STUDENT SESSION

# TWO-STEP SCIENTIFIC LITERATURE SEARCH BY TITLE AND ABSTRACT EMBEDDINGS

Mikhail Batura[1]*,
[0009-0007-4411-4906]

Timea Bezdan[2]
[0000-0001-6938-6974]

[1]Student,
Singidunum University,
Belgrade, Serbia

[2]Singidunum University,
Belgrade, Serbia

Abstract:

This paper presents the design and implementation of a 2-step similarity search system for scientific literature using their title and abstract embeddings. One of the approaches to semantic similarity is comparing text embeddings – numerical representations of text, obtained with transformer models. Usually, full-body text embeddings are used for search. This paper aims to find a middle ground between the two methods by using paper abstracts. Using a small dataset of papers grouped by topic, the optimal value for filtering abstract embeddings by cosine similarity was found. Another point of study was to find how the language of abstracts influences the similarity metric. For Serbian English papers, the language impact wasn't significant on the similarity search. A complete solution, including a web scraper, storage and a local embedder, was created, which showed positive results during the test run.

Keywords:

Similarity Search, Text Embeddings, Transformers, Semantic Retrieval, Web Scraping.

## INTRODUCTION

Literature review is a time-consuming process, where the researcher needs to read many scientific papers to find the relevant ones. It is further complicated by the search function, which in many journals works by only matching keywords without any additional context. Similarity search is an alternative solution to word matching. Similarity search consists of finding the closest data based on its numerical representation [1]. To get a numerical representation of a text, called text embedding, sentence transformer models could be used. Then the distance between embeddings can be calculated to represent how semantically close those texts are. In the case of science literature, one way to perform a search is to use title embeddings [2]. Another approach is to use the whole text but split it into smaller chunks [3]. This paper approaches the problem from a different angle and uses abstract embeddings for similarity search.

Correspondence:
Mikhail Batura

e-mail:
mikhail.batura.22@singimail.rs



DOI: 10.15308/Sinteza-2026-585-590

The objective of this study is to develop a local literature retrieval solution. A web scraper is employed to obtain paper titles and abstracts. A local embedding model converts retrieved text data into vectors. Data storage is implemented to save time and reduce internet traffic. Similarity algorithm finds the most relevant embeddings. The source code used in this study can be found in the following GitHub repository (github.com/m-batura/scientific-literature-search).

The contribution of the current article includes the following:

- Analysis of how well abstract embeddings work for science literature similarity search
- Findings on how different languages of abstracts (English-Serbian) affect cross-lingual similarity

## 2. RELATED WORK

Measuring semantic similarity between text data remains an open problem in NLP, as the versatility of natural language complicates rule-based approaches. To address this, various methods have been proposed over the years. Survey [1] covers the evolution of those methods from traditional kernel-based techniques to transformer-based models, categorising them into knowledge-based, corpus-based, deep neural network–based, and hybrid approaches. The knowledge-based methods analyse the underlying databases to disambiguate synonyms; corpus-based methods are versatile, and they can be used across languages. Deep neural network–based systems are computationally expensive but provide better results.

Retrieval-Augmented Generation or RAG is a modern technique that uses semantic similarity to enable Large Language Models (LLMs) to incorporate external information. The paper about automating systematic literature review (SLR) with RAG [4] covers the retrieval part of RAG in detail, discusses the difference between sparse and dense retrieval models, their efficiency, pros and cons. Sparse retrievers utilise keyword matching to assess how relevant each document is to a user. Their downside is that, beyond word matches, they have limited capacity to capture semantic similarity. Dense retrievers, on the other hand, while not being interpretable, transform text data into vector embeddings. That allows dense retriever models to identify relevant documents even when matching keywords aren't present.

RAG implementation in another study [5] uses usual chunking and embedding using transformer models, but on top of semantic similarity, it implements graph-based retrieval using PageRank algorithm and adds reranker model as final layer in retrieval.

In [2], the authors build a recommender system using a hybrid approach that combines citation networks and similarity search. Titles and abstracts were used to extract features, and for that purpose, titles showed better results. Study [6] relies on an already established citation database. They used titles and abstracts baked together as well as full body text. Their findings show that title-abstract combination performance was dependent on the model used and that using full body text is computationally expensive.

Study [7] explores a completely different approach to similarity search. They build a similarity network by indexing citations inside one paper based on their closeness.

