The Eastasouth Journal of Information System and Computer Science
Vol. 3, No. 03, April 2026, pp. 323-336
ISSN: 3025-566X, DOI: 10.58812/esiscs.v3i03

Semantic Search with Vector Database: A Comprehensive Review of Models, Indexing and Applications

Tanay Chowdhury
Data Science Lead – Gen AI Center of Innovation, Amazon Web Services, Seattle, USA

Article Info

Article history:

Received Mar, 2026
Revised Mar, 2026
Accepted Mar, 2026

Keywords:

Embeddings;
Indexing Techniques;
Information Retrieval;
Semantic Search;
Similarity Search;
Transformer Models;
Vector Databases

ABSTRACT

The use of semantic search with the help of vector databases has become an impressive paradigm of retrieving the pertinent information by offering the contextual and conceptual sense of the information searching more than using the conventional methods of keyword searching. This paper provides an in-depth overview of the models of vector representation, transformer-based semantic encoders, and technologies of vectors database that jointly allow efficient and error-free semantic search. Classical distributional semantics, word-level embeddings, and transformer architectures are presented as background methods of making designed generating meaningful vectors representations. The paper also looks at the contemporary databases of vectors and indexing mechanisms which enable scalable similarity search in high-dimensional data. Moreover, different distance measures, hash algorithms and indexing strategies based on graphs are evaluated to determine how they can be used to maximize retrieval. Lastly, the paper presents practical examples of semantic searching with the use of the vector databases with text, image, audio and conversational applications, outlining both the main challenges and research opportunities.

Corresponding Author:

Name: Tanay Chowdhury
Institution: Data Science Lead – Gen AI Center of Innovation, Amazon Web Services, Seattle, USA
Email: tanayz@outlook.com

1. INTRODUCTION

The evolution of the Internet era has brought information into the Big Bang stage. People may acquire vast volumes of ambiguous information in their everyday lives, particularly with the quick development and widespread use of multimedia technologies. Nowadays, text is the primary medium for conveying multimodal information. As a result, NLP activities such as textual information mining, analysis, and classification have become commonplace. Text categorization is one of NLP's primary duties. Text classification is primarily categorized into sentiment classification, news classification, subject classification [1], Q&A matching [2] and other categories based on the scenario and content. The information era produced news text, which is notable for its abundance of data, high demand for real-time information, challenges with human tagging, etc. High accuracy and speed are required for a model that can quickly classify news into the appropriate categories and minimize human labor [3].

The fact that the article deals with word meanings (semantics) and uses word meanings to explain ideas about word

meanings may be viewed as very sarcastic. This demonstrates two things: The first is that, at least when the words have been alphabetized, it is normal practice to debate the meanings of other words by using them; the second is that "handling the world with words" is a universal challenge." This looks reasonable so far, but there's a catch. It should be noted that, paradoxically, the term 'semantic(s)' lacks a precise definition when considering the terms listed above in the sequence they are introduced [4]. It is commonly recognized that Michel Boreal coined the term 'communique' in 1883 as a French word. Since then, the theoretical idea has been crucial to those working in the "language sciences," such as philologists and linguists who study word meanings and philosophers who attempt to characterize and explain the (mainly logical) mechanism of language meaning and the interpretations of what are referred to as sentences.

Essential information in texts is provided by the semantic connections between ideas. They have the ability to identify the text category under analysis. Additionally, automatic text categorization algorithms have the ability to express them in structures that can be processed. Natural language processing (NLP) tasks include representing words, relationships, context, and any other information from texts. The computer has generally been able to comprehend the data. It is possible to apply appropriate operations like addition, subtraction, and distance measurements to the vectors, which have an attractive and intuitive meaning [5].

The past ten years have seen a large number of works on NLP [6]. These have addressed the many stages of text processing, including text preparation, vectorization, and final text comprehension. By projecting words from a language corpus into a vector space, vector space modelling seeks to place words with similar meanings next to one another. At the moment, the subject of vectorization is approached from two perspectives. While the first one concentrates on contextualizing words, the second one addresses complete texts and how individual phrases and paragraphs are represented within the context of the papers in issue.

The scientific topic of automatic text processing is expanding quickly. The majority of the knowledge pertaining to this subject is focused on data mining, which is currently a very significant and well-liked area of computer science. The classification of written statements is one of the procedures that allows us to talk about intelligent text analysis. It is also so simple that it is ideal as a marker for the functioning of further text processing stages. It is possible to analyses any written statement [7], including scientific publications, messages from information services, and informal conversations. Automatic text analysis techniques face distinct obstacles depending on the kind of content.

Structure of the Paper

The paper is structured as follows: Section II presents vector representation models and vector databases for semantic search, Section III describes indexing and similarity search techniques, Section IV discusses key use cases of semantic search using vector databases, Section V reviews related literature, and Section VI concludes the study with future research directions.

