AI-Powered Semantic Search for Historical Documentation: A Collaborative Research with Hitachi Energy

Ivan Kelly Maranan Hansson*, Edvin Wiklund*, Alessio Bucaioni†, and Luciana Provenzano†
* Mälardalen University (Sweden)
† *name.surname@mdu.se*

***Abstract*—Companies with long operational histories often face the challenge of managing vast repositories of documentation, which hold critical knowledge needed for maintaining ongoing projects. Retrieving relevant information from these extensive archives is a time-consuming and complex task, requiring specialized expertise and familiarity with outdated terminology. Semantic search has emerged as a promising technology to address these issues by improving the precision and efficiency of information retrieval. In this paper, we present our collaborative research with Hitachi Energy, exploring the development of a semantic search engine based on existing open-source solutions to assist practitioners in searching large industrial historical document repositories. We first analyzed available No-SQL databases with search-engine interfaces, followed by an evaluation of pre-trained semantic transformers to determine which offers the best balance of accuracy and speed for semantic search. Our research identified OpenSearch as the most suitable No-SQL database due to its flexibility, free usage, and support for semantic transformers. After evaluating various pre-trained semantic transformers, we found all-MiniLM-L6-v2 to offer the best balance of accuracy and speed for semantic search. Based on the findings, we developed a prototype AI-powered semantic search tool, which was tested in a workshop involving Hitachi Energy professionals. Our findings demonstrate the feasibility and effectiveness of AI-powered semantic search for handling historical documentation, offering significant potential for industries tasked with managing large legacy archives.**

***Index Terms*—Semantic search, pre-trained semantic transformers, documentation, AI.**

I. Introduction

Corporations with long histories, such as Hitachi Energy, manage vast repositories of documentation spanning decades, often critical for maintenance, operations, and customer support. For instance, Hitachi's High Voltage Direct Current (HVDC) division still relies on projects initiated in the 1950s. However, these documents, often stored in paper-based or poorly digitized formats, are difficult to search efficiently due to inconsistent structures, heterogeneous formats, and irrelevant content like headers and footers. Searching these archives is time-consuming and error-prone. Traditional keyword-based search engines struggle with variations in terminology, misspellings, paraphrased concepts, and outdated terminology, making it harder to find relevant information. Recent advancements in semantic search, powered by AI [1], [2] and technologies like BERT and SBERT, address these challenges by understanding query intent and retrieving results based on semantic meaning, even in unstructured datasets [3].

This paper presents a collaborative effort with Hitachi Energy to develop an AI-powered semantic search solution. We evaluated No-SQL databases and pre-trained semantic transformers, selecting OpenSearch for its flexibility and free functionality and all-MiniLM-L6-v2 for its superior balance of accuracy and speed. Using the MS MARCO dataset to ensure data confidentiality, we measured performance with metrics such as Mean Reciprocal Rank ($MRR@k$) and Normalized Discounted Cumulative Gain ($nDCG@k$). The resulting prototype, tested with Hitachi Energy professionals, demonstrated the feasibility and effectiveness of AI-powered semantic search for managing large industrial archives, offering significant potential for similar use cases.

The remainder of this paper is structured as follows. Section II details the research process adopted for this work. Section III describes the design and execution of the experiments to evaluate pre-trained semantic transformers in conjunction with a No-SQL database. Section IV introduces the prototype of the AI-powered semantic search tool developed for Hitachi Energy. Section V presents insights from the validation workshop at Hitachi Energy, discussing the strengths and weaknesses of the solution, along with threats to validity. Section VI reviews related works from the literature, and Section VII concludes the paper with final remarks and possible future research directions.

II. Research Process

In this work, we employed a research process that is an adaptation of two methodologies: the engineering method by Basili [4] and the one by Gorschek et al. [5]. Our process focuses on implementing and connecting existing solutions [4] while also maximizing the transfer of technology between academia and industry through a multi-step validation process [5]. The first step is the elicitation of industrial needs, which was initiated by the Hitachi Energy representatives involved in this research. Starting from the identified need, we derived a generalized Research Goal (RG) that is:

RG: to investigate the feasibility of implementing an AI-powered solution to search historical documentation.

