https://doi.org/10.15407/intechsys.2026.04.068
UDC 004.274

O.G. IVANCHENKO, Assistant,
Department of Software Systems and Technologies,
Taras Shevchenko National University of Kyiv,
24, Bohdana Havrylyshyna St., Kyiv, 02000, Ukraine
https://orcid.org/0000-0002-8526-8211
duchnov@ukr.net
D.O. KOLOMARENKO, Student,
Department of Software Systems and Technologies,
Taras Shevchenko National University of Kyiv,
24, Bohdana Havrylyshyna St., Kyiv, 02000, Ukraine
https://orcid.org/0009-0006-0213-5111
dmytrii_kolomarenko@knu.ua

SEMANTIC SEARCH OF TEXT DOCUMENTS

Search systems are essential part of computer systems usage, and in particular for information retrival and usage. To address the problem of efectivness of information retrival semantic search systems can be used, as they provide deep searching capabilities due to usage of neural models for perception of text semantic and meaning. This enables a more comprehensive relevancy of search results comparing to classical lexical search methods. This paper address problem of designing a semantic search system that can be used for search in local documents collection. For achieving this goal an architecture with usage of vector database (ChromaDB), neural model execution service (Ollama) and asyncronous task queue service (Celery) is proposed. The developed system provides precision and recall of 56.67% and 94.44% corespondingly, which is on the level with analogs. Also for determing the best in terms of search relevance neural model such models were compared: Doc2Vec, BERT and EmbeddingGemma.

Keywords: *semantic search, embedding models, sentence embedding, system architecture, searching efectivness.*

Introduction

Today's world is filled with a variety of information, such as blog posts, articles in magazines and on Internet resources, books, etc. In such an environment, the problem of effectively finding the necessary data arises, which arises due to the large volume of information. The process of man-

Cite: Ivanchenko O.G., Kolomarenko D.O. Semantic Search of Text Documents. Information Technologies and Systems. 4 (10). 2026. 68—80. https://doi.org/10.15407/intechsys.2026.04.068

© Publisher PH "Akademperiodyka" of the NAS of Ukraine, 2025. This is an Open Access article under the CC BY-NC-ND 4.0 license (https://creativecommons.org/licenses/by-nc-nd/4.0/

ual search for such a volume of information is often routine and time-consuming, and often practically impossible for one person. Accordingly, effective methods of searching among documents are an important topic in the field of information technology. Many existing classical search methods, such as keyword search or full-text search, are not effective enough, since their maximum is taking into account word forms and synonyms. But unlike them, semantic search takes into account not only this, but also the context and meaning of documents and user requests. Therefore, a review of the use of semantic search in text document repositories to build a system for effectively obtaining relevant information is a relevant topic.

The results obtained will be useful in the development of information retrieval systems, where it is the search by context and content that is important, and not the direct correspondence of the keywords to the text of the document query. The methods used to determine the quality of the search by precision and recall can be used to determine the degree of relevance of the results obtained for search methods in comparison with other search methods. Comparative analysis of the effectiveness of semantic search based on EmbeddingGemma and ChromaDB in comparison with classical tools, such as Apache Solr, may be needed when comparing search systems in terms of search efficiency, namely how relevant the search results are to the search query.

In general, the results obtained are the basis for creating effective search systems for stored information using locally deployed neural models for semantic search and privacy of the processed data.

Problem Statement

Search engines are an integral part of the use of computer systems, in particular for searching and using information. To solve the problem of the efficiency of information search, semantic search systems can be used, since they provide the possibility of deep search using neural models of perception of semantics and meaning of the text. This allows you to get a more complete relevance of search results with classical methods of lexical search. Accordingly, it can be argued that the topic of this article is relevant in the modern scientific world. So, the need for semantic search systems is relevant, the need forms the task of the design of such a system, and the production aspect is what technical means are used for such systems, taking into account the importance of high relevance of the obtained search results.