2. VECTOR REPRESENTATION MODELS AND DATABASES FOR SEMANTIC SEARCH

The coding of concepts with vectors of a vector space, whereby simple operations like addition and tensor products combine two or more vectors to create new representations to accomplish tasks such as language processing and characterizing conceptual similarity, is what is known as vector representation. Semantic vectors are the attempts to describe words in a multidimensional semantic field as a point, which is an efficient method to model the meaning of words [8]. A common practice in natural language processing (NLP) involves representing words in a sentence as a context-based semantic embedding or language

statistics in the form of semantic vectors or embeddings. Semantic embedding spaces are Euclidean vector spaces, with semantic categories, as represented in language, being associated with vectors. These vector representations are of a much lower dimensionality than naive one-hot encodings, and the fact that semantically similar words should be represented by vectors nearby each other in the embedding space. Spatial semantic representations represent terms as vectors in a high-dimensional space according to how frequently they emerge in specific circumstances.

2.1 Vector Representation Models

Cognitive science has seen an increase in the use of word meaning based on vectors. These models are attractive because of the fact that they solely present the representation of meaning through distributional information with the assumption that words used in a similar context have similar semantics. Although commonly used, the existing literature on vector-based models generally aims at describing words in isolation, and little research has been done on how to build representations of phrases or sentences. This stands in stark contrast to experimental data that suggests semantic similarity is even more sophisticated than a simple relationship between individual words, such as that seen in sentential priming.

1. Classical Distributional Semantics

A usage-based model of meaning, distributional semantics (DS), also known as vector space semantics, is predicated on the idea that the statistical distribution of linguistic objects in context significantly influences their semantic behavior. The lexicon is its primary emphasis: DS is primarily an empirical method for the analysis of lexical meaning DS provides a format of capturing meaning and algorithms to learn to capture meaning based on language information. As digital texts become more widely available, distributional models may use vast quantities of empirical data to describe the semantic characteristics of words. Distributional representations, which are constructed from text corpora as examples of language use, provide fresh approaches to addressing the dynamicity and flexibility of meaning as well as the interaction between meaning and situations.

2. Term Frequency-Inverse Document (TF-IDF)

Term Frequency and Inverse Document Frequency are two distinct terms that are combined to form TF-IDF. The number of times a phrase appears in a document is measured using TF. Since it is commonly recognized that papers can range in length from extremely short to very long, it is possible that a term appear more frequently in large documents than in tiny ones. In order to address this problem, the frequency of a word is calculated by dividing its occurrence in a document by how many of terms there are in the text. shall now talk about inverse document frequency. The system evaluates all keywords identically when determining a document's term frequency, regardless of whether they are stop words like "of," which is erroneous. Every keyword has a unique significance. Words that occur frequently are given less weight by the inverse document frequency, whereas words that occur infrequently are given more weight [9]. It is therefore acknowledged that a word's bigger or higher occurrence in documents result in a higher term frequency, and a word's lower occurrence in papers result in a higher significance (IDF) for the keyword sought in that specific document. Term frequency (TF) and inverse document frequency (IDF) are simply multiplied to create TF-IDF.

3. **Latent Semantic Analysis**

Latent Semantic Analysis is a form of word meaning representation and extraction of contextual meaning of words by performing some statistical calculations on a corpus of text. It used to be called Latent Semantic Indexing (LSI). Prior to LSI, information was retrieved by employing lexical matching techniques to precisely match terms in texts with the query. Synonyms (missing documents about "automobile" when searching for "car") and polysemy (obtaining documents about a financial bank when searching for "river bank") were two issues that made these approaches challenging to utilize for information retrieval. Moreover [10], it has been recently established that it is possible to give a statistical interpretation of the traditional Latent Semantic Analysis (LSA) paradigm, it uses the "Singular Value Decomposition" (SVD) methodology, a linear algebraic method, to extract hidden ideas from the corpus' documents.

4. **Word Level Embeddings**

One of the pillars of the textual data representation and an input to the Machine Learning tools is word embeddings. They are shortenings of words to vectors of the real numbers. In the broadest sense, word embeddings are the numerical representation of words, often in the form of a vector in Rd. There are several definitions for word embeddings. More precisely, word embeddings are vectors of unsupervised learnt word representations whose relative similarity corresponds to semantic similarity. In computational linguistics they are often referred as distributional semantic model or distributed representations. As words are stored as indices in a dictionary, many existing NLP systems and approaches regard words as atomic entities; there is no concept of similarity between words. Simplicity, robustness, and the finding that straightforward models trained on vast volumes of data perform better than intricate systems trained on smaller quantities of data are some of the strong arguments for this decision.