To comply with the engineering method, we began by building a candidate solution. This process started with collecting and reviewing existing solutions through a comprehensive literature review. To strike a balance between rigor and agility[1], we did not opt for a full-fledged systematic review. We searched three scientific databases: ACM Digital

[1]For the sake of space, we omit all the details of the literature review.

Library[2], IEEE Explore[3], and Google Scholar[4], with the aim of identifying existing No-SQL databases with search-engine interfaces, pre-trained semantic transformers, and their benefits. In doing so, we excluded studies that were not written in English, were not peer-reviewed, or were shorter than 4 pages. We conducted experiments on the No-SQL databases with search-engine interfaces and pre-trained semantic transformers identified during the literature review. The goal was to select the most performant sentence transformer for implementing a prototype AI-powered semantic search tool for Hitachi Energy. In particular, we evaluated all the collected pre-trained semantic transformers using Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG), and search speed (milliseconds) on the MS MARCO Top 1000 Evaluation set. Section III provides further details on these activities. Building a candidate solution was an iterative step coupled with a preliminary validation step called validation in academia, where each iteration of the candidate solution was evaluated in academic settings to assess its initial feasibility and applicability. To further validate the candidate solution and measure its relevance and applicability in industrial settings, we conducted a workshop with Hitachi Energy employees, including three global managers, one line manager, one pre-eminent engineer, and a team of seven engineers. During this workshop, we allowed participants to test the solution with a small sample dataset from Hitachi Energy. Section IV provides further details on the workshop.

III. Evaluating semantic transformers with No-SQL databases: the experiment

This section describes the experiment to evaluate pre-trained semantic transformers with a No-SQL database using three key metrics: Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG), and search speed in milliseconds. The evaluation was conducted on the first 100,000 pairs of the MS MARCO Top 1000 Evaluation dataset, as the full 8.8 million query-answer pairs were impractical to process due to computational constraints. We focused on $k = 10$ and $k = 100$ for MRR and nDCG, assessing how often relevant answers appeared in the top results. High scores at $k = 10$ indicate that relevant answers are frequently found within the top 10 results, while high scores at $k = 100$ suggest relevance within a broader set of results.

A. No-SQL database selection

Handling unstructured data is a major challenge in industrial environments. While databases like SQL and No-SQL manage data efficiently, searching for relevant information remains difficult. Traditional keyword-based methods often fail to handle the nuances of unstructured data. The state-of-the-art approach uses No-SQL databases with search-engine interfaces and semantic transformers [6]. From our literature review, we identified three No-SQL databases with built-in search interfaces as potential candidates for our solution.

- ElasticSearch: Built on the Lucene library, ElasticSearch is a full-text search engine with a distributed search and analysis engine at the core of the Elastic Stack. It operates under the Elastic License.
- OpenSearch: Originally a fork from version 7.10.2 of ElasticSearch, OpenSearch also builds on the Lucene library and supports full-text search. It is distributed under the Apache License Version 2.0 (ALv2).
- Apache Solr: Another search engine built on the Lucene library, Apache Solr supports full-text indexing and searching. Its operations follow three main steps: indexing, querying, and ranking. Solr is licensed under ALv2.

All three search engines are built on Apache Lucene, an open-source library for indexing and managing documents in No-SQL databases, leading to similar baseline performance. Table I compares their key benefits based on six categories identified in the literature, referencing studies such as [3], [7]–[14].

- Document storage: How data is stored in the database.
- Free of charge: Whether the database is entirely free to use without additional charges for full functionality.
- Dense vector search: Whether the database supports semantic search using dense vector techniques.
- Built-in sentence transformers: Whether the database has native support for sentence transformers.
- Support for third-party embedding models: Whether the system can integrate with external embedding models.
- Pre-embedding requirement: Whether sentences must be pre-embedded before insertion into the database.

Capability	**OpenSearch**	**ElasticSearch**	**Apache Solr**
No-SQL storage type	Document	Document	Document
Free of charge	X		X
Dense vector search (Semantic search)	X	X	X
Built-in sentence transformers	X	X	
Supports third-party embedding models	X	X	
Pre-embedding of sentences before insertion			X

TABLE I: Comparison of No-SQL databases based on their capabilities.

From Table I and the literature review, OpenSearch emerged as the best choice for our implementation. It meets all criteria, including being open-source, supporting dense vector searches, sentence transformers, and third-party models, and offering a free license. Its flexibility and modifiability make it ideal for both academic research and industrial use, such as at Hitachi Energy.

