

Implementing a Retrieval Augmented Generation Architecture for the Power Engineering Sector: Some Fundamentals [†]

Vlad Aniței ¹ , Adrian-Valentin Boicea ^{2,*} and Mihai-Stelian Munteanu ¹

¹ Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; vlad.anitei@umfst.ro (V.A.); mihai.munteanu@ethm.utcluj.ro (M.-S.M.)

² Department of Electrical Power Systems, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania

* Correspondence: valentin.boicea@upb.ro

[†] Presented at the International Conference on Electromagnetic Fields, Signals and BioMedical Engineering (ICEMS-BIOMED), Suceava, Romania, 7–9 May 2026.

Abstract

This paper presents the implementation and evaluation of a Retrieval Augmented Generation (RAG) architecture applied to the power engineering domain. A scanned Romanian textbook on the reliability of energy installations serves as input data, requiring Optical Character Recognition (OCR) preprocessing. The extracted text has been segmented into chunks, embedded using a sentence-transformer model, and indexed in a FAISS vector database, with GPT-4o used for generation. The embedding space analysis reveals distinct thematic clusters, confirming semantic coherence despite OCR-induced noise. The evaluation results demonstrate that RAG architectures can provide valuable domain-specific insights even when operating on imperfect legacy engineering documentation.

Keywords: answer relevance; context precision; faithfulness; foundation model; information retrieval; LLM; RAG

1. Introduction

Even if the simultaneous use of a foundation model and a Large Language Model (LLM) is usually preferred in the energy sector, the Retrieval Augmented Generation (RAG) coupled with a pretrained LLM can still prove a useful tool as far as the operation of power grids is concerned. In this sense, it is important to understand that a foundation model is not the same as an LLM. A foundation model actually represents a pretrained artificial intelligence (AI) architecture that can use as data text, images, time series or can be even multimodal. On the other hand, an LLM is a text specialized foundational model. In most cases relevant to the power engineering sector, the LLM is used just to interact with the human operator at the verbal or written message level while the foundation model is used to reason. In many real-life scenarios, this is firstly trained on time series representing power generation, power consumption, and various meteorological or maintenance data. The end result can be represented by prediction, optimization, fault detection, various scenario generation for normal or abnormal operation of the power system, or even data augmentation. In this way, the foundation model will be able to make decisions in real time regarding how to render the operation of the power system more efficiently, containing different types of generation units, consumers and storage. By simultaneously using more than one data type, the result will become more accurate but, at the same time, more energy and various computational resources will be required. A good compromise in this direction



Academic Editors: L. Dan Milici,
Simona Miclăuș, Paul Bechet and
Radu Ciorap

Published: 16 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

would be the use of a RAG architecture and an already implemented GPT like Gemini or ChatGPT. Even if such a system cannot make decisions and influence the operation of a power network in real time, it still can provide valuable insights regarding its potential behavior under certain given scenarios. RAG, under these circumstances, will enhance the quality of the delivered responses by mitigating the number of hallucinations. In order to test the efficiency of this architecture, one has uploaded a scanned handbook as input data. As such, the system will firstly be forced to carry out the Optical Character Recognition (OCR) of the text and only after this, to correlate the resulting fragments of information with the query introduced by the user.