## 3. METHODOLOGY

### 3.1. EMBEDDINGS

Text embeddings are a numerical representation of written data. They consist of vectors of real numbers. Vectors are constructed in a way that texts with similar meaning have near vectors. To transform text data into embeddings, text embedding models are used. Programming language Python features a library called Sentence-Transformers (SBERT) [8] that can be used to interact with dense retrieval models. The library features many models, both closed and open source. Model Alibaba-NLP/gte-multilingual-base [9] was chosen for the experimental part of research because it is compact, multilingual, open source and has a high rank on the MTEB leaderboard. To define how close two vectors are, several metrics, shown in Equation 1, can be used to calculate the difference: Euclidian distance (1), Cosine distance (2) and Dot product (3) [10, 11]. While all three metrics could be used, Dot product and Euclidian distance depend on vector length, so Cosine similarity was chosen since the selected model doesn't have built-in vector normalization.

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_N - b_N)^2} \quad (1)$$

$$\cos\theta = \frac{v_1 \cdot v_2}{|v_1||v_2|} \quad (2)$$

$$a_1b_1 + a_2b_2 + \ldots + a_nb_n \quad (3)$$

**Equation 1.** Metrics for calculating vector similarity

To find out if the paper is relevant or not by looking at its similarity metric, a threshold value should be found. When a researcher looks for relevant literature, they already have a main idea or a short description of what they are looking for. This idea and description are

similar in length and semantic load to the title and abstract of a paper. A collection of groups of papers, where each group represents one topic, can be collected. Then, the titles and abstracts of those papers could be turned into embeddings. Finding the difference between those embeddings can show how effective using embeddings for semantic similarity is. Approximate threshold value for filtering relevant papers could be obtained in this process as well.

### 3.2. WEB SCRAPING

In this work, conference proceedings of the Sinteza Conference on Information Technology, Computer Science, and Data Science (portal.sinteza.singidunum.ac.rs) are used as a source for papers. Conference proceedings cover multiple topics of information technology and contain volumes from 2014 to 2026. The first step of the similarity search is obtaining paper titles and abstracts. This can be done with web scrapers. One of the most popular Python libraries for web scraping is BeautifulSoup4 [12]. To use this library journal, the structure should be identified. The number of requests should also be taken into consideration, since sending too many requests can result in a ban.

## 4. EXPERIMENTAL SETUP

### 4.1. SIMILARITY METRIC CALCULATION

A small dataset was created for finding the threshold value of cosine similarity. It consists of 3 groups of pages, each group shares the same topic, and all papers are related to Data Science. Group 1 is related Retrieval-Augmented Generation, group 2 discusses Web Scraping, and group 3 talks about the use of AI in language learning. 3 papers, one for each topic, were found on Sinteza. Then, for each, 4 more papers, linked in their cited works, were added. Titles and abstracts of those papers were collected and turned into embeddings. Then, for the paper from the first group, the following vector search was performed: title with title, abstract with title and abstract with abstract.

The results of the experiment can be seen in Table 1. When compared with themselves, titles are too short to result in a clear separation by group, so for accurate retrieval, a more detailed description is required. Comparing longer abstracts to titles, however, shows clearer group separation, so it can be used as a first step in retrieval. Abstract to abstract similarity shows minimal

**Table 1.** Results of the experiment for finding similarity cutoff

| Title to title similarity | | Abstract to title similarity | | Abstract to abstract similarity | |
|---|---|---|---|---|---|
| Group | Cosine Similarity | Group | Cosine Similarity | Group | Cosine Similarity |
| 1 | 1.000000 | 1 | 0.837151 | 1 | 1.000000 |
| 1 | 0.850234 | 1 | 0.835407 | 1 | 0.756685 |
| 2 | 0.770351 | 1 | 0.733402 | 1 | 0.718145 |
| 1 | 0.751242 | 2 | 0.713097 | 1 | 0.715886 |
| 2 | 0.706568 | 2 | 0.702307 | 2 | 0.702988 |
| 2 | 0.702756 | 2 | 0.696455 | 1 | 0.694153 |
| 1 | 0.679906 | 1 | 0.692194 | 2 | 0.687956 |
| 2 | 0.679779 | 2 | 0.666090 | 2 | 0.684685 |
| 2 | 0.635886 | 2 | 0.646910 | 2 | 0.655059 |
| 3 | 0.624370 | 3 | 0.599086 | 2 | 0.650961 |
| 3 | 0.616502 | 3 | 0.583786 | 3 | 0.584424 |
| 3 | 0.614821 | 3 | 0.582498 | 3 | 0.571730 |
| 3 | 0.601823 | 3 | 0.581880 | 3 | 0.557478 |
| 1 | 0.591715 | 1 | 0.580002 | 3 | 0.526542 |
| 3 | 0.498876 | 3 | 0.498126 | 3 | 0.474363 |

mixing between groups and can be used as is for the final stage of retrieval. Based on the results of the experiment, a value of 0.69 is chosen for both the title and abstract value cutoffs.