B. Dataset

The experiment used the first 100,000 lines of the MS MARCO Top-1000 Evaluation dataset [15], chosen for its similarity to Hitachi Energy's data and query patterns. A Python script automated the splitting and uploading of documents, creating passages of varying lengths, mirroring those in MS MARCO. The dataset's structure, with short queries matching both long and short passages, aligned with Hitachi Energy's typical queries. MS MARCO was also selected for its

[2]https://dl.acm.org/

[3]https://ieeexplore.ieee.org/Xplore/home.jsp

[4]https://scholar.google.com/

public availability, ensuring repeatability and generalizability for other researchers. It consists of question-answer pairs, each with a unique ID.

C. Evaluation Metrics

We used three evaluation metrics, where the first two metrics were used to evaluate the accuracy of the sentence transformers, while the final metric assesses the speed of the search.

1) Accuracy Metrics: We define accuracy based on how well the results of a query match the user's intent, which can be measured by the ranking of the retrieved results. To assess accuracy, we used two metrics that consider the ranking: MRR@k and nDCG@k.

MRR@k measures how quickly a user can find a relevant answer. Equation 1 shows how MRR is calculated by summing the reciprocal rank of the first relevant result for each query, where the rank of the correct answer is used.

$$\text{MRR@}k = \frac{1}{j}\sum_{j=1}^{j}\frac{1}{rank_j} \tag{1}$$

The result is an average across all queries, and the MRR value ranges between 0 and 1, with a higher value indicating better accuracy (i.e., relevant results appearing higher in the list of retrieved answers). The k value limits how many results are included in the retrieved list, and if no relevant result is found within the top k results, the query receives a reciprocal rank of zero. The MRR@k metric is well-suited for our goal, as it reflects how efficiently the system retrieves relevant answers, providing insight into whether the implementation is viable in real-world applications.

To further support our evaluation, we utilized the nDCG metric. Unlike MRR, nDCG takes into account not just the first correct result, but also the ordering of subsequent relevant results. nDCG compares the actual ranking of relevant documents to an ideal ranking. Equation 2 shows the formula for nDCG, which divides the Discounted Cumulative Gain (DCG) by the Ideal Discounted Cumulative Gain (IDCG).

$$\text{nDCG@}k = \frac{DCG@k}{IDCG@k} \tag{2}$$

The DCG at rank k is calculated using Equation 3, where rel_i is the relevance of the passage at rank i. A relevant result is assigned a value of 1, while an irrelevant result is assigned 0.

$$\text{DCG@}k = \sum_{i=1}^{k}\frac{rel_i}{\log_2(i+1)} \tag{3}$$

IDCG is computed by ordering the relevant passages in descending order and calculating the DCG for the ideal ranking (Equation 4).

$$\text{IDCG@}k = \sum_{i=1}^{k}\frac{rel_i}{\log_2(i+1)} \tag{4}$$

Both MRR and nDCG yield results between 0 and 1, with higher scores indicating that correct answers are ranked higher in the list of results. Together, these metrics provide a comprehensive view of the accuracy and ranking performance of the semantic transformers.

2) Speed Metric: Speed is defined as how quickly the system returns an answer to the user. To evaluate speed, we measured the execution time for each query in milliseconds, ensuring precise timing for each search operation. The total execution times for all queries were summed and averaged, as shown in Equation 5, where k represents the number of queries and t_i is the time in milliseconds for each query.

$$\text{Average Speed} = \frac{\sum_{i=1}^{k} t_i}{k} \tag{5}$$

This metric allowed us to assess whether the search system could return results quickly enough to be practical in real-world scenarios.

D. Experimental process

The experiment process, shown in Figure 1, was conducted on a laptop with an AMD Ryzen 7 4800U CPU and 16 GB of RAM using a Docker environment. A cluster of four OpenSearch nodes (version 2.13.0) was set up: two for data storage, one for machine learning tasks, and one for the dashboard interface. After initializing the cluster, a sentence transformer was uploaded (step 1, Figure 1), and an index containing answers, IDs, and sentence embeddings was created. The MS MARCO dataset answers were embedded and inserted into the database via a pipeline (step 2, Figure 1). Semantic searches were conducted using dataset queries (step 3, Figure 1). Queries were embedded and compared against the indexed data using k-NN search and cosine similarity to rank semantically closest matches (step 4, Figure 1). Execution times were averaged across all queries, and results were evaluated using nDCG and MRR metrics. Steps two through four were automated with Python to ensure accuracy and consistency, and the experiment was repeated for all selected sentence transformers.