2. The Classical RAG Architecture

A classical RAG architecture consists of the following components that are used in this order: information retrieval (IR), augmentation, and, finally, generation. Thus, the goal of the first component is to capture the information from various input data. In this sense, one can distinguish between multiple classes of IR [1]. The first class is represented by whole documents or fragments of text. On the other hand, modern architectures tend to be multimodal. This means that even if the various types of information can be retrieved at the same time, there are few important conditions that need to be fulfilled. Hence, each information type will correspond to a certain data set. As such, they have to be encoded, stored differently, and, of course, retrieved using different strategies. For example, the input data can be represented by image embeddings and, in this case, the query is either a text or an image. The retrieval will return either relevant images or image–text groups. In case of audio data, the raw audio is transformed into a numerical vector representation that captures its content and/or various features of the sound, depending on what is being requested. This kind of representation facilitates the searching, comparison, and the easy retrieval of the information. When it comes to video streams, they are too complex to be embedded directly as a whole. Usually, the video and the audio streams are integrated separately. For instance, a certain number of frames is extracted every n seconds, and then every frame that represents an image will be encoded into an embedding. On the other hand, the speech can be converted into text through ASR (Automatic Speech Recognition). In this case, the retrieval can ultimately consist of video frames, transcript snippets and timestamps. Moreover, this architecture can also capture structured data through hybrid methods, like time series, tables, or knowledge graphs. Hybrid methods imply that the architecture uses the Dense Retrieval, the Sparse Retrieval, as well as the Metadata Filtering at the same time. Through Dense Retrieval, one can convert documents or fragments of documents along with user queries into vectors. Then, one compares them using either the Cosine Similarity or the dot product. Usually, the encoding takes place in this case through BERT (Bidirectional Encoder Representations from Transformers) architectures. The Sparse Retrieval is based on exact words and frequency, and not on the meaning. The most common algorithm used here is the BM25 (Best Match 25). The Metadata Filtering, however, is done through structured constraints. In order to gain a better understanding of these hybrid methods, refer to Table 1.

Table 1. Strengths and weaknesses of the IR hybrid methods.

Method	Strength	Weakness
Dense	meaning	misses exact keywords
Sparse (BM25)	exact terms	misses paraphrases
Metadata Filtering	precision/constraints	not semantic

Regarding the augmentation, the most common form is the prompt stuffing. In this case, the LLM reads the query and the stored text fragments and finally generates the answer. Other more advanced augmentation techniques are the Retrieval-aware Prompting, the Chunk compression, the Reranking Before Augmentation, the Multi-step (Iterative) Augmentation, and the Tool-augmented Context.

While in the generation phase, the LLM does not search for text fragments anymore. It will simply read the augmented prompt and predict the next tokens. So, the first step is the tokenization, in which the sentence is split into words (tokens). Within the second step, the Transformer processes the question and the context through the Self Attention Mechanism. As such, all input parts are related.

More simply put, without a RAG, an LLM may hallucinate because it only relies on training data. The RAG helps the model “see” more evidence in the context, and therefore the generation becomes more factual.

3. Experimental Setup

As mentioned in Section 1, in order to test the efficiency of the proposed RAG architecture, one has selected as input data a scanned textbook [2]. This data set has been digitized and processed locally only for research purposes. The content was used exclusively for evaluating the RAG performance and was not redistributed or used in the training of external LLMs. As such, all these operations have been carried out also in accordance with the EU Directive 2019/790, regarding the text and data mining.

This particular source has been chosen for two main reasons. First, it covers a broad spectrum of topics that are representative for the power engineering field, ranging from reliability theory and failure rate analysis to the maintenance strategies of electrical equipment and protection systems. Second, being a scanned document rather than a natively digital one, it forces the system to first carry out the OCR before any information retrieval can take place. In this way, the experiment tests not only the RAG pipeline itself but also its robustness when dealing with imperfect, noisy textual input. The capturing of information is based on Dense Retrieval, as described in Section 2.

