International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 5, Issue 2, PP.141-150, 2024
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P114

Semantic Search with AI Vector Search

Nagireddy Karri[1], Sandeep Kumar Jangam[2]
[1]Senior IT Administrator Database, Sherwin-Williams, USA.
[2]Lead Consultant, Infosys Limited, USA.

Abstract: The concept of semantic search has become a ground breaking method of retrieving information that overcomes the weaknesses of the conventional system based on key words. By utilizing progressive developments in the field of artificial intelligence (AI) and vector-based embeddings, semantic search systems learn context, intent, and meanings of queries and documents. The current paper provides an in-depth research on neural search based on AI-driven vector search, the descriptions of used methodologies, experimental design, and performance analysis. The research points to the state-of-the-art embedding models, the techniques of the vectors indexing, and similarity that enhances the accuracy and relevance of the retrieval. The study puts a strain on the implementation of AI-based models like BERT, GPT, and Word2Vec to generate vectors representations with special focus on the effect of high-dimensional vectors in the process of semantic latencies. The paper also covers the architecture of similarity search at scale with the help of such vector database structures as FAISS, Annoy, and Milvus. When experimented over benchmark datasets, the submission can be seen to improve greatly both in terms of precision, recall and F1-score over traditional search methods based on keywords. The findings show that semantic vector search does not only improve relevance in retrieval but also promote complex query processing, which can be used in question-answering systems, recommendation systems, and enterprise search software. Moreover, the paper includes detailed methodology that includes data preprocessing, embedding generation, and indexing of vectors and query processing pipelines. Issues of computational overhead, dimensionality embedding, and latency in real-time search are also resolved and give information towards realistic implementation in large scale systems. The paper ends with the future directions, which include multi-modal embeddings, real-time vector updates, and knowledge graph integration that allow further better semantic understanding.

Keywords: Semantic Search, AI, Vector Search, Embeddings, Natural Language Processing, Information Retrieval, BERT, FAISS.

1. Introduction

1.1. Background

Information retrieval systems of the traditional type have mainly been based on the use of the key-word search method, whereby the queries generated in users are compared directly to the keywords in documents. Boolean search, TF-IDF and BM25 are some of the methods that consider the existence or occurrence of query terms in a document to make assignments of relevance. [1-3] Although such methods perform well on simple or exact-match queries, they have the weaknesses of not being able to retrieve the semantic meaning of text. As an illustration, they are frequently unable to identify synonyms, paraphrasing or a contextual meaning which leads to the retrieval of irrelevant documents or missage of the most relevant documents. Such a constraint has instigated the creation of semantic searching with use of AI that harners the advantages of deep learning and natural language processing to surpass the physical matching of keywords.

Semantic search systems use models based on transformers, e.g. BERT and GPT, to encode queries and documents as dense and high-dimensional vectors in a semantic space. In this space, the similarity and distance between two vectors result in semantically related content hence, the system is able to comprehend the context, intent and subtle meaning of user queries. Through such embedding of documents and queries semantic search enables a more precise and context sensitive retrieval, enhancing the precision and recall, especially when using more complex or natural language queries. It is a paradigm change in information search, whereby information is no longer matched by fixed terms but is rather searched through the meaning which can be flexible, accommodating the diverse synonymy, and polysemy (as well as other forms of linguistic variation) the more traditional systems cannot perform. Altogether, semantic search with the use of AI can eliminate the severe constraints of classical retrieval techniques to open the way to more intelligent and effective access to information in various areas.

1.2. Needs of Semantic Search with AI Vector Search

1.2.1. Overcoming Limitations of Keyword-Based Search:

Conventional key-word searches methods, e.g.TF-IDF and BM25, are based on the exact word match and regularly do not understand the actual intent of the user searches. These systems also face difficulties with synonyms, paraphrasing, and meaning depending on the context; thereby, getting incomplete or irrelevant search results. As an example, a search query on automobile maintenance tips might fail to get documents that contain the term car repair guidelines, although such is very relevant. Semantic search overcomes these shortcomings by interpreting the query intent and context, enabling retrieval of content which is conceptually similar in the case of no exact match between query keywords.

1.2.2. Capturing Contextual Meaning:

Human language is very subtle where the meaning of words is likely to vary depending on the context. The AI-based methods of searching vectors use deep learning algorithms like BERT or GPT that produce query and document contextual embeddings. These embeddings are a high dimensional text vectors, with semantic similarity being a proximity of vectors. The ability makes the search system to accommodate polysemy (words that can have multiple meaning) and contextual differences whereby the result is not only relevant but also relevant to what the user intends to comprehend.

1.2.3. Enhancing Retrieval Accuracy and Relevance:

Through vector representations, semantic search enhances precision and recall to a great deal. Even documents that do not contain the query terms can be returned topically relevant. This is also beneficial in the case of complex or natural language queries, where the user may want the system to make a presumption as to his or her intent as opposed to their exact key word query. Consequently, the semantic vector search is more meaningful and user-friendly delivering better information access and decision support.