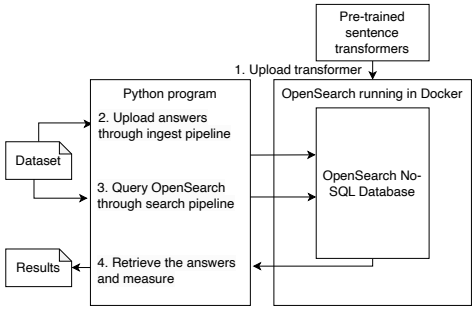

Fig. 1: Experimental process.

E. Results

The metrics used during the experiments are *MRR* and *nDGG* for accuracy of the search, and *average speed* for the time to provide the response to the search, as explained in Section III-C. A low score in MRR and nDCG indicates that the correct answers was ranked lower than expected, while a low average time per query suggests that the transformer executes queries quickly. From Table II, four sentence transformers stand out, demonstrating better performance across the different metrics. Upon analysis, it became evident that the

Sentence transformer	$MRR@10$	$MRR@100$	$nDCG@10$	$nDCG@100$	**Avg.** $ms/query$
alldistilroberta-v1	0.3377	0.3424	0.2546	0.4039	203
all-MiniLM-L6-v2	0.3617	0.3654	0.2876	0.4484	73
all-MiniLM-L12-v2	0.3572	0.3611	0.2824	0.4422	53
all-mpnet-base-v2	0.3470	0.3514	0.2640	0.4220	162
msmarco-distilbert-base-tas-b	0.3540	0.3582	0.2650	0.3867	94
multi-qa-MiniLM-L6-cos-v1	0.3566	0.3605	0.2810	0.4360	67
multi-qa-mpnet-base-dot-v1	0.3223	0.3278	0.2341	0.3722	169
paraphrase-MiniLM-L3-v2	0.3580	0.3618	0.2817	0.4192	74
paraphrase-multilingual-MiniLM-L12-v2	0.3060	0.3115	0.2202	0.3458	55
paraphrase-mpnet-base-v2	0.3341	0.3389	0.2459	0.3829	177
distiluse-base-multilingual-cased-v1	0.2531	0.2617	0.1648	0.2930	94
Average	0.3352	0.3401	0.2528	0.3957	111

TABLE II: Result from the experiment

fluctuations in results between most transformers were minimal. However, two models, distiluse-base-multilingual-cased-v1 and paraphrase-multilingual-MiniLM-L12-v2, performed significantly worse in both MRR and nDCG, indicating that these transformers ranked the correct answers much lower in the list of retrieved answers.

No clear correlation was observed between embedding dimensions and accuracy, nor did optimization for semantic search guarantee superior performance. Notably, paraphrase-mpnet-base-v2, with 512 dimensions, performed poorly across all metrics, particularly in terms of speed. The worst transformer in terms of speed was alldistilroberta-v1, which took 203ms per query, 92ms longer than the average. This shows that while some models may excel in certain areas, they may fall behind significantly in others.

IV. Hitachi AI-powered Semantic Search Tool Prototype

Based on previous section findings, OpenSearch was selected as the most suitable No-SQL database for its flexibility, open-source nature, and support for semantic transformers. The "all-MiniLM-L6-v2" transformer was chosen for its balance of accuracy and speed, forming the foundation of an AI-powered semantic search prototype for Hitachi Energy. The tool's architecture integrates OpenSearch with a PyQt6-based GUI, enabling intuitive user interaction. The opensearch-py library connects the UI to OpenSearch, which runs in a Docker containerized setup with three nodes: two data nodes for storing documents and embeddings and one node hosting the sentence transformer for efficient semantic processing. The GUI allows users to set query parameters, including the search text, the number of answers to retrieve, and the k-value for top-k results. Results are displayed with details such as scores, documents, and retrieved answers, while additional metrics on document retrieval are shown for user insight. This setup provides a seamless and efficient semantic search experience.

