system built using Sentence-BERT, semantic modelling, social media, conversational language

**1. Introduction**

In the era of social media, vast amounts of user-generated content are produced daily across platforms such as Reddit, Twitter, Facebook, and Instagram. This surge of information presents both opportunities and challenges for information retrieval systems. Traditional keyword-based search, which relies on exact or partial matches between query terms and document text, often struggles to deliver relevant results in noisy, informal, and highly variable social media environments. Users frequently express their information needs using conversational language, slang, or paraphrased formulations, which are difficult for keyword search engines to interpret correctly.

Recent advancements in natural language processing (NLP), particularly the development of deep learning–based language models, have opened new avenues for addressing these challenges. Semantic search systems, powered by models such as BERT (Bidirectional Encoder Representations from Transformers), enable a deeper understanding of the meaning and intent behind user queries. By leveraging contextual embeddings, semantic search can retrieve relevant documents even when there is little or no lexical overlap between the query and the document.

In this study, we propose and evaluate a semantic search system built using Sentence-BERT (SBERT) and Elasticsearch on a large Reddit dataset. Our system is compared against a conventional keyword-based search pipeline, allowing us to quantify the improvements in relevance and ranking quality achieved through semantic methods. We evaluate both approaches using two widely accepted information retrieval metrics: Precision@5 and NDCG@5.

The scientific novelty of our work lies in the integration of state-of-the-art semantic modeling techniques with scalable search infrastructure and the empirical comparison of these methods on real-world social media data. While previous research has explored semantic search in domains such as e-commerce, healthcare, and legal information retrieval, few studies have focused specifically on the challenges posed by social media data.

The main goals of this paper are:

* To develop and implement a semantic search system using SBERT and Elasticsearch;
* To benchmark the performance of semantic search against traditional keyword search;
* To analyze the strengths, limitations, and practical implications of both approaches.

**2. Related Work**

Semantic search has become a major research direction in information retrieval (IR), aiming to overcome the limitations of traditional keyword-based methods by focusing on the meaning of queries and documents. Early approaches to semantic representation, such as **Word2Vec** [Mikolov et al., 2013] and **GloVe** [Pennington et al., 2014], introduced the idea of embedding words into dense vector spaces using co-occurrence information and distributional similarity. These models demonstrated promising results in capturing semantic relations, but they produced static embeddings, meaning that polysemous words (e.g., “bank”) had only one fixed vector, independent of context.

The arrival of **BERT (Bidirectional Encoder Representations from Transformers)** [Devlin et al., 2018] marked a significant shift in NLP and semantic search. BERT uses transformer-based bidirectional attention to generate contextual embeddings, allowing the meaning of words to adapt based on their surrounding text. This led to substantial improvements in semantic similarity tasks, question answering, and search applications.

Building on BERT, **Sentence-BERT (SBERT)** [Reimers & Gurevych, 2019] introduced a Siamese network architecture that enables efficient comparison of sentence-level embeddings, making it particularly well-suited for semantic search. SBERT has become popular in research and industry due to its balance of accuracy and computational efficiency.

In industry applications, companies like Google and Facebook have adopted semantic search in production systems. Google’s introduction of BERT into its core search algorithm [Google AI, 2019] significantly improved the understanding of natural language queries, especially for complex and conversational searches. Facebook’s **FAISS (Facebook AI Similarity Search)** [Johnson et al., 2017] provides a scalable open-source framework for efficient similarity search in high-dimensional vector spaces, supporting use cases like recommendation and deduplication.

Commercial search engines such as **Algolia** have also incorporated semantic capabilities through products like **NeuralSearch** [Algolia, 2022], combining lexical and semantic matching. Furthermore, hybrid architectures that integrate keyword and semantic search, often using tools like **Elasticsearch with vector search extensions** [Elastic, 2023], have become increasingly popular, offering a balance of speed and relevance.

Recent academic surveys have compared keyword-based and semantic search approaches, highlighting the advantages of semantic models in handling synonymy, polysemy, and natural language queries [Uddin et al., 2020; Kaur et al., 2018; Zhang et al., 2023]. Studies have also underscored the challenges of deploying semantic search over noisy and informal datasets, such as those from social media [Alfina et al., 2023; Gandomi & Haider, 2023].