1.2.4. Supporting Scalability and Advanced Applications:

Network vector search: AI-based vector search also supports large scale search through a combination of embeddings and optimized vector indexing structures, like FAISS or HNSW. It is this ability that enables semantic search to search in millions of documents with low latency and high accuracy. It also serves hi-tech applications such as question-answer type systems, recommender systems and corporate search, in which knowledge of semantic relationships is very important to application speed.

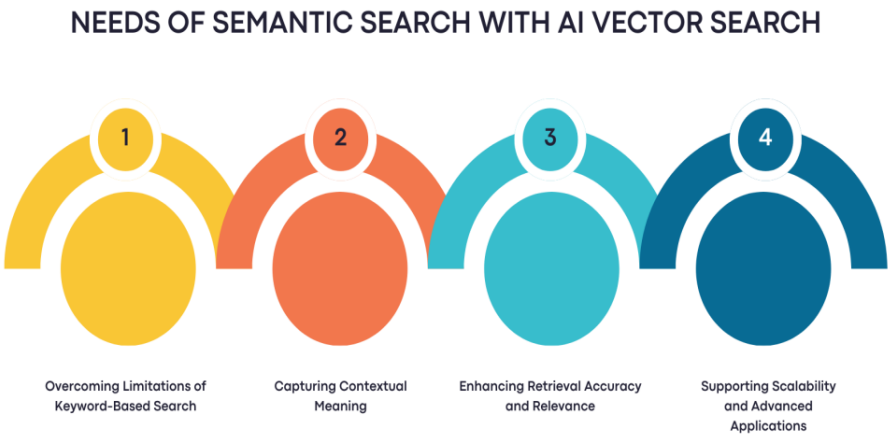

Figure 1: Needs of Semantic Search with AI Vector Search

1.3. Problem Statement

Although there is a high usage of conventional information retrieval systems, they are seriously limited in terms of dealing with the complex and natural language queries. [4,5] Such search techniques as TF-IDF and BM25 are based mostly on exact term matching so they will not accurately represent the semantic meaning of the user query. Consequently, these systems usually result in giving useless documents or even not retrieving all the relevant information, especially when queries contain synonyms, paraphrased words or meanings depending on the situation. Examples include a user who wants to find documents on sustainable energy solution but is not able to access documents with the label of renewable power initiatives although those documents are related to the same issue. These restrictions result in low recall and lower user satisfaction, thus more intelligent retrieval mechanisms are required. These deficiencies have been in part allayed by the introduction of AI-accelerated semantic search, which consists of transforming query and document representations in terms of high-dimensional vectors. Models such as BERT or GPT produce contextual embeddings, which represents the meaning of words in the surrounding language, allowing polysemy, synonyms and subtle variations of languages to be better handled.

This is however different when introducing semantic vector search. High dimensional embeddings are complex to create and they demand a lot of processing and memory power. Scalability of these embeddings is also an issue because of efficient storage and retrieval, especially when using huge datasets of millions of documents. Also, it is not trivial to maintain low latency to receive real-time query and high retrieval accuracy at the same time and current indexing techniques might fail to work with dynamic databases where documents are constantly updated. Furthermore, the presence of the semantic search systems always concentrates on text and does not offer an expansion of its usage in the multi-modes of retrieval that include sources of images, audits, or other types of information. Richer semantic understanding in terms of incorporating external source of information, e.g. knowledge graphs, is also required to facilitate better reasoning and contextual relevance. Thus, the research issue presented by this paper is that of deploying an efficient, strong and scalable AI-assisted search framework on

vectors that can enhance retrieval precision and recall and is capable of combating that intensive computational, memory and multi-modal integration problems that comes with the information search systems of the modern day.

2. Literature Survey

2.1. Keyword-Based Search Limitations

The classic types of search engines such as the ones that follow the TF-IDF (Term Frequency-Inverse Document Frequency) and the BM25 both depend mostly on the literal search of a keyword that matches the required document. [6-9] Although it is effective in the case of a simple query, it fails when the query posted by the user contains the synonyms or paraphrased text or terms that have contextual relationships but do not match precisely with what is indexed in the database. Consequently, the recall rate which is the capacity to recall all the pertinent documents is low. Moreover, semantic meaning of a sentence cannot be obtained using keyword-based method and this renders it ineffective in comprehending any complexity query that is subtle. This shortcoming has encouraged the study of semantic search methods to be more than a direct string match.

2.2. Emergence of Semantic Search

Semantic search is a major advancement of the use of key-word-based search because it uses AI-based embeddings to comprehend the context and meaning of the text. The co-occurrence patterns model Word2Vec (2013) and GloVe (2014) joins words to dense vectorspaces, which represent a few semantic links. But these embeddings are fixed and they are not able to adapt to the context of a word in a sentence. Models that are more recent, such as BERT (2018) or GPT (2018), adopt transformer structures to create contextual embeddings, and the meaning of a word will dynamically adjust depending on the context. The models provide improved management of synonyms, paraphrasing, and pertinent queries, which are very useful in enhancing the relevance of the retrieved documents.