Another aspect of the similarity search that should be taken into consideration is the language of the text. Alibaba-NLP/gte-multilingual-base is trained on multilingual data, including English and Serbian, so the model can work on those texts, but the similarity of embeddings of the semantically identical texts in different languages needs to be studied. To find out how language affects the similarity metric, a small dataset of parallel abstracts in English and Serbian was created. 10 pairs of abstracts were taken from the InfoM journal (infom.org.rs/index.php/infom).

From the results in Table 2, the following conclusions were made:

- The difference between the original text and its translations (entry 1) is significant because of the non-linearity of the cosine distance.
- Text with faithful translation (entries 3, 5, 6, 7) shows minimal difference between English and Serbian text.
- When translation was not accurate (entries 2, 4, 9), i.e. some parts were omitted in either text, the distance reflected those changes.
- In conclusion, the current model should work with both English and Serbian papers without adjusting the cutoff value per language.

### 4.2. DATA SCRAPING, STRUCTURE AND STORAGE

For collecting papers from the Sinteza web-resource, a scraper was developed. Links and titles of each paper are saved to a table. Then, the title embeddings are added, and the result is converted to a Parquet file for subsequent usage.

Search goes in two stages: first by title embeddings, then by abstract embeddings. For the first stage, after the table with titles and their embeddings has been loaded, the embedding of the user input is constructed. Cosine distance is calculated between the input embedding and title embeddings; results with a distance less than the threshold value (0.69) are returned. Before the second stage of search, for every paper, its abstract is retrieved by the link and added to the table. Embeddings of abstracts are calculated and added there as well. For the second stage, the input embedding is compared to abstract embeddings. After obtaining the distance between them and adding it to the table, rows are once again cut off by the threshold value (0.69), and the results are ranked by distance in ascending order.

## 5. EVALUATION AND RESULTS

For the demonstration abstract of paper [6] was used. Filtering by threshold value was turned off for better observation.

input = 'This study examines Retrieval-Augmented Generation (RAG) in large language models (LLMs) and their significant application for undertaking systematic literature reviews (SLRs). RAG-based LLMs can potentially automate tasks like data extraction, summarization, and trend identification. However, while LLMs are exceptionally proficient in generating human-like text and interpreting complex linguistic nuances, their dependence on static, pre-trained knowledge can result in inaccuracies and hallucinations. RAG mitigates these limitations by integrating LLMs' generative capabilities...'

Result of the program completion, seen in Listing 1 shows following:

- The first result is the paper on which the abstract search was performed, which passed the title filter, and the abstract similarity is 1, because they are identical,

**Table 2.** Results of the cross-lingual similarity experiment

| # | similarity_en | similarity_rs | difference |
|---|---|---|---|
| 1 | 1.000000 | 0.809972 | 0.190028 |
| 2 | 0.614419 | 0.546868 | 0.067551 |
| 3 | 0.501396 | 0.484479 | 0.016917 |
| 4 | 0.524313 | 0.571315 | -0.047002 |
| 5 | 0.554649 | 0.545670 | 0.008979 |
| 6 | 0.546874 | 0.563902 | -0.017027 |
| 7 | 0.619606 | 0.604596 | 0.015011 |
| 8 | 0.448108 | 0.474018 | -0.025910 |
| 9 | 0.456961 | 0.520029 | -0.063068 |

- The second result passed both filters, and its topic is a framework for semantic search using RAG, which does match the user input,
- The third result shares some topics with the input, like using LLMs for generations and retrieving data, but its scope differs from the input's abstract. It doesn't pass either filter, however, and as such won't be recommended for use,
- The rest of the results follow the same pattern of having similar methods of topic, but not matching the scope of the input study, and none of them passed the filter.

Overall, the search was successful; however, the small size of the Sinteza corpus (1068 papers) does not allow for a bigger picture.

## 6. CONCLUSION

The study shows that using embeddings of paper titles and abstracts for a two-stage similarity search is effective and retrieves relevant studies. Because the proposed solution performs similarity search locally, it can potentially work on any resource containing scientific publications if a scraper and storage for that resource is provided.