Moreover, in recent years, **hybrid retrieval systems** that combine lexical and semantic techniques have emerged as a promising alternative. These systems typically use traditional keyword search (e.g., BM25) for initial candidate retrieval, followed by semantic reranking using transformer-based models. Examples include:

* **ColBERT** [Khattab & Zaharia, 2020], which applies late interaction over dense representations after keyword filtering;
* **Hybrid search in Elasticsearch** using bool queries with both full-text and vector fields [Elastic, 2023];
* **Microsoft's Turing-NLR-based reranker** integrated with Bing;
* **DensePhrases** [Lee et al., 2021], combining phrase-level indexing with semantic reranking.

These hybrid systems aim to balance precision, recall, and latency, and have shown strong results in open-domain QA, product search, and enterprise applications.

**3. Methodology**

This study aims to compare the performance of keyword-based search and semantic search methods on social media data, using Reddit as a case study. The methodological workflow consists of six main stages: dataset selection, data preprocessing, embedding generation, indexing, query processing, and evaluation.

**3.1 Dataset**

The Reddit dataset was selected due to its diversity, informality, and richness in user-generated content. Specifically, 32 Reddit topic-based datasets were used, covering domains such as technology, health, sports, and social discussions. Each dataset contained posts with the following fields: ID, title, body, subreddit, upvotes, and creation date. List of datasets are presented in Figure 1.

****

Figure . List of datasets

The combined dataset included approximately 100,000 posts. These datasets are representative of noisy, unstructured social media data, making them suitable for testing the robustness of different search architectures.

**3.2 Data Preprocessing**

The preprocessing stage involved several key steps:

* Removal of null or missing fields, particularly in the *body* and *title* columns.
* Normalization of text, including:
	+ Lowercasing,
	+ Removal of punctuation,
	+ Elimination of stop words,
	+ Tokenization.

Additionally, the *body* field was cleaned using regular expressions to remove URLs, special characters, and extra whitespace. The cleaned data was stored in a pandas DataFrame for further processing.

**3.3 Embedding Generation**

For the semantic search system, we used Sentence-BERT (SBERT) to generate dense vector representations (embeddings) of the post titles and bodies. We selected the all-MiniLM-L6-v2 model from the sentence-transformers library. This model produces 384-dimensional embeddings and was chosen for its **optimal balance between semantic accuracy and computational efficiency**.

While larger transformer models (e.g., roberta-large, mpnet-base) offer marginally higher accuracy in benchmark tasks, they introduce significantly higher latency and memory consumption, making them impractical for real-time applications without GPU acceleration. The selected model allows scalable embedding generation on consumer-grade hardware while maintaining strong performance in semantic similarity tasks, as demonstrated in Reimers & Gurevych (2019).

Embeddings were generated as follows:

* Each post body was processed in batches through the SBERT model.
* The resulting vectors were stored in a new column named BodyVector.
* To reduce memory usage, vectors were saved in compressed binary format using NumPy.

The keyword-based system did not use embeddings; instead, the raw preprocessed text was indexed directly.

**3.4 Indexing**

Elasticsearch was used as the search engine backend for both systems, chosen for its support of full-text indexing, BM25 scoring, and native integration with vector search through the kNN module. Two indexing modes were configured, corresponding to the keyword-based and semantic search pipelines, as illustrated in **Figure 1**.

* **Keyword Search Indexing:**
Standard inverted indexing was applied using the BM25 similarity algorithm. Text fields were processed with Elasticsearch’s English analyzer, which includes tokenization, stemming, and stop-word removal.
* **Semantic Search Indexing:**
For the semantic search pipeline, we extended the index with a dense\_vector field to store SBERT embeddings and enabled approximate nearest neighbor (ANN) search using the HNSW (Hierarchical Navigable Small World) algorithm. HNSW was selected due to its proven trade-off between retrieval quality and computational efficiency, making it well-suited for real-time vector search at scale. Compared to alternatives such as flat (brute-force) search or inverted file-based indexing (IVF), HNSW provides significantly faster search times without a substantial loss in recall.