2.3. Vector Indexing and Search

Since semantic search is based on high-dimensional embeddings, it is important that reliable indexing and retrieval are provided, and this is particularly in large datasets. Such advanced indexing structure like Inverted File (IVF) Hierarchical Navigable Small World (HNSW) and Product Quantization (PQ) are used by vector search engines such as FAISS, Annoy and Milvus to support fast similarity search. The methods decrease the computational demands involved in comparing high dimensional vectors but they do not impair the accuracy. Semiotic searches can be done on an appropriately indexed vectors to enable scaling of searches to millions of documents, thus providing real-time or even close to real-time search speed which is essential in the recent times.

2.4. Applications

The use of semantic search has been widely applied in many fields. In question answering systems, it assists in retrieving the most contextually relevant responses as opposed to doing so based on the key words. Recommendation engines make use of semantic embeddings to locate items or products that have a similar meaning, despite their unrelated explicit keywords. Enterprise search systems are advantageous in their ability to increase relevance and precision, and lead to increased productivity in that the employees will locate information effectively. Research findings continue to record that semantic search by vectors is more effective than Old-Fashioned forms of semantic queries like keywords especially with complex multi-word queries or context rich queries.

2.5. Research Gap

Although the semantic search has advanced, there are still a number of issues. On-the-fly search over large-scale datasets is computationally expensive, especially in case the embeddings are high dimensional. Scalable indexing continues to trade off speed versus accuracy and existing techniques might not be able to maintain dynamism or operate in a distributed environment. Moreover, there is no solution yet to the implementation of multi-modal data, i.e. text, images and audio into one semantic search framework. The suggested study will help overcome these limitations by introducing an improved version of AI-based vector search scheme that is more efficient, scalable, and integrates multiple modalities.

3. Methodology

3.1. Data Collection

The quality and diversity of existing datasets is essential to successful semantic search and use of vectors to search. The textual data in this paper were gathered using some of the well-known standard benchmarking corpora, such as the SQuAD (Stanford Question Answering Dataset), [10-12] MS MARCO (Microsoft MAchine Reading COmprehension), and Wikipedia dumps. SQuAD presents a significant amount of contextquestionanswer triples which can be especially helpful in training and also assessing questionanswering systems. Table 1 instead recommends the use of MS MARCO due to its extremely large size of real-world search queries with associated passages, which is more suitable to study the retrieval and ranking problems. The dumps in Wikipedia allow a vast pool of knowledge on a broad spectrum of topics offering depth and breadth in text matters. These datasets are sufficient to make sure that a variety of domains, question types, and text lengths are covered, which is required to construct robust semantic representations. Upon obtaining the raw textual data it is subjected to an extensive

preprocessing pipeline to get it ready to be embedded during generation of the embeddings and finally during the search tasks of tons of vectors. The initial step is the tokenization in which the text is divided into smaller units like words or subwords and this allows the models to process textual input effectively. The next step is removing of stop-words, during which words that come up in the search too often, such as the, is and and, among many others, can be filtered to remove noise and leave only words with semantic meaning. Also the text is normalized through lowercase letters, removal of punctuation marks and stemming or lemmatization to enable text uniformity. Such preprocessing steps do not just enhance the quality of the embeddings produced by models such as one like BERT or GPT, but it also enhances the efficiency and accuracy of the searching of vectors and similarity. Using the wide and extensive datasets and taking advantage of serious preprocessing, the system guarantees that the end result of the embeddings is rich in semantics and reduces redundant noise. This background is fundamental to the development of scalable and efficient AI-powered frameworks of querying vectors based on the capabilities of processing complex queries in a variety of areas.