5. **Word2Vec**

The Word2Vec [11] word embedding generator aims to detect the meaning and semantic relationships between the words by looking at how frequently a term appears in a specific corpus of documents. This algorithm's concept is to use statistics and ML to model word context and create a vector representation for every word in the corpus. The generated word vector representations make it possible to identify word relatedness. The verbs capture and catch, for instance, are linked to comparable vectors while having different syntactic structures and sharing a shared meaning. It is possible to train a Word2Vec model using either the Skip-gram technique or the Continuous Bag-Of-Words (CBOW). Since a preliminary study showed that the Skip-gram method produced better results, used it in work.

6. **FastText**

Facebook's FT algorithm treats all words as n-grams of characters. Providing vector representations for words that are not in the lexicon is beneficial [12]. The current method uses FastText embeddings to create token vectors of dimension 300. The token vectors are averaged to form each vector that represents a tweet.

7. **Glove**

Glove is a dimensionality reduction-based word embedding learner [13]. To differentiate between important and irrelevant words,

instead of learning raw co-occurrence probabilities, it learns co-occurrence probability ratios. GloVe embeddings that are trained on big Common Crawl were used in work.

2.2 *Transformer-Based Models for Semantic Search*

Mobile App Stores employ search engines extensively, and millions of consumers use them daily worldwide. Even though recent advancements in the fields of ML, information retrieval, and NLP provide more advanced semantic approaches, several retailers continue to employ simple lexical-only search engines. Due to the widespread use of mobile devices in society, most businesses now view them as essential tools for maintaining close communication with their clients.

1. **Lexical Models**

   The lexical model is a modification of Aptoide's present method, seeing an application's name and Elastic Search is used to index the description as bags of words. After rating each indexed application, the top N are given back in answer to a query q. The Lucene Scoring Function is used to calculate the scores for an application across the selected fields (name and/or description).

2. **Bi-Encoders and Cross-Encoders**

   Modern neural models for transformer-based encoders are used for retrieval as either cross-encoders or bi-encoders.

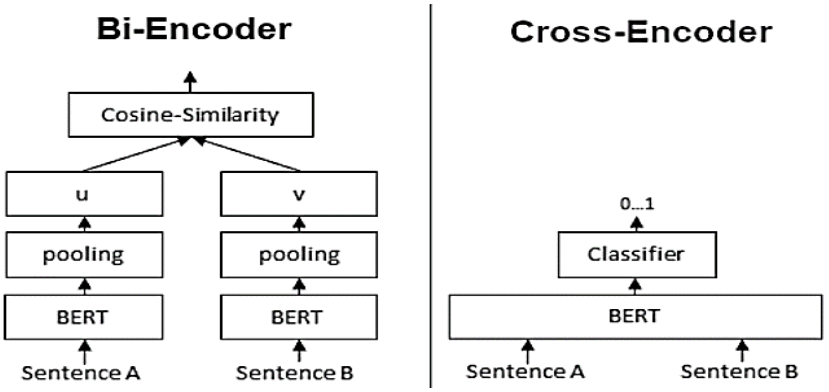

Figure 1. Bi-encoder and Cross-encoder

The input into cross-encoders is a concatenation of sentences and the representation of both is produced using the [CLS] token, e.g. A score can be calculated using this representation e.g. by passing it through a linear layer with sigmoid activation. On the other hand, bi-encoders represent sentences individually and this enables indexing of individual representations by techniques that facilitate quick implementation of maximum inner product searches [14]. More often, cross-encoders perform better because they capture two-sentence interactions according to the performances. However, as reaction speed is a critical feature in the situation, bi-encoders would be more suited for a mobile app search engine due to their higher processing capacity. The overall design of the Bi-encoder and Cross-encoder is seen in Figure 1.

2.3 *Vector Databases for Semantic Search*

A database of vectors, in particular, called Vector Databases (VDBs) is created with the express purpose of managing and storing vectors of high dimensions. In particular, VDBs keep the information in the high-dimensional vectors that are the features or attributes of data. The number of these high-dimensional vectors is typically dozens or thousands, based on the underlying data's granularity and complexity. In contrast to the conventional relational databases, VDBs

offer effective storage, management, and search of high-dimensional vectors at scale.

1. **Pinecone Vector DB**

   Pinecone is not a database, it is a cloud-native vector database that is specifically designed to support similarity search and recommendation systems. It offers a very effective and scalable system in the storage and query of high dimensional vector data. Pinecone uses high-level indexing and search algorithms to enhance the process of similarity search queries. It provides an indexing method known as Approximate Nearest Neighbor (ANN) search, which enables one to search and retrieve vectors that are nearest or similar to a particular query vector. Approximate methods enable Pinecone to provide the low-latency search operations with very large datasets.

2. **Weviate**

   Weaviate is a vector database which supports the storage of both objects and vectors and supports a combination of vectors search. It is scalable with very large machine learning models, which are easy to scale because it is modular, cloud-native and real-time. Weaviate has optional text, image, and other media type modules which can select depending on task and data. It is also possible to use a combination of more than one module as the data differs.