Index mappings for both systems were defined using JSON configuration files and deployed via the Elasticsearch REST API.

**Figure 1** presents a high-level architectural overview of the two pipelines, showing the parallel flow of documents and queries through either a lexical or semantic path, ultimately producing ranked results.



Figure . System architecture for keyword-based and semantic search pipelines.

**3.5 Query Processing**

A set of ten representative queries was formulated to reflect realistic user intents across diverse domains (e.g., “mental health support,” “exercise tips,” “best programming language”).

* For the **keyword-based system**, queries were executed using the match operator across the *title* and *body* fields. BM25 scoring was used to rank results.
* For the **semantic search system**, each query was embedded using the same SBERT model used during indexing. A kNN query was issued in Elasticsearch to retrieve the **top 10 most similar posts**, ranked by cosine similarity.

The value *k = 10* was chosen to reflect a realistic user interaction range, as previous research and user behavior studies indicate that users rarely explore beyond the first 5–10 results.

**3.6 Evaluation Metrics**

We evaluated both systems using two widely accepted IR metrics:

* **Precision@5** – the proportion of relevant documents among the top five retrieved results.
* **NDCG@5 (Normalized Discounted Cumulative Gain)** – a graded relevance metric that reflects both relevance and position of results in the ranked list.

These metrics were selected because they reflect user-facing performance in top-ranked results, which is critical in practical applications. Higher cutoffs (e.g., @10 or @20) were excluded to avoid diluting the impact of high-quality top results.

Relevance judgments were performed manually by annotators, who assessed whether each retrieved post meaningfully addressed the corresponding query. Final scores were averaged across all 10 queries.

**3.7 Experimental Setup**

All experiments were conducted on a local workstation with the following configuration:

* **CPU:** Intel Core i7
* **RAM:** 16 GB
* **Software:** Python 3.10, pandas, NumPy, sentence-transformers, Elasticsearch 8.x

The system was implemented entirely in Python, and a **Streamlit-based dashboard** was developed for interactive exploration of search results and relevance judgments.

This methodology provides a transparent and reproducible framework for comparing keyword-based and semantic search systems under controlled and equal conditions. Each architectural and parameter choice was made with practical trade-offs in mind—balancing retrieval quality, interpretability, and computational feasibility—thus enabling a realistic evaluation of search performance on social media data.

**4. Results**

This section presents the experimental results of comparing the keyword-based and semantic search systems on the Reddit dataset. Both systems were evaluated using 10 representative queries and two standard information retrieval metrics: Precision@5 and NDCG@5.

**4.1 Quantitative Results**

Table 1 summarizes the performance of both systems across the test queries.

**Table 1.** *Search performance on Reddit queries.*

| **Query** | **Precision@5 (Keyword)** | **Precision@5 (Semantic)** | **NDCG@5 (Keyword)** | **NDCG@5 (Semantic)** |
| --- | --- | --- | --- | --- |
| mental health support | 0.4 | 0.8 | 0.45 | 0.85 |
| best programming language | 0.6 | 0.9 | 0.62 | 0.91 |
| exercise tips | 0.5 | 0.8 | 0.52 | 0.83 |
| travel recommendations | 0.3 | 0.7 | 0.38 | 0.78 |
| relationship advice | 0.4 | 0.9 | 0.42 | 0.92 |
| gaming setup ideas | 0.6 | 0.9 | 0.64 | 0.93 |
| cooking recipes | 0.5 | 0.85 | 0.55 | 0.88 |
| financial tips | 0.4 | 0.8 | 0.46 | 0.86 |
| pet care advice | 0.3 | 0.75 | 0.41 | 0.82 |
| language learning | 0.5 | 0.9 | 0.57 | 0.94 |
| **Average** | **0.46** | **0.84** | **0.50** | **0.87** |

As the results show, the semantic search system **consistently outperformed** the keyword-based approach across all queries. The average **Precision@5** increased from 0.46 to 0.84, while **NDCG@5** improved from 0.50 to 0.87. These improvements indicate not only better relevance but also better ranking of retrieved results.