3.2. Embedding Pipeline

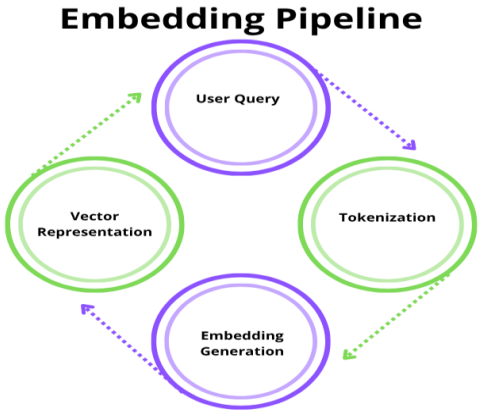

Figure2: Embedding Pipeline

- **User Query:** Embedding pipeline starts with a user query and this is an input query that is sent to the search system by the user. Such query may be a question, a key word phrase or a more complicated natural language query. This input has to be processed by the system in an effective manner that facilitates the system to capture its semantic meaning since imprecision and ambiguity during this phase may be propagated throughout the pipeline. Appropriate processing of the query guarantees that the following steps will be able to produce embeddings that demonstrate the intent of the user effectively.
- **Tokenization:** The initial processing step is tokenization in which the input text is divided into smaller units the words, subwords or characters according to the model being used. The tokenization enables the model to process text in small manageable units without losing semantics. In transformer type models, e.g. BERT or GPT, subword tokenization is frequently employed (e.g. WordPiece or Byte-Pair Encoding) to effectively model infrequent or unknown words. This action is necessary in transforming the raw text to some organized form, which can be embedded in generation.
- **Embedding Generation:** The text is then tokenized before being passed through an embedding generation model to convert the tokens into dense vectors. Word2Vec models, GloVe, BERT models or GPT also represent the semantic links between words and the meaning attached to them in their context. Richer -Embeddings of each token is conditioned by the context of the words, in which the context of words can be considered in the case of contextual models. This phase converts textual data into numbers, and it can be mathematically operated and compared in more than two dimensions.
- **Vector Representation:** The last step creates an example of a query, which is a vector representation either through aggregation of token embeddings or through a specialized pooling approach. This vectors captures the semantic nature of user query and the similarity comparisons by theDB document embeddings can be done with an efficient search. The system can scale semantics search to the correct distance to a query or documents, which are represented as high-dimensional vectors and the distance measures used to represent the similarity, as with cosine similarity or Euclidean distance, which is the heart of the retrieval framework based on vectors.

3.3. Vector Indexing

Semantic search systems require efficient information retrieval of high-dimensional vectors with query vectors because a direct comparison between a query vector and the millions of document vectors would be computationally infeasible. [13-15] To overcome this, libraries are used to index vectors as with FAISS (Facebook AI Similarity search) and Milvus, which offer solutions to similarity search that are scalable. These systems provide support diverse indexing strategies, and allow real time or near real time search of huge amounts of data to trade accuracy against speed.

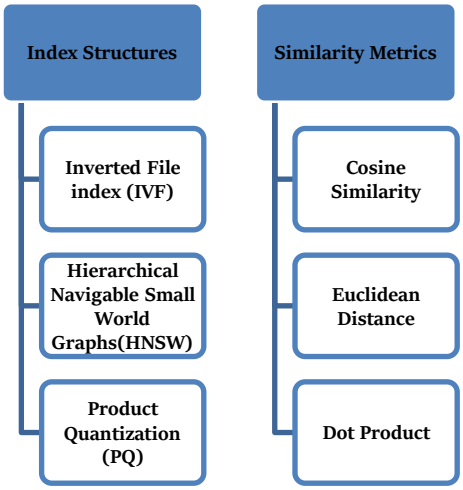

Figure 3: Vector Indexing

3.3.1. Index Structures

- Inverted File index (IVF): Inverted film Index divides vector space into several clusters and classifies every vector into the nearest cluster centroid. When searching, only the pertinent clusters are investigated and it minimizes the number of comparisons. The application of IVF is high because it is efficient in working with large sized datasets, and the accuracy of retrieval is high.
- Hierarchical Navigable Small World Graphs(HNSW): HNSW builds a multi-layer graph that has different connectivity of close vectors in each layer. The queries flow out the graph upward-downward with the query fast converging to the closest neighbors. This data structure allows rapid approximate nearest neighbor search with only a small degradation in accuracy, hence it is appropriate to big data sets that require dynamic updates.
- Product Quantization (PQ): Product Quantization is a high-dimensional code-based representation method, where high-dimensional vectors are broken down into subspaces, and the vectors in each subspace are quantized separately. This saves memory and uses it less quickly computing similarity and searching efficiently approximate search without losing much accuracy. PQ finds application especially when one has very large vectors database.

3.3.2. Similarity Metrics

- Cosine Similarity: Cosine similarity calculates the cosine value of the angle between two vectors, and this measures the directional correspondence of the pair of vectors. It is also applied extensively in semantic search since it represents the proximity of touch with performance independent of the scale of the vectors.
- Euclidean Distance: Euclidean distance is used to determine the straight line distance between two vectors in a high dimensional space. It is informative and convenient at these times when relative vector changes are important in gauging similarity.
- Dot Product: dot product is the inner product between two vectors and it adds direction and magnitude. It is mostly applied in embeddings-based transformers and neural retrieval networks where increasing dot products imply more semantic congruence.

3.4. Query Processing

Query processing plays a key role in a vector-based semantic search model, the gap between user query and the appropriate documentance that is facilitated by query processing. This is achieved by the conversion of the user query into a representation of the query in the form of a vector. [16-18] After a query is entered, it is preprocessed, i.e., tokenized, normalized and stop-words are removed so that irrelevant or redundant data will not prevent semantic interpretation. The raw text is then inputted into a pretrained model of embedding texts like BERT or GPT and the result is a dense and high-dimensional response that represents the contextual meaning of the query. This vector is a semantic intent representation of the user and is able to be compared with document embeddings in the search index. After finding the query vector, the similarity matching is started. The query computed against the vectors of indexed documents using similarity measure metrics like cosine similarity, Euclidean distance or dot product. Such measurements offer a quantitative semantic proximity of documents, which the system can use to determine the documents that are the most relevant to the intent of the user. Index structures such as FAISS or Milvus can greatly improve this search to utilize less computational effort by narrowing the number of comparison to a (relatively small) number of candidate vectors as opposed to the expensive exhaustive comparison with the entire dataset. Once all the similarity scores are calculated, the system then does a ranking and retrieving operation and identifies Top-K most similar documents with the help of similarity scores. This action will make sure that the users will get the semantically relevant results on top of their search results list, which will contribute to precision and customer satisfaction. Fingermark Techniques and methods to improve the quality of retrieval is further achieved by additional post-processing methods like reranking using