Literature Review

The topic of semantic search is also presented in the scientific literature. The architecture of the document repository was presented in the article [1]. The main problems that this study aims at are the low efficiency of classical keyword search methods and ensuring data confidentiality without significant loss of performance. The proposed solution is based on a

two-level indexing architecture. The first level is an encrypted inverted keyword index that supports search taking into account typos, and the second level is responsible for semantic embeddings. Experimental results confirm the high efficiency of the system. Thus, the recall increased by 22% compared to traditional methods, reaching 90.2%.

Another important problem in semantic search is distinguishing the meaning of words depending on their context. This problem is considered in the article [2]. It proposes a hybrid approach based on a combination of metaheuristic algorithms and deep learning. The main idea is to use a neural network as a surrogate fitness function within the simulated annealing algorithm, i.e. the neural network is used to find the optimal value, which in this case is the closest meaning of the word in a certain context. Experimental evaluation conducted on standard datasets (SensEval and SemEval) demonstrated the high efficiency of the proposed approach. The system achieved precision rates in the range from 74.8% to 84.6%.

As for the search process itself and the process of obtaining relevant information, these problems are also presented in the scientific literature. Yes, the article [3], which proposes an innovative approach to extracting information from descriptions of road accidents using semantic search mechanisms, is an example of using semantic search to effectively obtain relevant information. And the problem of increasing the efficiency of semantic search is investigated in the article [4]. The authors justify the need to move from basic vector search to hybrid architectures that combine linguistic analysis with neural networks. In addition, an important problem in semantic search is determining the proximity of texts. Using the example of searching for books in digital libraries, this problem is considered in the article [5]. This study found that a simple increase in the number of terms or their complete removal from the analysis process leads to a steady decrease in the relevance of search results. Also, a comparative analysis with traditional literal search confirmed the advantages of the semantic approach: while conventional search demonstrates a complete inability to understand context in long queries (efficiency drops to 53.5%), the proposed hybrid model maintains high quality results.

In addition to the actual problems of relevance of search results, an important topic is the use of which technologies are needed. An overview of embedding models is presented in the article [6]. The central part of the study is a structured comparative analysis of the technical and operational characteristics of different models. In conclusion, the authors justify the advantages of using open models as a cost-effective and transparent alternative to proprietary APIs, which is critical for ensuring data confidentiality and independence of research projects. This article can be a basis for developers and researchers when developing their own projects.

Other problems when building semantic search are taking into account the specifics of a particular subject area for which it is being developed. Such a subject area can be the tuning of search for a specific language or some industry. The topic of creating an embedding model for

languages with insufficient training data on the example of Urdu is considered in the article [7]. The study proposes a methodology for creating large-scale vector representations based on the collection and cleaning of a significant amount of unstructured text data. Optimization of embedding models for a specific industry using the example of the biomedical industry was investigated in the article [8]. The main idea of the work is to use unsupervised deep learning methods that can capture context and semantic relationships with great efficiency without the need for prior manual data markup. The study found that sentence embedding models significantly outperform approaches based only on single-word embeddings, as they better take into account the structure and context of utterances. The work proves that unsupervised neural networks are a powerful tool for natural language processing in subject areas with a large amount of specific terminology. The effectiveness of specialization of embedding models for a specific industry is proven by the study in the article [9]. The results of experiments conducted on large scientific datasets demonstrate the significant advantage of transform models, especially specialized solutions such as SciBERT, in the tasks of interest modeling and content recommendation. The study confirms that the use of semantic encoders allows to significantly increase the accuracy of information search compared to classical statistical methods.

Another important problem in the field of semantic search is the presence of bias and insufficient representativeness of certain aspects in the training data and the neural networks themselves. This problem is investigated in the article [10] using the example of the issue of gender inequality in sentence encoders. The authors note that although embedding models are the basis of high search efficiency, they can also have biases associated with the presence of social stereotypes in the training data. To solve this problem, a new flexible metric is proposed — bias score, which allows to quantitatively assess the level of gender bias at the level of entire sentences. The practical significance of the study is confirmed by the development of two methods for improving the fairness of language models: removing the most stereotypical records from the training data set and the data augmentation strategy by replacing gender markers with opposite ones.