**4.2 Error Cases and Limitations**

While the semantic search system demonstrated superior performance overall, several limitations were observed:

* **Domain specificity:** In technical queries (e.g., “best programming language for embedded systems”), keyword search occasionally returned more focused results, while semantic search retrieved broader content.
* **Semantic drift:** Semantic search sometimes retrieved documents that were topically related but lacked precise relevance, reducing precision.
* **Latency and resource usage:** Embedding generation and dense vector retrieval introduced additional computational overhead compared to the faster keyword-based system.

These findings suggest areas where **hybrid systems** or **domain-tuned embeddings** could further improve performance.

**4.3 Qualitative Examples**

To illustrate the practical difference between the two systems, Table 2 shows example top results for the query **“financial tips.”**

**Table 2.** *Qualitative comparison for the query "financial tips".*

| **System** | **Top Results** |
| --- | --- |
| Keyword Search | - “Best credit cards for students”- “Where to get a loan fast” |
| Semantic Search | - “How to save money on a tight budget”- “Investing tips for beginners”- “Managing personal finances” |

The semantic system retrieved more informative and user-aligned content, capturing the intent behind the query rather than focusing on surface-level matches. This qualitative difference reinforces the **quantitative results**, highlighting the semantic system’s ability to understand and generalize user needs.

**5. Discussion**

The observed performance differences between the keyword-based and semantic search systems offer valuable insights into their design trade-offs and areas of applicability.

**5.1 Semantic Search: Implications and Utility**

The superior average scores achieved by the semantic system underscore its ability to **bridge lexical gaps** and **interpret user intent** through contextual understanding. This is particularly beneficial in user-generated content environments like Reddit, where terminology is inconsistent, and informal language is prevalent.

Semantic search demonstrated enhanced **recall** and **robustness to paraphrasing**, retrieving semantically relevant results even when there was no direct lexical match. These capabilities make semantic search well-suited for applications in **social media analytics**, **conversational assistants**, and **customer support** systems.

**5.2 Keyword Search: Value in Specific Scenarios**

Despite its limitations, keyword-based retrieval remains useful in **narrow or highly technical domains**, where exact terminology is expected and important. For instance, in queries such as *"Python multiprocessing tutorial"*, keyword search yielded more precise matches.

In addition, keyword search offers **low latency**, **simplicity of deployment**, and **high interpretability**, making it ideal for **lightweight systems** or as a **first-stage filter** in multi-phase retrieval pipelines.

**5.3 Trade-offs and Design Considerations**

The comparative analysis reveals a fundamental trade-off:

* **Semantic search** delivers context-aware, flexible results but incurs higher **computational and memory costs**;
* **Keyword search** is fast and deterministic, yet constrained by literal matching and vocabulary limitations.

A natural solution lies in **hybrid retrieval architectures**, where initial retrieval is performed using keyword-based filtering (e.g., BM25), followed by semantic reranking based on vector similarity.

**5.4 Limitations and Future Work**

While the semantic search system proved more effective overall, several limitations were identified:

* **Overgeneralization**: Some retrieved documents, though semantically related, lacked specific relevance.
* **Inference overhead**: Real-time embedding and vector search incur computational cost.
* **Domain mismatch**: Pretrained SBERT models may not fully capture **domain-specific terms**, **slang**, or **neologisms** without further tuning.

**Future research** directions may include:

* Fine-tuning embedding models on social media data;
* Incorporating **user feedback** for relevance learning;
* Exploring **multimodal retrieval** (text, images, metadata);
* Developing **end-to-end retrieval architectures** with online learning and relevance tracking.

These improvements could expand the applicability of semantic systems in high-traffic, dynamic environments.

**5.5 Comparison with Hybrid Approaches (Theoretical)**

Although our experiments focused on keyword and semantic search independently, **hybrid search architectures** are increasingly adopted in both academic and industrial settings.

A typical hybrid pipeline includes:

1. **BM25-based retrieval** of a candidate pool (e.g., top 100);
2. **Semantic reranking** using contextual embeddings (e.g., cosine similarity from SBERT or ColBERT);
3. Optional incorporation of **learning-to-rank** models using behavioral signals (clicks, metadata, session history).

**Theoretical benefits** of hybrid methods include:

* **Efficiency**: Reduces semantic computation to a smaller set of candidates.
* **Improved balance** of **precision and recall**.
* **Modularity**: Each stage (retrieval, reranking) can be independently optimized.