3. **Milvus**

   An open source vector database called Milvus aims to enhance AI and encourage similarity search embedding [15] applications. It's a ground-breaking tool that makes searching for unstructured data more accessible and offers a consistent user experience across all deployment scenarios. In 2019, the source code for Milvus was made available on GitHub under the Apache 2.0 license. By September 2023, Milvus had amassed over 22,868 GitHub stars, ranking first among all vector search technologies.

3. INDEXING AND SIMILARITY SEARCH TECHNIQUES

The potential to exploit semantic information in information retrieval has been strengthened by the expanding use and improving performance of automated [16] semantic annotation and entity linking systems. Incorporating semantic information can enhance retrieval performance by facilitating more accurate sense disambiguation, purpose determination, and instance identification, among other things. Excellent performance has been shown by keyword-based information retrieval systems. The capacity to index data other than what is often seen in inverted indices, including type relationships and entity mentions, is one of the difficulties in efficiently designing a search engine that can take semantic information into account. Data broadcasting is a compelling way to distribute large amounts of data to several mobile users at once, meeting their demands. By reducing the request time, indexing helps find data from the database rapidly, eliminating the need to scan the whole database for a specific data item request.

3.1 Data Metrics for Vector Similarity

Extraction of information from data is known as data mining. The use of huge datasets for data mining is common. Predicting future events, aiding in medical diagnostics, and anticipating chronological relationships are just a few of the disciplines that have made use of data mining skills. Distance metrics are one method for calculating the separation between a new data point and an existing training dataset.

1. **Euclidean Distance**

   The matrices of point squared distances are known as Euclidean distance matrices (EDM). Its description is surprisingly straightforward: because of their numerous beneficial qualities, they

have been used in acoustics, wireless sensor networks, machine learning, psychometrics, crystallography, and more [17]. EDMs are valuable, but the signal processing community doesn't appear to know enough about them.

2. **Manhattan Distance**

In situations, the Manhattan distance, city block distance, or taxicab geometry can be used to determine the distance between two data points on a grid-like pattern. As a result, the Manhattan Distance dominates the Euclidean distance metric as the dimension of the data increases. This is a consequence of the "curse of dimensionality."

3. **Hamming Distance**

The Hamming distance is a statistic used to compare two binary data strings. The number of bit locations where two binary strings of equal length differ from one another is known as the Hamming distance. The Hamming distance between two strings, a and b, is represented by the symbol d(a,b). It applies the XOR operation (a⊕b) on two strings to get their Hamming distance, and then count the number of 1s in the resulting string.

4. **Cosine Distance and Cosine Similarity**

The cosine distance and cosine similarity metrics are mostly used to find similarities between two data points. As the cosine distance between two data points increases, the degree of similarity between them decreases, and vice versa. Points that are closer to one another are therefore more similar than those that are farther away. Cos θ denotes cosine similarity, whereas 1-cos θ denotes cosine distance.

3.2 Hashing Techniques

Hash algorithms enable efficient and secure data gathering, retrieval, and transfer, they are now a crucial tool in computing. Using a variety-length input, a hashing algorithm generates a quantity-length result, which is the hash value and digest of a message. The outcome is suitable for efficiently storing and sending vast volumes of data since it is frequently significantly less than input data. The hashing technique finds application in several aspects in computing including data handling, information security, and others. Hashing serves a purpose, for instance, in handling data to build indexes that offer quick use of data kept in databases. Moreover, hashing ensures the integrity of data with regard to safety whereby it is used to detect any changes that may have taken place in the middle of transmission or storage. In addition to other safety-related tasks, hashing is utilized in encrypted signatures and password verification. Finding more efficient projection functions is the main goal of most hashing algorithms. But the quantization step's accuracy loss has been overlooked and little examined, despite the fact that it is just as crucial to the accuracy of the final search. To make the modern computers secure and efficient, new algorithms should be developed, and old algorithms should be examined on a regular basis. There were a few techniques of hashing techniques are shown in Figure 2:

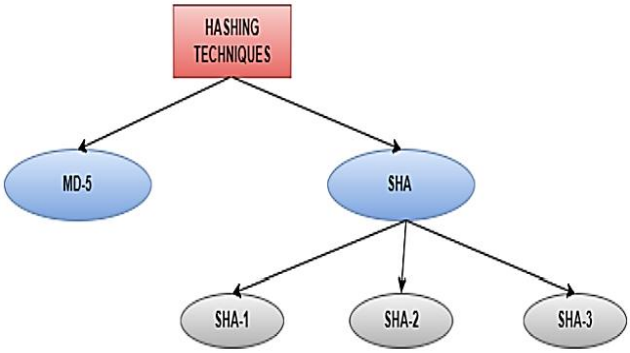

Figure 2. Hashing Techniques

3.3 Hashing Techniques

a. **Message Digest Algorithm 5:** A 128 bits hash code generated by the popular encrypted hashing technique Message Digest Algorithm 5 (MD5) is commonly shown as a 32-character hex integer. It gives an output whose output size of the data has a fixed size dependent on any dimension on an input communication. However, because of flaws discovered in MD5, this is regarded as a poor hashing method and is not suggested for situations that require privacy.