So, as can be understood from the completeness of the presentation of the topic of semantic search and document repositories in the reviewed articles, it can be understood that its issues and problems are important and interesting for society. Thus, the problems considered are both architecture at a general level and problems of using specific technologies and methods. An important aspect when using semantic search in a specific subject area, as can be understood, is the optimization and tuning of semantic search, namely, embedding models for a specific industry, which is achieved by selecting appropriate training data. Also, the correct selection of training data and their processing are important for ensuring the quality and efficiency of the search by cleaning the training data from noise and preventing further bias of the embedding models. In addition,

the reviewed studies demonstrate a significant advantage of the semantic approach over classical search methods, such as keyword search, in those search tasks where the relevance of documents is determined by semantic proximity.

Also, since the reviewed publications did not consider a complete practical software system for semantic search in a text document repository, this study can help in the design and creation of document repository systems with the presence of their search functionality.

The Purpose of the Paper

The aim of the research is the software implementation of a text document repository using semantic search to increase the recall of the retrieval.

Within the chosen research direction, both a comparative analysis of semantic search methods will be carried out to clarify the general principles of their operation and identify the most optimal method, and the design of a software system for storing documents and searching among them to consider the practical aspect of using and implementing such systems.

Achieving the goal includes solving the following tasks:

- conduct a comparative analysis of existing semantic search methods and justify the choice of the most effective approach to maximize the recall of text document retrieval;
- design architectures and develop a software implementation of a document repository that integrates selected algorithms for semantic processing and data indexing;
- test the developed system and evaluate the search results to confirm the improvement of the recall of the retrieval.

Materials and Methods

If we consider the methods and tools used for semantic search, they are based on vector search and sentence embedding — a set of natural language processing methods where sentences are mapped to vectors of real numbers. Here, we will consider such methods and tools based on neural networks as Doc2Vec, Google BERT and EmbeddingGemma.

Doc2Vec is based on the Word2Vec methodology. Doc2Vec is a natural language processing (NLP) approach that allows for vector representation of larger blocks of text, such as sentences, paragraphs or entire documents. This method was developed by Le and Mikolov and is also known as the paragraph vector.

Doc2Vec is based on the same principles as Word2Vec, which uses a neural network model to learn word associations from a large corpus of text and generates a multidimensional space (typically several hundred dimensions), where each unique word in the corpus is assigned a corresponding vector in this space. Word vectors are arranged in a vector space in such a way that words with similar contexts are close to each other.

Doc2Vec, unlike Word2Vec, is designed to capture the context of entire texts or paragraphs, rather than just the local context of individual words. It achieves this by introducing a new vector, the document vector (or paragraph vector), which is trained together with the word vectors.

Doc2Vec can capture semantic relationships between documents, allowing for tasks such as document similarity and clustering. Moreover, documents are represented as dense vectors in a continuous space, making them suitable for various machine learning algorithms and thus allowing for efficient processing of large datasets.

Although Doc2Vec is very powerful, it also has some limitations. The quality of the vectors is highly dependent on the size and representativeness of the training corpus. It can be computationally intensive, requiring significant hardware resources for large datasets. Additionally, like many machine learning models, it may require tuning a number of hyperparameters to optimize performance for specific tasks [11].

BERT is a deep learning language model designed to improve the performance of natural language processing (NLP) tasks. It is known for its ability to take context into account by analyzing the relationships between words in a sentence bidirectionally. It was introduced by Google researchers in a 2018 paper titled "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Since then, BERT has been refined for use in a variety of fields, including biology, data science, and medicine.

BERT stands for Bidirectional Encoder Representations from Transformers. Bidirectionality means that each token — the elementary part of a document used for processing — is simultaneously associated with every other token in the input data.

Encoders are components of a neural network that transform input data into representations that are easier for machine learning algorithms to process. After an encoder reads the input text, it generates a hidden state vector. Hidden state vectors are lists of values and internal parameters that provide additional context. This packaged representation of the information is then passed to a transformer.