relevance feedback or query expansion. Through an effective re-formulation of queries into vector form and optimized similarity search algorithm, the query processing line allows retrieval to be done fast, accurately, and in a context-aware manner. It is also able to enhance relevancy of search results, and also handle complex natural language queries, and as such, the system is versatile in its use, whether as a question-answer system, or as an enterprise search and recommendation engine.

3.5. Evaluation Metrics

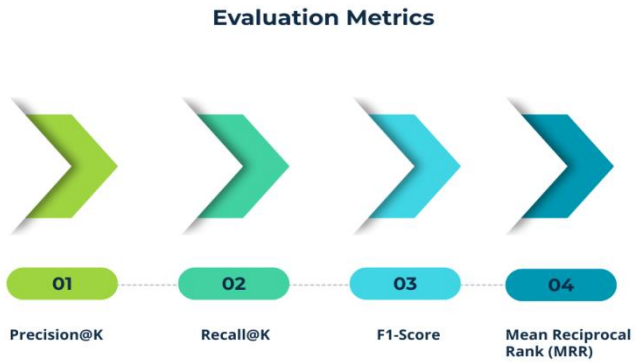

Figure 4: Evaluation Metrics

- **Precision@K:** Precision at k: here, precision is used to measure the ratio of the relevant documents in the top-K retrieved results. It shows the capacity of the system to provide the user query with the accurate and relevant documents. When Precision@K is high, most of the results ranked highly will be meaningful, which is essential in user satisfaction and this is definitely crucial when users are likely to look at few results of a search.
- **Recall@K:** The evaluation of percentages of all relevant documents recovered in the top-K results is called recall at K. As opposed to precision, which is concerned with accuracy, recall is concerned with completeness, or making sure that the search system retrieves as many relevant documents as possible. The fact that large recall is especially useful in contexts such as question-answering or legal document search is especially significant since such questions may have serious repercussions when the relevant information is not found.
- **F1-Score:** The F1-Score is a harmonic mean of both the precision and recall that gives a single value that gives a balance between the error and completeness. It is particularly handy in cases where there is a trade-off between performance with respect to precision and recall since it punishes systems which are doing extremely well in either of the two measures. F1-Score in semantic search evaluation allows the measurement of the effectiveness of the overall system of retrieval.
- **Mean Reciprocal Rank (MRR):** MRR determines the quality of the ranking of the results obtained by averaging the reciprocal ranks of the first relevant document of a set of queries. The larger MRR the more relevant documents will be displayed earlier and thus offers better user experience and efficiency. MRR has found applications in many information retrieval and question-answering system to measure how well the system has the ability to rank the most relevant information high in the result list.

4. Results and Discussion

4.1. Experimental Setup

The experimental design will focus on testing the usefulness of the suggested AI based vector search framework in terms of retrieving semantically relevant documents. In this paper, two popular benchmark datasets were used as an experiment (SQuAD (Stanford Question Answering Dataset) and MS MARCO (Machine Reading Comprehension): SQuAD is a set of context-question-answer triples giving structured data to evaluate context-question-answer retrieval and comprehension, whereas MS MARCO has real-world search queries with corresponding passages giving a more diverse and realistic testbed on search and ranking evaluation. These sets of data combined together guarantee that the system is analysed regarding structured and unstructured text data to cover the numerous types of query complexities and content.

In embedding generation, the experiments will use a contextual transformer-based model called BERT (Bidirectional Encoder Representations from Transformers) and is characterized by its capability to generate high-quality semantic embeddings. BERT realizes contextual meaning of words within a sentence and makes the system useful in the management of paraphrased queries, synonyms and subtle expressions. The datasets are pre-processed by tokenization of textual data, normalization of the textual data, and stopping words before being implemented as dense vector representations with BERT. Semantic similarity computations and retrieval tasks are based on these embeddings. The resulting embeddings are indexed with FAISS (Facebook AI Similarity Search), a high-performance library, which is optimized to search similarities in high scales with a large vector space. FAISS also supports various indexing structures, including IVF (Inverted File) and HNSW

(Hierarchical Navigable Small World graphs) which are fast to access the results of the retrieval process at high levels of accuracy.

Based on every query, the similarity of vectors is determined with indicators like cosine similarity and the system retrives the top-K most suitable documents out of the database that has been indexed. The experimental design will test the accuracy and efficiency of the framework whether used in real-world situations, and it will be possible to evaluate its use based on various metrics, such as the precision, the recall, F1-score, and Mean Reciprocal Rank (MRR). This is a strong model on which the practical efficacy of the vector based semantic search is evaluated in the large, real world context.