b. **Secure Hash Algorithm 1:** Another popular, secure hashing method is Hexadecimal numbers of 40 characters are used to represent the 160-bit hash values generated by Secure Hash Algorithm 1 (SHA-1). Similar to MD5, it generates a result with a set hash size from an input message of arbitrary length [18]. Therefore SHA-1 is also regarded as weak and recently deprecated because of security flaws. Therefore, its use is not recommended for crypto applications.

c. **Secure Hash Algorithm 2:** A group of cryptography hashing approaches known as Secure Hash Algorithm 2 (SHA-2) contains hash result lengths for 224, 256, 384, 512 along with 512/224 bits. These techniques are frequently employed for many safety-related tasks and generate hash coefficients with multiple ranges, comprising 224 bits up to 512 bits. As a result, most cryptographic users advocate using SHA-2 since it is far safer than MD5 or SHA-1.

d. **Secure Hash Algorithm 3:** The most recent encryption hash method, Secure Hash method 3 (SHA-3), was created to provide more security than SHA-2. With an increasing number of flaws found in MD5, SHA1 and SHA-2, this algorithm attempts to counter this. Compared with preceding generations, SHA-3 is intended to have been highly safe and immune to assaults. In addition, it provides hash result lengths for 224, 256, 384, along with 512 bits.

3.4 Graph-Based Indexing Techniques

The development of an index in query processing is necessary to overcome the following difficulties: effectively building a big database, effectively managing dynamic changes, and effectively handling workloads involving bulk queries [19]. Batch processing can also improve indexing by enhancing query processing, particularly when the query graphs make use of commonality.

a. **KR+ index and secondary indexes:** The KR+-index is a multi-dimensional, scalable index that is based on the current cloud data management (CDM). First, data is divided using the R+ tree, and the rectangular nodes of the tree index leaf are thought of as dynamic grids. R+-tree works well since it allows us to balance grid size and access time by varying the parameters. Additionally, R+-tree prevents leaf

nodes from overlapping, making it impossible to get comparable data twice. As a result, each leaf node's rectangle may be set using a distinct key.

b. **R-tree and B-tree index:** An indexing approach for both location-aware and region-aware searches is used to determine the top k queries. In a novel method, the R-tree for spatial database searches and the inverted file for text retrieval are closely combined. This method combines text relevance and spatial closeness to reduce the search space during query processing, and it contains algorithms that use the recommended index to evaluate the top-k queries.

c. **Lindex and network structured index:** a graph index called Lindex that incorporates each database graph subgraph. The nodes in Lindex stand for key-value pairs, in which a list of database graphs that include the key is the value and a database subgraph is the key. The purpose of Lindex is to increase sub's efficiency. Built with any feature set of choosing, this subgraph querying architecture is scalable and quick. An index with a network topology and graph clustering enhances query processing scalability. The straightforward K-medoids algorithm—a discrete variant of the K-means data clustering method—is used. A distance metric and a number of node annotations make up the index.

4. USE-CASES OF SEMANTIC SEARCH USING VECTOR DATABASES

The vast volume of information in today's digital age necessitates the development of user-friendly and efficient search features and apps. To maximize their information management, organizations require internal search capabilities in addition to publicly accessible search apps. In the digital age live in today, the amount of information is continually increasing. Web search engines and other platforms that offer search capabilities are highly advanced technologies. Nearly everyone who uses the internet is familiar with Google Search in particular. In addition to the standard Google Search, Google has created search features that are optimized for photos, videos, books, travel, finance, and scholarly publications. Even for a novice user, all of these features are quite simple to use, which raises expectations for all other search apps and features.

4.1 Similarity Search in General

There are several applications for vector data. Approximate similarity search, the foundation of nearly all vector database retrieval procedures, may effectively employ any data item that can be meaningfully vectorized. One should note that while the following subsections focus on some of the most typical use-cases of the vector database, this is by no means an exhaustive list. Vectors are utilized in automatic black-and-white picture colorization, facial expression detection, digital image asset tracking, recommender systems, and the storage and comparison of molecular structures and rental flats, for instance.