A transformer uses the above information to infer patterns or make predictions. A transformer is a deep learning architecture that transforms input data into another type of output data. Almost all NLP applications use transformers. Transformers typically consist of an encoder and a decoder. However, BERT uses only the encoding part of the transformer [12].

EmbeddingGemma is a multilingual text embedding model with 308 million parameters based on Gemma 3. It is optimized for use on everyday devices such as phones, laptops, and tablets. The model creates numerical representations of text for use in subsequent tasks such as information retrieval, semantic similarity search, classification, and clustering [13].

EmbeddingGemma uses an encoder-only model adapted from a pretrained decoder-only Gemma 3 model with 300 million parameters. Specifically, the Gemma 3 model was adapted into an encoder-decoder model according to the T5Gemma recipe, and then EmbeddingGemma was

initialized with the encoder of this encoder-decoder model. Thus, EmbeddingGemma can leverage Gemma 3's vast world knowledge. Using encoder-decoder adaptation further ensures that the encoder can produce expressive representations from the start [14].

EmbeddingGemma includes the following key features. This model has a broad understanding of linguistic data and has been trained on over 100 languages. It can be tuned to output sizes from 768 to 128 for speed-size trade-offs using Matryoshka Representation Learning (MRL). This model also has a substantial input context of 2000 tokens for processing text data and documents. In terms of performance, the model uses less than 200 MB of RAM with quantization, and generative embedding occurs in less than 22 ms on EdgeTPU [13].

To select an effective and optimal method for semantic search, a research method such as comparative analysis is used. During the analysis, a quantitative comparison was made by the relevance of the received documents, namely, a comparison of the precision and recall of the search results for the given topics. To study the effectiveness of the developed system, both a qualitative comparison was used to prove the quality of the functionality provided by the system, and a quantitative comparison was used to prove the relevance of the documents provided based on the query with analogues in terms of precision and recall of the results obtained.

Results

Comparasion of the Selected Embedding Models.

The above-described search methods were compared in terms of precision and recall, namely Doc2Vec, BERT, and EmbeddingGemma. Precision is the proportion of relevant samples among those selected, and recall is the proportion of selected relevant samples among all truly relevant ones. The following interpretation can be given for precision and recall: precision shows the percentage that a randomly selected document from

Table 1. **Comparison of embedding models**

Theme	Model	Precision	Recall
Society & Culture	Doc2Vec	0.1100	0.0183
	BERT	0.0900	0.0150
	EmbeddingGemma	0.2200	0.0367
Science & Mathematics	Doc2Vec	0.0400	0.0067
	BERT	0.0500	0.0083
	EmbeddingGemma	0.1750	0.0292
Health	Doc2Vec	0.0600	0.0100
	BERT	0.0700	0.0117
	EmbeddingGemma	0.4700	0.7830
Computers & Internet	Doc2Vec	0.0400	0.0067
	BERT	0.0850	0.0142
	EmbeddingGemma	0.5500	0.0917

the set of found documents is relevant, and recall shows the percentage that a randomly selected document from the set of all relevant documents will be found.

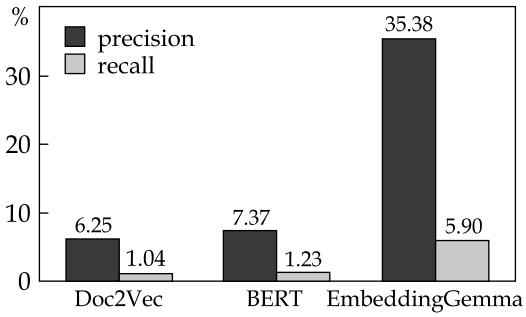

Fig. 1. Average precision and recall values for the considered models

For comparison, several questions were asked for each topic, and the search results obtained were checked for belonging to the searched topic: if the topic of the document matches, then it is counted among the relevant results, if not, then among the irrelevant ones. Categorized texts The pietrolesci/yahoo_answers_topics dataset [13] was used for comparison.