A. Evaluating the Hitachi AI-powered Semantic Search Tool Prototype

We evaluated the prototype in a workshop with 12 Hitachi Energy employees, including managers and engineers who regularly work with historical documents and face challenges in retrieving information. The workshop had three parts: a 15-minute introduction presenting the tool's purpose, functionality, and architecture; a 35-minute hands-on session where participants tested the prototype using domain-specific queries and discussed its limitations and improvements; and a 30-minute discussion on the tool's broader applicability and potential integration within the company. Insights and feedback from the workshop are detailed in Section V.

V. Discussion

In this section, we reflect on the results of our study and the development of the AI-powered semantic search prototype. We evaluate the insights gained from both the technical experiment and the practical validation workshop conducted with Hitachi Energy. Additionally, we discuss the strengths and limitations of our approach, while also addressing potential threats to the validity of our findings.

A. On the results and prototype

As discussed in Section III, we identified three search engines that met the needs of our solution. Ultimately, OpenSearch was chosen due to its flexibility and the fact that it offers all functionalities free of charge. While we focused on feasibility rather than optimization (refer to Section V-B for more details on this topic), we acknowledge that the other solutions, such as ElasticSearch, could potentially perform better in terms of speed, memory usage, and redundancy, as some papers suggest ElasticSearch has faster execution. However, since all the identified search engines are built on the Apache Lucene library, we believe they would perform similarly and be relatively easy to switch between due to their shared underlying architecture. This assumption is further supported by CERN's successful transition from ElasticSearch to OpenSearch [7]. Our experiments revealed that the all-MiniLM-L6-v2 sentence transformer was the best fit for our solution, outperforming other transformers across most of our chosen metrics. While it was marginally slower in query time compared to the second-best model, the difference of 20ms was considered negligible. We prioritized accuracy metrics, which showed that all-MiniLM-L6-v2 consistently returned correct answers higher in the result list, thereby improving the overall search speed by minimizing user effort. Interestingly, transformers optimized specifically for semantic search did

not outperform the others in terms of accuracy; only multi-qa-MiniLM-L6-cos-v1 made it into the top four. However, the performance gap between the top four transformers was minimal, suggesting that any of them could work well in our solution. It's important to note that the experiment was limited to 100,000 query-answer pairs due to time constraints. While this limitation may have affected the results, we argue that our sample size was sufficient to draw meaningful conclusions. Another observation is that sentence transformers producing larger embedding dimensions performed worse in terms of speed, which aligns with the expectation that k-NN search in OpenSearch performs less efficiently with larger embeddings. Surprisingly, larger embeddings did not correlate with higher accuracy, which seems counter-intuitive, as we expected larger dimensions to capture more nuanced differences in passages and sentences. Our findings align with the work of Xian et al. [3], demonstrating that existing tools can be combined easily to create a powerful vector search solution. We expanded on their work by testing some of the AI tools provided by OpenSearch, and our validation activities at Hitachi Energy confirm that it is feasible to achieve AI-powered semantic search for historical documentation. Lastly, our study did not focus on optimizing the performance of the tested sentence transformers. A more in-depth fine-tuning process, specifically tailored to a dataset like Hitachi Energy's, could yield better results. During the validation process, we observed indications that such customization would be highly beneficial, as a model trained on domain-specific data would likely outperform the pre-trained models we used.

B. Insights from the validation workshop

The practitioners at Hitachi Energy were generally satisfied with the potential offered by our prototype for searching large historical documentation. One key observation was that search speed was prioritized over accuracy. In the current process at Hitachi Energy, searching for information in historical archives involves locating documents by title and manually sifting through an average of 40 to 50 pages containing varied information, such as installed equipment details, user guides, and release notes. In this context, the primary requirement for a semantic search engine is that it completes searches swiftly, ideally within seconds, and without exceeding a 30-second threshold. A faster search engine mitigates the need for perfect accuracy, as engineers can refine their queries and quickly repeat searches until they obtain more satisfying results. This balance between speed and accuracy aligns with the primary operational needs at Hitachi Energy. Another strength of our solution was the use of a GUI. The 12 practitioners who tested the tool found the GUI intuitive and easy to interact with. Despite the difference in the way the tool operates—where the search process does not require document titles—the new search method was quickly understood and adopted. This ease of use is an encouraging result that may positively influence the future adoption of the tool within the company. Finally, we believe that the decision to use off-the-shelf and open-source solutions to develop an AI-powered semantic search tool offers a cost-effective and feasible approach for companies looking to introduce semantic search into their workflows. By leveraging readily available, continuously updated technologies and existing know-how, companies can significantly reduce the cost and time associated with developing and maintaining such tools. This approach provides flexibility and scalability without requiring the heavy investment associated with proprietary solutions.