The experimental pipeline has been structured in four sequential stages. Within the first stage, the scanned PDF document has been processed through an OCR engine in order to extract the raw text. Given the age and the quality of the original scan, the OCR output has been subject to a certain degree of character recognition errors, particularly affecting diacritical marks specific to the Romanian language as well as mathematical symbols and subscripts that frequently appear in the power engineering literature. A representative excerpt of the OCR preprocessing stage is presented in Figure 1. The original scanned page (a) and the corresponding raw output produced by the OCR engine (b) are placed side by side, with nine representative recognition errors highlighted and enumerated in the accompanying legend. The errors selected for annotation cover the four main failure categories observed across the corpus: substitutions in body text (e.g., *analiza* → *ansliza*), loss of subscripts and inline mathematical notation (e.g., W_e rendered as *VW*), corruption of displayed equations including primed variables (e.g., the relation $t_e = t_1 + t'_1$ hallucinated as the unrelated word *Statele*), and degradation of Romanian-specific diacritics (e.g., *t* misrecognized alternately as *t* and as *j* within the same word *nefuncționare* on different occurrences). Crucially, the same lexical item is not corrupted consistently across the document, which propagates downstream into the chunking and embedding stages and explains a portion of the variance observed in the Faithfulness metric reported in Section 4. No manual correction has been applied to the OCR output, so that the experiment can reflect a realistic deployment scenario [3].

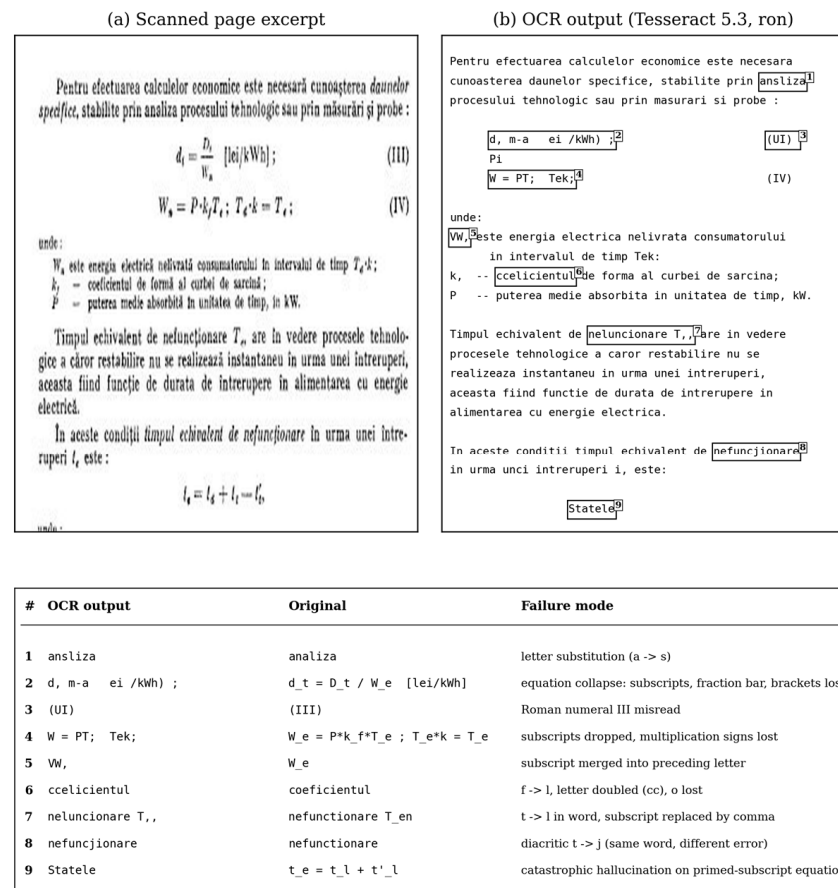


Figure 1. Comparative excerpt illustrating the OCR preprocessing stage. (a) Original scanned page from *Fiabilitatea Instalațiilor Energetice* ([2]). (b) Raw text produced by Tesseract 5.3 with the Romanian language model (ron), with no manual correction applied. The numbered rectangles mark representative recognition errors, enumerated in the legend below the panels.

Within the second stage, the extracted text has been segmented into chunks. The chunking strategy has employed a sliding window approach with a chunk size of 512 tokens and an overlap of 64 tokens. This overlap ensures that sentences that fall at the boundary between two consecutive chunks are not split in a way that would compromise their semantic integrity. After the chunking process, a total of 108 text fragments have been obtained from the entire document.