To compare the described methods, a comparison of how well the methods classify texts was used. This approach was used to make it easier to automate comparisons and avoid manual selection of relevant texts.

The results of the comparison are shown in Table 1 below.

A comparison of the average precision and recall between the models is shown in Figure 1.

Design of the Developed System.

The system consists of the following parts: the client part (frontend) and the server part (backend), which consists of the application itself, the neural model execution service, and the vector database.

The frontend is responsible for the end user interface, which provides users with functionality for managing files and searching among them.

The application itself is responsible for the business logic of the backend, i.e. the actual search, outputting results, initiating and tracking the status of processing the uploaded document, and is also responsible for the stability of processing long documents.

The neural model execution service executes a sentence embedding model, which generates vectors corresponding to the documents or their fragments sent for processing. These vectors reflect the meaning of the document and the semantic relationships between its parts.

The vector database stores sentence embeddings, i.e. the corresponding vectors of the processed documents, and provides the functionality for searching for documents by the proximity of their vectors and the query vector.

The general algorithm of the semantic search system is based on the interaction of the client part, the task queue and the vector database. The main stages of the system operation can be presented as follows.

The user uploads a file via the /files/upload API endpoint, implemented on FastAPI. The system checks the document format for compliance with the list of supported types and saves it in local storage.

Since text indexing is a resource-intensive process, the main application does not process the file immediately, but creates a document pro-

cessing task via Celery. Information about the task is transmitted to the Redis message broker, and the user instantly receives a task identifier for status tracking.

Next, the task extracts text content from the uploaded file using the textxtract library. After this the semchunk library breaks the text into fragments (chunks), which allows you to work effectively with large documents without exceeding the limit of the neural model context.

After preparing the text, embeddings are generated. For each text fragment, the system contacts the Ollama service, which uses the EmbeddingGemma model to transform the text into vector representations (numeric vectors). These vectors reflect the semantic meaning of the fragments in a multidimensional space.

The resulting vectors, along with metadata (file name), are stored in the ChromaDB collection. This completes the process of preparing the document for search, making it available for semantic analysis.

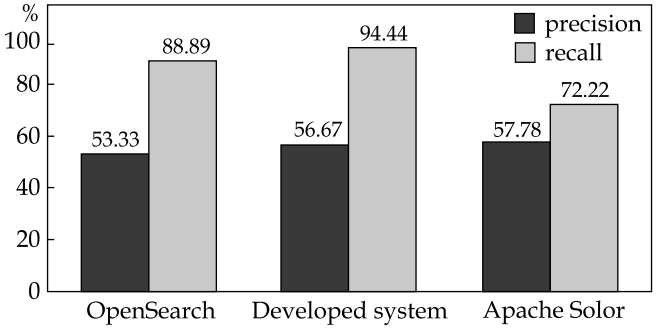

Fig. 2. Average precision and recall values for the considered systems

Table 2. **Search relevance testing results**

Theme	Request	System	Precision	Recall
bugs_and_science	what species of bugs exist	Developed	0.4000	0.6667
		OpenSearch	0.4000	0.6667
		Apache Solr	0.4000	0.6667
bugs_and_science	biological information about bugs and beetles	Developed	0.6000	1.0000
		OpenSearch	0.6000	1.0000
		Apache Solr	0.6667	0.6667
Linux	how is linux present in Europe	Developed	0.6000	1.0000
		OpenSearch	0.6000	1.0000
		Apache Solr	0.4000	0.6667
Linux	structure of linux os	Developed	0.6000	1.0000
		OpenSearch	0.4000	0.6667
		Apache Solr	0.4000	0.6667
Music	famous performers in the music industry	Developed	0.6000	1.0000
		OpenSearch	0.6000	1.0000
		Apache Solr	0.6000	1.0000
Music	romantic songs about love	Developed	0.6000	1.0000
		OpenSearch	0.6000	1.0000
		Apache Solr	1.0000	0.6667

Also, when receiving a search query via the /search method, the system vectorizes this query using the same model in Ollama. ChromaDB then performs a mathematical comparison of the query vector with the vectors of the stored documents and returns the most relevant results based on semantic proximity, rather than a simple match of words.