C. Threats to validity

We identified several possible threats to the validity that may have affected this study [16]. The first threat concerns the metrics we used to evaluate the accuracy of the search engine, namely MRR and nDCG. Since there are no universally recommended metrics for evaluating accuracy in the literature, the choice of these metrics may affect the construct validity of the experiment. However, we selected MRR and nDCG because they are widely used in similar studies, as explained in Section III. Furthermore, MRR is the standard metric used for the MS MARCO dataset, which we used in our experiment. The choice of the sentence transformers and dataset may represent a further threat to the validity of our work because they can introduce unknown factors that affect the causal relationship we are investigating. We limited the number of evaluated sentence transformers to those already provided in OpenSearch, the No-SQL database chosen for our experiment (Section III). However, the selected transformers are among the most popular and well-regarded multilingual and English transformers on the HuggingFace community website[5], making them appropriate for our study. A further threat is related to the computational environment used for our experiment. We ran the prototype on a laptop with limited computational resources, which may have impacted the performance of the evaluated transformers. Although this setup reflects realistic constraints in terms of hardware for some users, the system might perform differently in a high-performance computing environment, potentially yielding faster results or better scalability. Another threat concerns the confidentiality of our test data, which may limit the generalizability of our findings. To maintain the confidentiality of Hitachi Energy's documents, we used the publicly available MS MARCO dataset to simulate their documentation, as previously explained. While this approach enhances the relevance of our work for other research studies, the solution might perform differently when applied to proprietary or domain-specific datasets. Finally, the validation process used in our study may introduce another threat to external validity. Although we conducted a workshop with Hitachi Energy employees to evaluate the usability and effectiveness of the tool, the feedback was limited to a small number of participants within a specific organizational context. Broader evaluations involving different industries, datasets, and end-users may reveal additional insights or limitations.

VI. Related work

In this section, we present state-of-the-art research that aligns with our work.

A. About sentence transformers

The use of sentence transformers has expanded significantly, with key contributions from Devlin et al., who introduced

[5]https://huggingface.co/models?pipeline_tag=sentence-similarity&sort=likes

BERT [17], and Reimers and Gurevych, who developed SBERT [18]. SBERT addressed BERT's limitations for Semantic Textual Similarity (STS) tasks by using a Siamese network, improving processing speed while retaining accuracy. These foundational works underpin our research, which evaluates pre-trained transformers based on BERT and SBERT, focusing on performance in speed and accuracy on external test data. Zhang et al. introduced the Hybrid List Aware Transformer Reranking (HLATR) module, outperforming BERT and RoBERTa in efficiency and robustness for document retrieval [19]. Using the MS MARCO dataset and Mean Reciprocal Ranking (MRR), their study aligns with ours but differs in scope. While they used the full MS MARCO dataset, we focused on the top 1000 passages, resembling Hitachi Energy's data. Additionally, our study evaluates off-the-shelf transformers in OpenSearch, offering insights into practical, ready-to-use tools.

B. About AI-powered search in large-scale database

The challenge of retrieving relevant information from large-scale databases has gained attention with AI advancements, though the field remains relatively new and underexplored. Few papers provide comprehensive evaluations, and ongoing developments make it difficult to identify the most feasible solutions. Andre et al. explored semantic search for COVID-19 publications using the TREC-COVID and CORD-19 datasets [20]. Their evaluation employed metrics like nDCG, top-N precision, and MAP, but their CO-search engine relied on complete relevance judgments, highlighting a limitation. While their work developed a new system, our study focuses on evaluating existing sentence transformers within a search engine context rather than creating a system from scratch. Xian et al. examined vector search technologies, comparing Lucene (used in OpenSearch and ElasticSearch) with Facebook's Faiss [3]. They identified differences in speed, query latency, and throughput, emphasizing the importance of dense vector manipulation for modern search. Their findings provided a foundation for our work, which leverages Lucene for dense vector searches. We extend their approach by evaluating pre-trained sentence transformers integrated with Lucene in OpenSearch. Our study builds on these works by focusing on practical applications, evaluating pre-trained models, and integrating them into real-world systems like OpenSearch to improve information retrieval in unstructured datasets.