Within the third stage, each chunk has been transformed into a dense vector representation (embedding) using a sentence-transformer model, namely all-MiniLM-L6-v2. This model maps each text fragment into a 384-dimensional vector space [4]. The resulting embeddings have been stored in a FAISS (Facebook AI Similarity Search) vector database, which enables efficient similarity search through approximate nearest neighbor algorithms. The indexing structure chosen for it is the flat L2 index, which computes exact Euclidean distances and is suitable for data sets of this size [5,6].

Within the fourth and final stage, the Retrieval Augmented Generation pipeline has been assembled. For a given user query, the system first encodes the query using the same sentence-transformer model, then retrieves the top-k most similar chunks from the vector store (with $k = 5$), and finally passes both the query and the retrieved context to the generation model. The generation has been carried out using the GPT-4o model accessed through the OpenAI API. The system prompt has been designed to instruct the model to answer strictly based on the provided context, and then to indicate whenever the context is insufficient to formulate a reliable answer.

4. Results

The first aspect that has been analyzed is the quality of the vector space in which the 108 document chunks have been embedded. To visualize the high-dimensional embedding space, the t-SNE (t-distributed Stochastic Neighbor Embedding) dimensionality reduction technique has been applied, projecting the 384-dimensional vectors onto a two-dimensional plane [7]. The result of this projection is presented in Figure 2. It is important to note that the two axes of the t-SNE plot do not carry an intrinsic physical meaning. They represent abstract dimensions resulting from the nonlinear projection of the original 384-dimensional embedding space onto a two-dimensional plane, chosen by the algorithm such that the local neighborhood structure of the high-dimensional data is preserved as faithfully as possible. The t-SNE perplexity parameter has been set to 30 and the number of iterations to 1000, values that have been selected through preliminary experimentation in order to obtain a visually interpretable separation of the clusters.

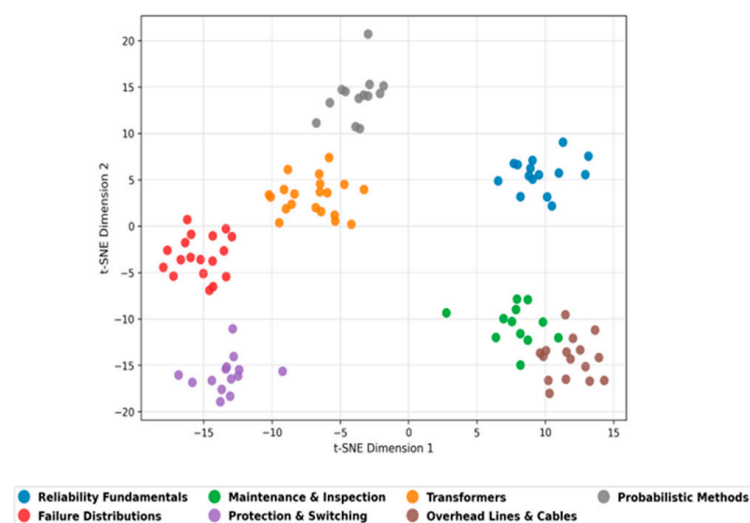


Figure 2. t-SNE projection of the 108 document chunk embeddings into a two-dimensional space. The two axes represent the abstract components obtained through the nonlinear dimensionality reduction and do not carry a direct physical interpretation. The 7 identified thematic clusters are distinguished through different colors, as indicated in the legend. Overlapping points within the same cluster correspond to chunks with nearly identical semantic content.

Consequently, the chunks are not uniformly scattered across the embedding space. Instead, they form seven distinct clusters that correspond to the main thematic areas covered by the source textbook. These clusters have been identified through K-Means clustering and have been labeled accordingly: reliability fundamentals, failure distributions, maintenance and inspection, protection and switching, transformers, overhead lines and cables, and probabilistic methods. The fact that semantically related chunks are grouped together in the vector space indicates that the embedding model is capable of capturing the topical structure of the document, even though the input text has been obtained through OCR and contains a certain level of noise.