Search Relevance Testing for the Developed System.

The following methodology was used to test the relevance of search results. A set of topics was given, and for each topic, related documents were created (the texts were taken from the English-language Wikipedia). Also, for each topic, "query-document" pairs were defined, where the query is the actual query string to the search engine, and the document is the file names of documents relevant to the query, which is its identifier. The number of relevant documents for each query is 3 documents, and the maximum number of results obtained is 5.

The testing was performed by comparing the results of the developed system, OpenSearch, which was used as a semantic search platform, and Apache Solr for full-text search. For OpenSearch, Ollama was used as a model provider, namely EmbeddingGemma, to generate embeddings. For Apache Solr, the Extended DisMax (eDisMax) Query Parser is used as the query parser, allowing for the processing of stop words and synonyms, and the percentage of word matches is set to 50%, which adds relative flexibility to determine the relevance of documents.

Accordingly, the test is how well the designed system provides relevant features compared to another widely used system using the same facilities to generate sentence embeddings. The test results for the considered systems are shown in Table 2 below.

The chart in Fig. 2 presents the average values of precison and recall for the developed system and the considered systems for comparison.

Discussion of the Results

Regarding the results of the "Comparasion of the Selected Embedding Models" section, the low recall values can be explained by the fact that the number of relevant results is taken as the number of texts on the corresponding topic in general, and not as the number of search results that are issued. That is, when calculating recall, the denominator takes into account the number of all texts on this topic, and not the maximum possible number of documents that can be issued on this topic, which is less than the previous number.

The significant difference between the values of the Health and Computers & Internet topics and other topics for EmbeddingGemma can be explained by the fact that these topics are represented in the training datasets for this model and also by the fact that they have clear terms and semantic connections in these topics are quite dense.

Also, as can be seen from Fig. 1, EmbeddingGemma has the best precision and recall values, while Doc2Vec has the worst. This can be ex-

plained by several factors. First, Doc2Vec is limited to the dataset on which the neural network was trained, which is the dataset used. Second, BERT and EmbeddingGemma use transformers and an attention mechanism that allow building higher-quality semantic connections. Third, EmbeddingGemma is trained on a larger amount of data and is built not from scratch, but on the basis of the Gemma family of neural models, i.e. it has a more advanced structure than BERT.

The results of the section "Design of the Developed System", namely the designed system model, guarantee, in addition to the necessary functionality for a document repository with support for semantic search, the system's resilience to failures and data preservation. The use of Docker containerization technology allows the system to be deployed in an isolated environment with a clear distribution of resources between AI models and the server part.

Regarding the results of the section "Search Relevance Testing for the Developed System", as can be seen from the diagram in Figure 2, the search accuracy of the developed system is not inferior to the considered analogues, and in recall it even exceeds them, which shows that the developed system will find a relevant document with a higher probability among all documents relevant to a specific query.

Conclusions

Several semantic search methods were analyzed, namely Doc2Vec, BERT, and EmbeddingGemma. All three methods are based on document vectorization — sentence embeddings, but achieve this in different ways. Doc2Vec uses Word2Vec to vectorize individual words and a document vector to obtain a general document vector. The main feature of BERT is the use of bidirectional encoders — that is, each word during processing is simultaneously associated with all other words in the sentence. EmbeddingGemma, on the other hand, uses an encoder-decoder model and is built on the basis of the large Gemma 3 language model, as well as the support of this model for over 100 languages. Based on this analysis and comparison of methods in terms of accuracy and completeness, it was shown that the most effective method in terms of the relevance of the results obtained is EmbeddingGemma. This result can be explained by the fact that EmbeddingGemma, unlike Doc2Vec, is based on a more advanced architecture using transformers and an encoder-decoder, and that this model, unlike BERT, due to its construction based on Gemma 3, is trained on a larger amount of data.