4.2 Image and video similarity search

In a Similar to Greek plays, pictures may be vectorized, although the process is usually more involved and includes feature extraction before vectorization and picture normalization with respect to pixel values and size. One method for feature extraction, which is usually done outside of the VDBMS, is to run the pictures through a convolutional neural network one at a time. From basic aspects like the existence of vertical and horizontal borders and basic forms to textures like hair, grass, and water, the technique extracts progressively more abstract data from the image. These vectorized characteristics are employed in similarity searches. In terms of the captured characteristics, Images with comparable vector representations most

likely appear to be similar. However, only a representative portion of the frames may be taken into consideration, videos are often split into single frames for video vectorization. similar stand-alone images, it uses each frame's properties to create a feature vector that represents the content. Furthermore, temporal information is frequently required to fully comprehend the video's content.

Voice recognition The voice [20] recognition as a vector is applied in the same way as video vectorization and search. The audio is digitized and split into brief frames, each of which represents a chunk of the audio, if it is in analogue format. The frames are finally saved as a feature vector after being normalized, filtered, and processed using various methods. This means that the entire audio is a collection of feature vectors that together represent a spoken word, sentence, song, or other type of audio. A similar procedure might be conducted to a spoken key phrase if speech recognition is utilized for user authentication, and Vectorized recordings and the vectorized spoken key phrase might be compared. However, if a conversational agent uses speech recognition, the vector sequences may be utilized as input for neural networks, for example, to identify and categories spoken words and then react appropriately in voice or text that has been synthesized using a generative model like.

4.3 Chatbots and long-term memory

To enhance their long-term memory, chatbots and other generative models can be trained on VDBMs. The vectors can be stored and indexed using a VDBMS, while there are alternative methods. The inability of generative models to recall previous discussions or context is a problem that is currently exacerbated by a number of technological constraints. For instance, while producing a response, a number of models are only able to take into account a small portion of the previous text. Generative models are also limited in the aspect of recalling the previous conversation or context, and at the moment, multiple technical limitations add to the problem. There is no underlying memory of prior interactions in generative models, which respond to the present situation given in the input. When a conversation gets too long or complicated, the capability of the model to refer to previous sections of the conversation reduces. Moreover, the models produced through generation are trained on massive data, and they do not have the capability to differentiate between factual data and individual interactions with the user.

5. LITERATURE REVIEW

Across various areas, semantic search with vector databases has been and is still being thoroughly researched, for example, text and image retrieval, healthcare analytics, domain-specific knowledge discovery, and multimedia search. Here present the methods of analysis and computation that make use of semantic representations and embedding-based similarity retrieval. Earlier work is compiled in a Table as I, which provides details of problems addressed, core techniques, domains of application, advantages, and connection to semantic search with vector databases.

Kirilenko et al. (2022) suggested a novel two-stage method called TSVLoc. It enhances any well-known technique and resolves the place identification challenge as the image retrieval problem. The first model-agnostic step may be used to any modern neural network model, such as HF-Net, NetVLAD, or Patch-NetVLAD, that does not explicitly utilize semantics. In the second stage, they use the Vector Symbolic Architectures (VSA) framework to construct semantic scene representation. Their method uses semantic segmentation of a picture to extract objects and their connections, then uses VSA operations to generate a semantic scene representation [21].

Amin, Mondal and Mathew (2022) suggested a semantic hashing system that learns structural and hierarchical information via hyperbolic metric learning. The hashing network is trained utilizing this data in the form of proxy labels using the proposed novel Structure-Semantic Disagreement (SSD) loss. By forcing the model to learn to hash using both structural and semantic information, SSD produces hash codes that are more reliable and evenly dispersed. The efficacy of the suggested method is demonstrated through tests on many public domain datasets. Furthermore, by encouraging the model to make better use of the structural information, to improve the representation, the suggested SSD loss might also be used to other classification models [22].

Sheng et al. (2021) The mining of vertical knowledge domain content has grown increasingly difficult due to the Internet's rapid expansion of data. This paper proposes a vertical semantic search engine for the electric power metering industry to efficiently collect, arrange, and make use of the extensive corpus of information. The engine outperforms standard keyword-based search engines in terms of accuracy and persistent recall, allowing for relational analysis and semantic understanding. On the one hand, it addresses the shortcomings of traditional general-purpose search engines, namely their lack of specialization and targeting [23].

Ivanova, Zemtsov and Minaev (2020) suggested using a semantic search system to solve the shortcomings of the key-based data discovery technique brought on by the extraction of a significant volume of unnecessary data. A method for locating knowledge in the semantic network during the phase of selecting pertinent pre-knowledge to analyses relational databases and unstructured data using associated open data is presented. The outcomes of the employed algorithm in conventional techniques are compared [24].

Kalmukov and Valova (2019) All of these pictures should be indexed by search engines, nevertheless, in order to boost their effect and public appeal. It is far from easy to create an effective non-textual search engine. It should use contemporary image processing and information retrieval methods to extract, index, and save appropriate metadata from photos so that searching and further processing may be done quickly. Before any image processing is done, the search engine should be able to find and handle the enormous amount of material that is constantly being added to the Internet. Naturally, the World Wide Web is a huge directed cyclic graph that is weightless. Such a building's blind crawling is a never-ending waste of time [25].