VII. Conclusion and future work

This study examined the feasibility of using AI-powered semantic search to retrieve historical documentation in an industrial setting, focusing on Hitachi Energy. We identified OpenSearch as the most suitable No-SQL database and selected the all-MiniLM-L6-v2 transformer for its balance of accuracy and speed. A prototype integrating these components into a graphical user interface was developed and evaluated by 12 Hitachi Energy employees using real-world data. The tool delivered accurate results in under one second, meeting the company's primary requirements. These findings demonstrate the potential of AI-powered semantic search for managing extensive legacy archives and contribute valuable insights to this emerging field.

Future work could include training a custom sentence transformer on organization-specific data, enhancing scalability for larger user loads, refining search parameters, and improving the GUI. Testing with larger datasets and across industries would further validate and expand the tool's applicability.

References

[1] B. B. Musabimana and A. Bucaioni, "Integrating aiaas into existing systems: The gokind experience," in *International Conference on Information Technology-New Generations*. Springer, 2024, pp. 417–426.

[2] R. Nazir, A. Bucaioni, and P. Pelliccione, "Architecting ml-enabled systems: Challenges, best practices, and design decisions," *Journal of Systems and Software*, vol. 207, p. 111860, 2024.

[3] J. Xian, T. Teofili, R. Pradeep, and J. Lin, "Vector search with openai embeddings: Lucene is all you need," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, ser. WSDM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1090–1093.

[4] V. R. Basili, "The experimental paradigm in software engineering," in *Experimental Software Engineering Issues*, 1992. [Online]. Available: https://api.semanticscholar.org/CorpusID:14982809

[5] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE software*, vol. 23, no. 6, pp. 88–95, 2006.

[6] C. Ni, J. Wu, H. Wang, W. Lu, and C. Zhang, "Enhancing cloud-based large language model processing with elasticsearch and transformer models," 2024.

[7] S. Papadopoulos, P. Saiz, and S. Ulrich, "Architecting the opensearch service at cern," in *26th International conference on computing in high energy and nuclear physics (CHEP2023)*, 2023.

[8] Z. Parker, S. Poe, and S. V. Vrbsky, "Comparing nosql mongodb to an sql db," in *Proceedings of the 51st ACM Southeast Conference*, ser. ACMSE '13. New York, NY, USA: Association for Computing Machinery, 2013.

[9] J. Hansen, K. Porter, A. Shalaginov, and K. Franke, "Comparing open source search engine functionality, efficiency and effectiveness with respect to digital forensic search," 2018. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2584227/577-Article%2BText-1167-1-10-20181009.pdf?sequence=2&isAllowed=y

[10] X. Chang, "The analysis of open source search engines," *Highlights in Science, Engineering and Technology*, vol. 32, p. 32–42, 2 2023.

[11] N. Kathare, O. V. Reddy, and V. Prabhu, "A comprehensive study of elasticsearch," *International Journal of Science and Research (IJSR)*, 2020.

[12] S. Gupta and R. Rani, "A comparative study of elasticsearch and couchdb document oriented databases," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, 2016, pp. 1–4.

[13] G. Liu, C. Li, W. Tian, and Z. Li, "Distributed geospatial data service based on opensearch," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 100–104.

[14] R. LeVan, "Opensearch and sru: A continuum of searching," *Information Technology and Libraries*, vol. 25, no. 3, p. 151–153, 9 2006.

[15] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang, "Ms marco: A human generated machine reading comprehension dataset," 2018.

[16] P. Runesson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Emperical Software Engineering*, pp. 131–164, 2009.

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[18] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 8 2019, event Title: The 2018 Conference on Empirical Methods in Natural Language Processing. [Online]. Available: https://tubiblio.ulb.tu-darmstadt.de/117723/

[19] Y. Zhang, D. Long, G. Xu, and P. Xie, "Hlatr: Enhance multi-stage text retrieval with hybrid list aware transformer reranking," 05 2022.

[20] A. Esteva, A. Kale, R. Paulus *et al.*, "Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization," *npj Digit. Med.*, vol. 4, p. 68, 2021.