The use of EmbeddingGemma together with Ollama for its execution, ChromaDB for storing document embeddings and their search, and dividing input documents into fragments for processing large texts ensured the creation of a workable, as well as reliable and efficient system. In addition, it is also worth describing the general algorithm for EmbeddingGemma. It consists of using a series of carefully designed steps to create a concise

vector representation: transformer processing — uses bidirectional attention to understand the meaning and context of each word and generate appropriate embeddings, merging — combining all word embeddings into a single embedding, compactification — performing a linear projection into a higher-dimensional vector and reducing the resulting vector afterwards, and normalization — allows for efficient similarity comparisons.

The effectiveness and functionality of the developed system was demonstrated by comparing the developed system with analogues, which proved that it is not inferior to analogues in terms of search efficiency.

DECLARATION

Author Contributions. *Tetiana Holub: Conceptualization, Software, Investigation, Data Curation, Writing – Original Draft, Visualization. Ivanchenko Oleksii: Methodology, Supervision, Writing – Review & Editing.*

AI and Automated Tools Usage. *During the preparation of this manuscript, the authors used Gemini (Google) for summarazing papers considered in the literature review section. Also Google Translate was used for translating from inital draft in Ukrainian to English.*

Conflict of Interest. *The authors declare no conflict of interest.*

Grant or Financial Support. *This research received no external funding.*

REFERENCES

1. Mohammed H.K., Indrani B. Secure Semantic Retrieval over Encrypted Cloud Data: A Dual-Index System with Lightweight Fuzzy Encryption and BERT-Based Semantic Indexing. *Diyala Journal of Engineering Sciences,* 2026, Vol.19 (1), 160–180. https://doi.org/10.24237/djes.2026.19112
2. Nodehi A.K., Charkari N.M. A metaheuristic with a neural surrogate function for Word Sense Disambiguation. *Machine Learning with Applications,* 2022, Vol. 9, Article 100369. https://doi.org/10.1016/j.mlwa.2022.100369
3. Arteaga C., Ray U., Park J., Paz A. Semantic search approach for information extraction from traffic crash narratives. *Transportation Research Interdisciplinary Perspectives,* 2026, Vol. 36, Article 101888. https://doi.org/10.1016/j.trip.2026.101888
4. Mageshwari V., Shrijith J. Linguistic-Neural Hybrid Architectures for Semantic Search: Optimizing Recall with POS Filtering and Query Expansion. *IEEE Access*, 2026, Vol. 14, 28564–28572. https://doi.org/10.1109/access.2026.3663716
5. Daffa M.R., Yaqin M.A. Implementation of Semantic Similarity for Book Search in Digital Library. *Electrics, Electronics, Communications, Controls, Informatics, Systems,* 2025, Vol. 19 (3), 190–196. https://doi.org/10.21776/jeeccis.v19i3.1796
6. Abimbola J.O., Kuaban G.S., Ajayi S.A. Open-Source Embedding Models: A Comprehensive Survey of Techniques, Benchmarks, and Applications. *IEEE Access,* 2026, Vol. 14, 41284–41302. https://doi.org/10.1109/access.2026.3670100
7. Nazir S., Asif M., Sahi S. A., Ahmad S., Ghadi Ya.Ya., Aziz M.H. Toward the Development of Large-Scale Word Embedding for Low-Resourced Language. *EEE Access,* 2022, Vol. 10, 54091–54097. https://doi.org/10.1109/access.2022.3173259
8. Blagec, K., Xu, H., Agibetov, A. et al. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. *BMC Bioinformatics,* 2019, Vol. 20, Article 178. https://doi.org/10.1186/s12859-019-2789-2
9. Guesmi M., Chatti M.A., Kadhim L., Joarder S., Ain Q.Ul. Semantic Interest Modeling and Content-Based Scientific Publication Recommendation Using Word Embeddings and Sentence Encoders. *Multimodal Technologies and Interaction,* 2023, Vol. 7 (9), Article 91. https://doi.org/10.3390/mti7090091
10. Dolci T., Azzalini F., Tanelli M. Improving Gender-Related Fairness in Sentence Encoders: A Semantics-Based Approach. *Data Science and Engineering,* 2023, Vol. 8, 177–195. https://doi.org/10.1007/s41019-023-00211-0
11. What is Doc2Vec? educative.io. URL: https://www.educative.io/answers/what-is-doc2vec [Accessed 23 Jan.2026]
12. What Is the BERT Model and How Does It Work? Coursera. URL: https://www.coursera.org/articles/bert-model [Accessed 23 Jan.2026]