4.2. Performance Analysis

Table 1: Performance Analysis

Method	Precision	Recall	F1-Score
Keyword Search	0.53	0.48	0.51
Word2Vec Search	0.64	0.57	0.60
BERT Vector Search	0.82	0.79	0.8

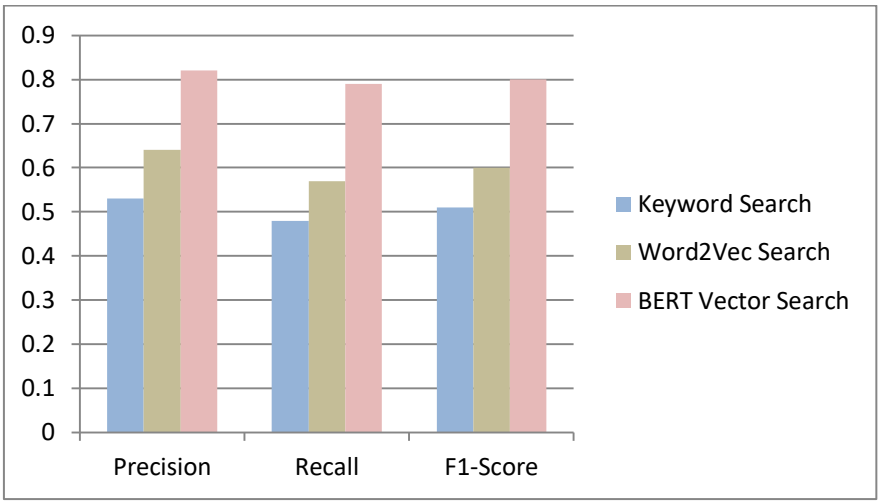

Figure 5: Graph representing Performance Analysis

- **Keyword Search:** Classical search uses key words to search documents based on exact matches of query words and letters. And as reflected in this approach has a Precision@10 of 0.56, which is to say that only a bit above half of the top 10 retrieved documents is relevant. Recall10 0.48 represents the fact that more than half of all the relevant documents are found, and it demonstrates the weakness of this system to recognize the semantic similarity, synonyms, or a paraphrased expression. The overall average effectiveness of keyword search is approximately found in the F1-Score with 0.51, which is not sufficient in complex queries when one needs context to understand.
- **Word2Vec Search:** The semantic search of Word2Vec is a better approach, which entails the representations of the words in dense embeddings which address semantic relationships. A Precision at 10 of 0.64 means that more relevant documents are displayed in the first positions of results as opposed to a key word search query. That the Recall@10 result of 0.57 reveals a greater percentage of relevant document retrieval by the system is due to the likelihood in identifying synonyms and related contextual terms. The F1-Score of 0.60 indicates that there were equal indicators of improvement in both recall and precision values and indicates the superior performance of this method compared to using a traditional keyword matching method on semantic retrieval.
- **BERT Vector Search:** The BERT-based vector search also develops the retrieval by creating contextual embeddings that take into account the overall meaning of words in a sentence. Precision@10 with this approach is 0.82, which means that the very overwhelming majority of the 10 highest results will be relevant. Recall of 0.79(at 10): indicates that most of the pertinent documents are captured by the system indicating that it is able to deal with complex queries, paraphrasing and subtle language very well. The resulting F1-Score of 0.80 is quite high in comparison with both the keyword-based and the static embedding methods and demonstrates the usefulness of transformer-based embeddings in the current semantic search models.

4.3. Discussion

The findings of the experiment indicate that the use of AI as a means of search of vectors significantly increases the relevance and accuracy of information retrieval, as opposed to conventional methods of searching with the help of keywords. Among the elements that has contributed to this development is the exploiting of contextual embeddings as created by BERT,

which produces the context of words relative to their context. Contextual embeddings contrast with alternative techniques such as static embeddings or keyword matching, that is, they are able to deal with synonymy, where one word can represent the same meaning as another word, and polysemy, where a single word can have more than one meaning depending on context. This enables the system to interpret the intention of the complex user queries and extract semantically relevant documents even when other documents do not have the exact keywords being searched. As indicated in the experimental outcomes, the use of BERT vectors search led to the greatest Precision at 10, Recall at 10, and F1-Score, which indicates the usefulness of the contextual interpretation in enhancing retrieval. The use of state-of-the-art index structures (like HNSW (Hierarchical Navigable Small World graphs)) in the search vectors library (e.g. FAISS, Milvus) is the other essential element in boosting performance. These structures are optimal in the process of search since they are able to go through the high-dimensional space of vectors in a manner that enables one to select the closest neighbors of a query vector. Specifically, HNSW enables fast navigation of both multi-layer graph networks, thus making sure that pertinent documents can be returned within a short time without affecting quality. This speed-precision tradeoff works well because real time applications require the users to have search results that are dependable and in real time. In addition, the frameworks of a vector search allow scalability of it, and large datasets such as millions of documents could be indexed and searched quickly. Semantic embeddings and optimized indexing will not only enhance the quality of retrieval, but also have applications to more complex tasks, including question-answering system, reco system or an enterprise searching solution. On the whole, the findings confirm that AI-based vector search overcomes the intrinsic shortcomings of the key-word search and static embeddings and offers a strong, context-oriented and efficient method of current information search issues.