Song et al. (2018) The primary semantic composition of distributed representations for query subtopic mining is explored and compared. In particular, they focus on two types of distributed representations that directly describe word sequences of arbitrary length: paragraph vectors and word vector composition. Extensive research is done on the impact of semantic composition approaches and the types of data used in distributed representation learning. A public dataset provided by the Community for Information Access Research and the trials were conducted using the National Institute of Informatics Testbeds [26].

Table 1. Summary of Related work on Semantic Search using Vector Databases

Author(s) & Year	Problem Addressed	Core Techniques	Application Domain	Strengths	Relevance to Semantic Search with Vector Databases
Kirilenko et al. (2022)	Limited semantic understanding in visual place recognition	Two-stage TSVLoc framework, model-agnostic visual	Place recognition, image retrieval	Combines visual and semantic representations; flexible with different neural	Demonstrates enrichment of vector embeddings with semantics, aligning with vector

Author(s) & Year	Problem Addressed	Core Techniques	Application Domain	Strengths	Relevance to Semantic Search with Vector Databases
	and image retrieval systems	descriptors, Vector Symbolic Architectures (VSA), semantic segmentation		models; improved semantic robustness	database–based image retrieval
Amin, Mondal & Mathew (2022)	Inadequate semantic hashing due to lack of hierarchical and structural awareness	Hyperbolic metric learning, semantic hashing, Structure-Semantic Disagreement (SSD) loss	Semantic hashing, information retrieval	Learns both semantic and structural information; uniform and robust hash codes	Supports efficient similarity search and indexing in vector databases through structured semantic representations
Sheng et al. (2021)	Inefficiency of general-purpose search engines in domain-specific knowledge retrieval	Vertical semantic search engine, semantic understanding, relational analysis	Electric power metering domain	Higher accuracy and stable recall; strong domain specialization	Highlights the need for semantic indexing, which can be efficiently implemented using vector databases
Ivanova, Zemtsov & Minaev (2020)	Excessive irrelevant results from keyword-based data discovery	Semantic networks, associated open data, knowledge detection algorithms	Semantic data discovery, knowledge management	Improves relevance; integrates structured and unstructured data	Emphasizes semantic representations that can be embedded and queried using vector databases
Kalmukov & Valova (2019)	Challenges in indexing and searching large-scale non-textual (image) data	Image processing, metadata extraction, intelligent web crawling	Image search engines, multimedia retrieval	Addresses scalability and efficiency in non-textual search	Relevant for vector databases storing image embeddings for semantic image search
Song et al. (2018)	Limited understanding of semantic composition for effective query subtopic mining	Distributed representations, paragraph vectors, word vector composition	Query subtopic mining, text retrieval	Comprehensive comparison of semantic composition strategies	Provides foundational embedding techniques that underpin semantic search in vector databases

6. CONCLUSION AND FUTURE WORK

The semantic search of the vector databases is a great progress in relation to the traditional system of retrieval based on the keywords as it allows retrieving the information in a manner sensitive to meanings and in context. This paper has conducted a review of the basic elements that support semantic search, which are models of Vector representation, transformer-based embedding model, similarity measures, indexing and new vector database designs. Classical and neural embedding models demonstrated to be essential in representation of semantic relations and that the use of vector databases offered scalable and efficient tools of storing and querying high dimensional representations. Similarity

measures, hashing techniques and graph-based indexing techniques are analyzed and their relevance in achieving high retrieval accuracy and performance is emphasized. Moreover, there are several practical applications of semantic search systems in real world such as text and image retrieval, voice recognition, and conversational systems; this variety is enough to show the flexibility and importance of this technology. The paper highlights the increasing significance of the vectors databases in the contemporary information retrieval and outlines room to be filled in the future research through issues of scalability, interpretability and optimization according to domain.

Future studies have the potential to improve vector-based semantic search engines' interpretability and scalability, multimodal embedding, and indexing methods to enable real-time search. Also, the idea of hybrid methods between symbolic knowledge and searching with vectors should be considered which can be further refined to increase the search accuracy and the understanding of the context.

REFERENCES

[1] S. Mao, L.-L. Zhang, and Z.-G. Guan, "An LSTM&Topic-CNN Model for Classification of Online Chinese Medical Questions," *IEEE Access*, vol. 9, pp. 52580–52589, 2021, doi: 10.1109/ACCESS.2021.3070375.

[2] A. Perevalov and A. Both, "Improving Answer Type Classification Quality Through Combined Question Answering Datasets," in *Knowledge Science, Engineering and Management*, Cham, 2021, pp. 191–204.

[3] K. Mao, J. Xu, X. Yao, J. Qiu, K. Chi, and G. Dai, "A Text Classification Model via Multi-Level Semantic Features," *Symmetry (Basel).*, vol. 14, no. 9, 2022, doi: 10.3390/sym14091938.