13. EmbeddingGemma model overview. Google AI for Developers. URL: https://ai.google.dev/gemma/docs/embeddinggemma [Accessed 23 Jan.2026]
14. Vera H.S. et al.EmbeddingGemma: Powerful and Lightweight Text Representations. ArXiv, 2509.20354. https://doi.org/10.48550/arXiv.2509.20354

Received 15.06.2026
Accepted 09.09.2026
Published XX.XX.2026

О.Г. ІВАНЧЕНКО, асистент,
каф. програмних систем і технологій,
Київський національний університет імені Тараса Шевченка,
вул. Богдана Гаврилишина, 24, Київ, 02000, Україна
https://orcid.org/0000-0002-8526-8211
duchnov@ukr.net
Д.О. КОЛОМАРЕНКО, студент,
каф. програмних систем і технологій,
Київський національний університет імені Тараса Шевченка,
вул. Богдана Гаврилишина, 24, Київ, 02000, Україна
https://orcid.org/0009-0006-0213-5111
dmytrii_kolomarenko@knu.ua

СЕМАНТИЧНИЙ ПОШУК ТЕКСТОВИХ ДОКУМЕНТІВ

Вступ. Пошукові системи є невід'ємною частиною використання комп'ютерних систем, зокрема для пошуку та використання інформації. Для вирішення проблеми ефективності пошуку інформації можна використовувати системи семантичного пошуку, оскільки вони забезпечують можливості глибокого пошуку завдяки використанню нейронних моделей для сприйняття семантики та значення тексту. Це дає змогу отримати більш повну релевантність результатів пошуку порівняно з класичними методами лексичного пошуку. Відповідно можна стверджувати, що тематика цієї статті є актуальною в сьогоднішньому насиченому інформацією світі.

Мета статті. Отже потреба в семантичних пошукових системах є актульною, ця потреба формує задачу проєктування таких систем, і також важливим аспектом є які саме технічні засоби використовувати для таких систем з урахуванням важливості високої релевантності отримуваних пошукових результатів.

У межах цієї роботи було спочатку виконанно порівняння ефективності нейронних моделей вбудовувань, а саме *Doc2Vec, BERT* та *EmbeddingGemma*. Це порівняння проводилося за тим, як добре певна модель класифікує текст до певної теми, і для порівняння були використанні показники точності та повноти. Після цього було спроєктовано та розроблено власну систему семантичного пошуку, яка вже для тестування її ефективності порівнювалася з іншими пошуковими системами: *OpenSearch* як система семантичного пошуку та *Apache Solr* як система повнотекстового пошуку. Це порівняння здійснювалося вже за тим, скільки релевантних до пошукового запиту документів для кожної пошукової системи є з усіх отриманих, і порівняння теж проводилося за точністю та повнотою.

Результати. Порівняння розглянутих моделей вбудовувань показало, що найефективнішою моделлю є *EmbeddingGemma* як і за точністю — 35,38 %, так і за повнотою — 5,9 %. А порівняння розробленої системи з вище вказаними аналогами показало працездатність і ефективність розробленої системи, оскільки показники точності й повноти є на рівні з *OpenSearch* і вище ніж у *Apache Solr*, що доводить більшу якість пошукових результатів для семантичних пошукових систем ніж для класичних лексичних систем.

Висновки. Розроблена система показала, що семантичні пошукові системи з пропонованою спроєктованою архітектурою з використанням моделей вбудовувань на основі великих мовних моделей дають високі ефективність пошуку й релевантність реузльтатів пошуку.

Ключові слова: *семантичний пошук, моделі вбудовування, вбудовування речень, архітектура системи, ефективність пошуку.*