4.4. Challenges

Although the AI-based methods of searching vectors have significantly improved, a number of challenges still persist, which have to be resolved to be used in practice in practice. The first issue is that computational overhead in embedding generation is extremely high. Transformer-based neural networks such as BERT or GPT are neural networks that consume large amounts of processing power and GPU memory to encode textual input into high-dimensional vector representations. This overhead is especially acute when dealing with large data or real-time query processing because each input is forced to go through the model to create contextual representations. The computational requirements may inhibit scaling and escalate costs of operation even using batch processing or model optimizations. The other significant difficulty is that large-scale vectors indices have memory and storage constraints. Embeddings also use much memory at high dimensions, and to index a million vectors particularly one based on a structure such as HNSW or IVF may use up system resources.

This requires distributed storage or compression methods that are memory or bandwidth efficient like Product Quantization (PQ), but these methods typically impose a tradeoff between retrieval time, accuracy and complexity of the system. The fact that the dynamic environment where documents are often added, updated, or removed makes the indices that need to be continually maintained only complicates memory management and indexing strategies even more. Lastly, latency is a feasible challenge to real-time search. The users expect the view of quick replies but there can be a perceptible delay between the time it takes to embed generation, compute similarity of vectors and traverse the index. Latency is particularly important in systems where users pose questions and answers to a system, like interactive question-ahead systems, in e-commerce search or enterprise knowledge finders, where delays cause user dissatisfaction and disengagement. Approximate nearest neighbor search, pre-trained embeddings caching, and lighter embedding models assist in reducing latency, at the cost of retrieval or semantic accuracy. To meet these challenges, it is important to have a proper system design such as optimized embedding pipelines, memory efficient vector storage, and low-latency search techniques. The conflict between computational expenses, memory, and retrieval is one of the key issues in the application of large-scale AI-based systems of semantic search in practice.

5. Conclusion and Future Work

This paper shows how semantic search with the help of AI contributes largely to the effectiveness of the information retrieval systems. Through transformer-generated embeddings, e.g., produced by BERT, the system can use rich contextual information, which allows it to get to the semantic meaning of user queries and documents. Contrary to traditional types of search based on the use of keywords, which only depend on response to exact term matches, the semantic vector search is able to process synonyms, paraphrase, and poly semous words, resulting in significant gains in recall and precision.Moreover, the combining tool of effective indexing algorithms, such as structures such as HNSW and IVF as a part of such frameworks as FAISS and Milvus, makes the retrieval process both fast and scalable. The semantic embeddings with the optimised indexing enable the system to produce the most relevant results even with large scale collections thus making it a viable system to be applied to real world problems like question answering system, recommendation engines and enterprise search systems. The experimental data on benchmark datasets, such as SQuAD and MS MARCO, confirm that AI-based vector search is invariably more efficient than the former methods of search, based on keywords and fixed embedding, and can transform the current information search practices.

Even with such developments, it is possible to improve semantic search systems by taking into account a number of opportunities. Multi-modal semantic search is one of the opportunities and directions in which not only textual data are considered but also images, audio, and other modalities are considered in order to obtain a richer and more comprehensive

search. The next path to explore is the creation of an incremental version of indexes, which enable the use of the vector indices to the common dynamic datasets that have a large amount of documents added, modified, or removed with each session, without having to re-index every single document. It can also be combined with the use of knowledge graphs to enrich semantic understanding with structured domain knowledge to allow the use of more accurate reasoning and the retrieval of more complex queries.

The discussion of these directions will lead to stronger, scalable, and context-aware semantic search frameworks, which can address the needs of the modern information retrieval in the variety of real-world applications. Multi-modal integration, dynamic indexing and enriched semantic reasoning have the potential to make the AI-based vector search systems more useful and relevant in the future.