While the main goal of a study has been achieved, future work on the topic can be outlined. Building a bigger dataset will allow for more accurate filtering during both stages of retrieval. Web scraper for this work was written manually for Sinteza and InfoM. For automation, a single, more flexible scraper should be created, or scrappers can be generated by LLM for each new source, like in the recent study [13].

```
input = 'This study examines Retrieval-Augmented Generation (RAG) in large language models (LLMs)
and their significant application for undertaking systematic literature reviews (SLRs). RAG-based
LLMs can potentially automate tasks like data extraction, summarization, and trend identification.
However, while LLMs are exceptionally proficient in generating human-like text and interpreting
complex linguistic nuances, their dependence on static, pre-trained knowledge can result in in-
accuracies and hallucinations. RAG mitigates these limitations by integrating LLMs' generative
capabilities...'

output=
                  link                title  title_dist             abstract  abstract_dist
76   https://portal.s...  Enhancing Retrie...    0.837151  This paper prese...       1.000000
22   https://portal.s...  Multimodal Retri...    0.846353  Maintaining stru...       0.811962
46   https://portal.s...  Leveraging LLMS ...    0.666090  Web forums conta...       0.687956
644  https://portal.s...  Development of O...    0.634476  This paper prese...       0.612257
113  https://portal.s...  Artificial Intel...    0.642561  The topic of thi...       0.594948
39   https://portal.s...  The Design Chara...    0.624681  This paper compr...       0.578272
859  https://portal.s...  APPROXIMATE SEAR...    0.609703  This paper is a ...       0.562659
258  https://portal.s...  Conversational S...    0.621597  The aim of the p...       0.552910
222  https://portal.s...  Autonomous Grade...    0.621232  Assessment is an...       0.545006
169  https://portal.s...  Generative Artif...    0.621148  Every stratum of...       0.454875
```

**Listing 1.** Output of completed similarity search

## REFERENCES

[1] D. Chandrasekaran, V. Mago. "Evolution of Semantic Similarity—A Survey," in ACM Computing Surveys, vol. 54, no. 2, pp. 1–37, 2021. doi: 10.48550/arXiv.2004.13820

[2] W. Li. "Scientific paper recommender system using deep learning and link prediction in citation network," in Heliyon, vol. 10, no. 14, 2024. doi: 10.1016/j.heliyon.2024.e34685

[3] M. Stäbler, S. Turnbull, T. Müller, C. Langdon, J. Marx-Goméz, and F. Köster, "The Impact of Chunking Strategies on Domain-Specific Information Retrieval in RAG Systems," in 2025 IEEE International Conference on Omni-layer Intelligent Systems (COINS), pp. 1-6, 2025. doi: 10.1109/COINS65080.2025.11125724

[4] B. Han, T. Susnjak, and A. Mathrani, "Automating Systematic Literature Reviews with Retrieval-Augmented Generation: A Comprehensive Overview," Applied Sciences, vol. 14, no. 19, 2024. doi: 10.3390/app14199103

[5] D. Vujić, A. Njeguš, N. Bačanin Džakula, "Enhancing Retrieval - Augmented Generation with Graph-Based Retrieval and Generative Modeling," in Sinteza 2025 - International Scientific Conference on Information Technology, Computer Science, and Data Science, Belgrade, Singidunum University, Serbia, 2025, pp. 3-9. doi: 10.15308/Sinteza-2025-3-9

[6] T. Nguyen, C. Pruski, M. Da Silveira. "Semantic Similarity Analysis of Scientific Papers in Scholarly Knowledge Graphs", 2025. [Online]. Available: https://ceur-ws.org/Vol-3977/NSLP-04.pdf

[7] N. Tran, et al, "Enriching PubMed related article search with sentence level co-citations," in AMIA Annual Symposium Proceedings, 2009, pp. 650. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC2815371/

[8] N. Reimers, I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, pp. 3982–3992. https://doi.org/10.48550/arXiv.1908.10084

[9] X. Zhang et al, "mgte: Generalized long-context text representation and reranking models for multilingual text retrieval," in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024, pp. 1393–1412. https://doi.org/10.48550/arXiv.2407.19669

[10] Google Developers, "Measuring similarity from embeddings," Machine Learning Clustering, Google, Accessed: Apr. 24, 2026 [Online]. Available: https://developers.google.com/machine-learning/clustering/dnn-clustering/supervised-similarity

[11] H. Steck, C. Ekanadham, N. Kallus, "Is cosine-similarity of embeddings really about similarity?," in Companion Proceedings of the ACM Web Conference 2024, 2024, pp. 887–890. https://doi.org/10.1145/3589335.3651526

[12] L. Richardson. "Beautiful soup documentation," in April, 2007. Accessed: Apr. 24, 2026. [Online]. Available: https://beautiful-soup-4.readthedocs.io/en/latest/

[13] M. Pavkovic et al, "Leveraging LLMS for Automatic Forum Scraper Generation," in Sinteza 2025-International Scientific Conference on Information Technology, Computer Science, and Data Science, 2025, pp. 78–85. https://doi.org/10.15308/Sinteza-2025-78-85