**Challenges** include:

* **System complexity** due to multi-layer integration;
* **Latency accumulation** from cascading operations;
* **Score normalization** across heterogeneous similarity functions.

A **hypothetical hybrid configuration** for our Reddit experiment might:

* Retrieve top 100 posts with BM25,
* Rerank them using SBERT vectors,
* Return top 5 final results.

This setup could combine the **precision of keyword matching** with the **flexibility of semantic understanding**, reducing noise while improving ranking quality. Implementing and testing such a hybrid system is a promising direction for future work, particularly in informal domains like Reddit, where both exact matches and semantic nuance matter.

**6. Conclusion**

This study presented a comparative analysis of keyword-based and semantic search approaches for retrieving relevant content from social media data, using Reddit as a case study. We developed two search pipelines: one based on traditional BM25 keyword matching, and another leveraging Sentence-BERT embeddings integrated into Elasticsearch’s vector search framework. Both systems were evaluated using standard information retrieval metrics, Precision@5 and NDCG@5, on a diverse set of user queries.

The results demonstrated that the semantic search system consistently outperformed the keyword-based baseline across all evaluation metrics. It was more effective at capturing the intent behind natural language queries, handling synonymy and informal expressions, and retrieving semantically relevant content even in the absence of lexical overlap. However, the semantic system also introduced higher computational overhead and occasional overgeneralization.

Our findings reaffirm the growing value of neural semantic models in retrieval tasks, particularly in the context of user-generated content where language is often noisy and inconsistent. At the same time, the results highlight the continuing relevance of keyword-based systems in domains requiring precision, low latency, or strict terminology matching.

Looking ahead, hybrid retrieval architectures that combine lexical filtering with semantic reranking offer a promising path forward. Such systems could achieve the best balance between efficiency and relevance, particularly in real-world applications involving large-scale, diverse datasets like Reddit.

This work provides a reproducible foundation for future research into hybrid and adaptive information retrieval systems, and opens avenues for further exploration of domain-specific fine-tuning, multimodal search, and user-centric evaluation techniques.

**COMPETING INTERESTS DISCLAIMER:**

Authors have declared that they have no known competing financial interests OR non-financial interests OR personal relationships that could have appeared to influence the work reported in this paper.

**References**

1. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.
2. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186).
4. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3982–3992).
5. Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*.
6. Google AI Blog. (2019). Understanding searches better than ever before. Retrieved from https://blog.google/products/search/search-language-understanding-bert/
7. Algolia. (2022). Introducing NeuralSearch. Retrieved from https://www.algolia.com/neuralsearch
8. Elastic. (2023). Elasticsearch Relevance Engine. Retrieved from <https://www.elastic.co/blog/introducing-elasticsearch-relevance-engine>
9. Uddin, S., et al. (2020). A Comparative Study of Keyword-Based and Semantic Search Engines. *Journal of Information Science*, 46(4), 530–546.
10. Kaur, J., et al. (2018). A survey on semantic information retrieval. *International Journal of Computer Applications*, 179(7), 1–5.
11. Zhang, C., et al. (2023). Strategic Alignment Frameworks for Large Language Models. *IEEE Access*, 11, 136–150.
12. Alfina, I., et al. (2023). Sentiment Analysis in Social Media Using Deep Learning. *Journal of Big Data*, 10(1), 45–62.
13. Gandomi, A., & Haider, M. (2023). Beyond the Hype: Big Data Concepts, Methods, and Analytics. *International Journal of Information Management*, 63, 102437.
14. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
15. Jurafsky, D., & Martin, J. H. (2022). *Speech and Language Processing* (3rd ed.). Draft available at <https://web.stanford.edu/~jurafsky/slp3/>
16. Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and Effective Passage Search via Late Interaction over BERT. Proceedings of SIGIR.
17. Lee, K., et al. (2021). Learning Dense Representations of Phrases at Scale. Transactions of the Association for Computational Linguistics.
18. Elastic. (2023). Hybrid Search with Elasticsearch. <https://www.elastic.co/blog/bringing-hybrid-search-to-elasticsearch>