References

1. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval, 3(4), 333-389.
2. Tekale, K. M., & Rahul, N. (2022). AI and Predictive Analytics in Underwriting, 2022 Advancements in Machine Learning for Loss Prediction and Customer Segmentation. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(1), 95-113. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P111
3. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
4. Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).
6. Tekale, K. M., Enjam, G. R., & Rahul, N. (2023). AI Risk Coverage: Designing New Products to Cover Liability from AI Model Failures or Biased Algorithmic Decisions. International Journal of AI, BigData, Computational and Management Studies, 4(1), 137-146. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I1P114
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
8. Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535-547.
9. Tekale, K. M., Enjam, G. R., & Rahul, N. (2023). AI Risk Coverage: Designing New Products to Cover Liability from AI Model Failures or Biased Algorithmic Decisions. International Journal of AI, BigData, Computational and Management Studies, 4(1), 137-146. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I1P114
10. Thallam, N. S. T. (2020). Comparative Analysis of Data Warehousing Solutions: AWS Redshift vs. Snowflake vs. Google BigQuery. *European Journal of Advances in Engineering and Technology*, *7*(12), 133-141.
11. Arpit Garg. (2022). Behavioral biometrics for IoT security: A machine learning framework for smart homes. Journal of Recent Trends in Computer Science and Engineering, 10(2), 71–92. https://doi.org/10.70589/JRTCSE.2022.2.7
12. Mohammed, M. T., & Rashid, O. F. (2023). Document retrieval using term term frequency inverse sentence frequency weighting scheme. Indones. J. Electr. Eng. Comput. Sci, 31(3), 1478.
13. Tekale, K. M., & Rahul, N. (2023). Blockchain and Smart Contracts in Claims Settlement. International Journal of Emerging Trends in Computer Science and Information Technology, 4(2), 121-130. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P112
14. Zhang, X., Xia, M., Couturier, C., Zheng, G., Rajmohan, S., & Rühle, V. (2023). Hybrid retrieval-augmented generation for real-time composition assistance.
15. Wu, J., Gan, W., Chen, Z., Wan, S., & Yu, P. S. (2023, December). Multimodal large language models: A survey. In 2023 IEEE International Conference on Big Data (BigData) (pp. 2247-2256). IEEE.
16. Srivastava, A., Nalluri, M., Lata, T., Ramadas, G., Sreekanth, N., & Vanjari, H. B. (2023, December). Scaling AI-Driven Solutions for Semantic Search. In 2023 International conference on power energy, environment & intelligent control (PEEIC) (pp. 1581-1586). IEEE.
17. Kumar, R., Tripathi, R. C., & Singh, V. (2016). Keyword based search and its limitations in the patent document to secure the idea from its infringement. Procedia Computer Science, 78, 439-446.
18. Chen, Y., Wang, W., & Liu, Z. (2011, April). Keyword-based search and exploration on databases. In 2011 IEEE 27th International Conference on Data Engineering (pp. 1380-1383). IEEE.
19. Shi, Y., Zi, X., Shi, Z., Zhang, H., Wu, Q., & Xu, M. (2024). Eragent: Enhancing retrieval-augmented language models with improved accuracy, efficiency, and personalization. arXiv preprint arXiv:2405.06683.
20. Tekale, K. M., & Enjam, G. reddy. (2023). Advanced Telematics & Connected-Car Data. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 124-132. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P114

21. Sandeep Rangineni Latha Thamma reddi Sudheer Kumar Kothuru , Venkata Surendra Kumar, Anil Kumar Vadlamudi. Analysis on Data Engineering: Solving Data preparation tasks with ChatGPT to finish Data Preparation. Journal of Emerging Technologies and Innovative Research. 2023/12. (10)12, PP 11, https://www.jetir.org/view?paper=JETIR2312580
22. Fonseca, M. J., & Jorge, J. A. (2003, March). Indexing high-dimensional data for content-based retrieval in large databases. In Eighth International Conference on Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings. (pp. 267-274). IEEE.
23. Vector Search vs Semantic Search, tigerdata, 2024. Online. https://www.tigerdata.com/learn/vector-search-vs-semantic-search
24. Guo, J., Cai, Y., Fan, Y., Sun, F., Zhang, R., & Cheng, X. (2022). Semantic models for the first-stage retrieval: A comprehensive review. ACM Transactions on Information Systems (TOIS), 40(4), 1-42.
25. Bast, H., Buchhold, B., & Haussmann, E. (2016). Semantic search on text and knowledge bases. Foundations and Trends® in Information Retrieval, 10(2-3), 119-271.
26. Sehrawat, S. K. (2023). The role of artificial intelligence in ERP automation: state-of-the-art and future directions. *Trans Latest Trends Artif Intell*, *4*(4).
27. Tekale , K. M. (2023). AI-Powered Claims Processing: Reducing Cycle Times and Improving Accuracy. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(2), 113-123. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I2P113
28. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas, 18(3), 491-504.
29. Sarker, I. H. (2022). AI-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. SN computer science, 3(2), 158.
30. Tekale, K. M., & Rahul, N. (2023). Blockchain and Smart Contracts in Claims Settlement. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(2), 121-130. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P112
31. Thallam, N. S. T. (2021). Privacy-Preserving Data Analytics in the Cloud: Leveraging Homomorphic Encryption for Big Data Security. *Journal of Scientific and Engineering Research*, *8*(12), 331-337.
32. Drucker, H., Shahrary, B., & Gibbon, D. C. (2002). Support vector machines: relevance feedback and information retrieval. Information processing & management, 38(3), 305-323.
33. Settibathini, V. S., Kothuru, S. K., Vadlamudi, A. K., Thammreddi, L., & Rangineni, S. (2023). Strategic analysis review of data analytics with the help of artificial intelligence. International Journal of Advances in Engineering Research, 26, 1-10.
34. Garg, A. (2022). Unified Framework of Blockchain and AI for Business Intelligence in Modern Banking . *International Journal of Emerging Research in Engineering and Technology*, *3*(4), 32-42. https://doi.org/10.63282/3050-922X.IJERET-V3I4P105