[4] H. Götzsche, "An Approach to Conceptualisation and Semantic Knowledge: Some Preliminary Observations," *AI*, vol. 3, no. 3, pp. 582–600, 2022, doi: 10.3390/ai3030034.

[5] A. L. Lezama-Sánchez, M. Tovar Vidal, and J. A. Reyes-Ortiz, "An Approach Based on Semantic Relationship Embeddings for Text Classification," *Mathematics*, 2022, doi: 10.3390/math10214161.

[6] U. Krzeszewska, A. Poniszewska-Marańda, and J. Ochelska-Mierzejewska, "Systematic Comparison of Vectorization Methods in Classification Context," *Appl. Sci.*, vol. 12, no. 10, 2022, doi: 10.3390/app12105119.

[7] A. J.-P. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman, "Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports," *Autom. Constr.*, vol. 62, pp. 45–56, Feb. 2016, doi: 10.1016/j.autcon.2015.11.001.

[8] H. Aujla, M. J. C. Crump, M. T. Cook, and R. K. Jamieson, "The Semantic Librarian: A search engine built from vector-space models of semantics," *Behav. Res. Methods*, vol. 51, no. 6, pp. 2405–2418, 2019, doi: 10.3758/s13428-019-01268-4.

[9] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018.

[10] V. K. Garbhapu and P. Bodapati, "A comparative analysis of Latent Semantic analysis and Latent Dirichlet allocation topic modeling methods using Bible data," *INDIAN J. Sci. Technol.*, vol. 13, no. 44, pp. 4474–4482, 2020.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[12] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *Arxiv J.*, 2016.

[13] G. Kovács, P. Alonso, and R. Saini, "Challenges of Hate Speech Detection in Social Media," *SN Comput. Sci.*, vol. 2, no. 2, p. 95, 2021, doi: 10.1007/s42979-021-00457-3.

[14] R. Ribeiro and F. Batista, "Transformer-based Language Models for Semantic Search and Mobile Applications Retrieval," *Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag.*, vol. 1, no. Ic3k, pp. 225–232, 2021, doi: 10.5220/0010657300003064.

[15] S. Garg, "Intelligent Tutoring Systems: The Future of AI-Powered Personalized Learning," *Int. Sci. J. Eng. Manag.*, vol. 01, pp. 1–6, 2022, doi: 10.55041/ISJEM00114.

[16] V. M. L. G. Nerella, "Automated Cross-Platform Database Migration and High Availability Implementation," *Turkish J. Comput. Math. Educ.*, vol. 9, no. 2, pp. 823–835, 2018.

[17] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean Distance Matrices: Essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, Nov. 2015, doi: 10.1109/MSP.2015.2398954.

[18] H. A. H. Hasan, "A Review of Hash Function Types and their Applications," *Wasit J. Comput. Math. Sci.*, vol. 1, pp. 120–139, 2022, doi: 10.31185/wjcm.52.

[19] V. T. Kesavan and B. S. Kumar, "Graph Based Indexing Techniques for Big Data Analytics: A Systematic Survey," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6, pp. 641–647, 2019.

[20] H. P. Kapadia, "Voice and Conversational Interfaces in Banking Web Apps," *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 6, pp. g817–g823, 2021.

[21] D. Kirilenko, A. K. Kovalev, Y. Solomentsev, A. Melekhin, D. A. Yudin, and A. I. Panov, "Vector Symbolic Scene Representation for Semantic Place Recognition," in *2022 International Joint Conference on Neural Networks (IJCNN)*,

2022, pp. 1–8. doi: 10.1109/IJCNN55064.2022.9892761.

[22] F. Amin, A. Mondal, and J. Mathew, "Deep Semantic Hashing with Structure-Semantic Disagreement Correction via Hyperbolic Metric Learning," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, 2022, pp. 1–6. doi: 10.1109/MMSP55362.2022.9948733.

[23] L. Sheng *et al.*, "A Vertical Semantic Search Engine in Electric Power Metering Domain," in *2021 IEEE International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*, 2021, pp. 640–644. doi: 10.1109/ICEEMT52412.2021.9602260.

[24] O. Ivanova, I. Zemtsov, and E. Minaev, "Database Integration Based on the Selection of Preliminary Knowledge Using a Semantic Network," in *2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, 2020, pp. 435–438. doi: 10.1109/SUMMA50634.2020.9280710.

[25] Y. Kalmukov and I. Valova, "Design and development of an automated web crawler used for building image databases," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1553–1558. doi: 10.23919/MIPRO.2019.8756790.

[26] W. Song, Y. Liu, L.-Z. Liu, and H.-S. Wang, "Semantic Composition of Distributed Representations for Query Subtopic Mining," *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 11, pp. 1409–1419, 2018, doi: 10.1631/FITEE.1601476.