This clustering behavior has a direct positive impact on the retrieval quality. When a user query is encoded and projected into the same vector space, the nearest neighbor search will preferentially select chunks that belong to the same thematic cluster, thus ensuring that the retrieved context is semantically coherent. On the other hand, there is also a certain degree of overlap between clusters, particularly between the reliability fundamentals cluster and the failure distributions cluster. This is expected, given that these two topics share a significant amount of terminology and mathematical formalism.

RAG Evaluation Metrics

The quantitative evaluation of the RAG pipeline has resulted in the data summarized in Table 2. The evaluation has been conducted using the RAGAS framework [8], which provides reference-free metrics specifically designed for assessing RAG pipelines. Across all 40 test queries, the mean Faithfulness score has been 0.784, the mean Answer Relevance has been 0.836, and the mean Context Precision has been 0.728. These values indicate that the system performs reasonably well for a domain-specific application built on top of OCR-processed text.

Table 2. Overall RAG evaluation metrics across all 40 test queries.

Metric	Mean (μ)	Std. Dev. (σ)	Min
Faithfulness	0.784	0.092	0.571
Answer Relevance	0.836	0.078	0.643
Context Precision	0.728	0.114	0.482

The detailed distribution of these metrics across the 10 query categories is presented in Figure 3. The Answer Relevance metric is consistently the highest across all categories, which suggests that the generation model is effective at producing responses that directly address the user's question. The Faithfulness metric, while slightly lower, still maintains acceptable values. The relatively lower Context Precision scores can be attributed to the fact that certain queries touch upon topics that span multiple thematic areas, causing the retrieval to include chunks that, while not entirely irrelevant, do not directly contain the information needed.

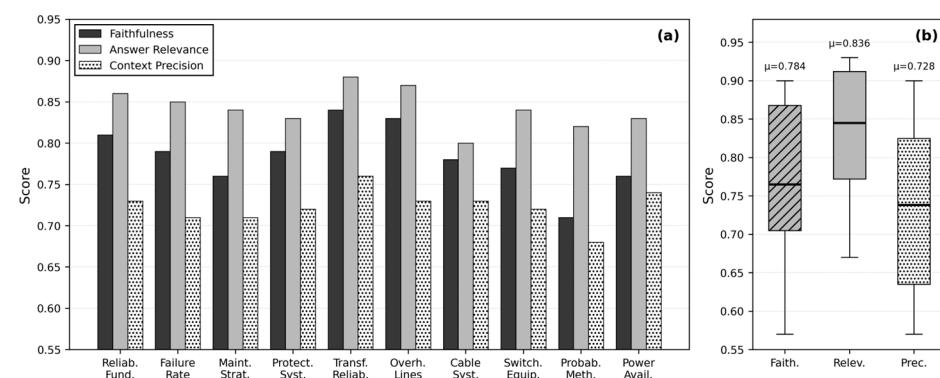


Figure 3. RAG evaluation metrics (Faithfulness, Answer Relevance, and Context Precision) across the 10 query categories. The grouped bar chart (a) shows the mean values per category, while the boxplot (b) presents the overall distribution of each metric.

The query categories that have achieved the highest combined scores are transformer reliability and overhead line faults. This can be explained by the fact that these topics are discussed in dedicated, well-delineated chapters in the source textbook, which results in more focused and less ambiguous chunk embeddings. On the other hand, the probabilistic methods category has the lowest Context Precision score (0.482 for the worst-performing query). This is due to the fact that probabilistic concepts appear throughout the entire textbook in various contexts, making it more difficult for the retrieval to isolate the most relevant passages.

An important observation concerns the impact of OCR quality on the evaluation results. Several queries that involved mathematical expressions or formulas have received lower Faithfulness scores, because the OCR engine has not been able to correctly recognize certain subscripts, Greek letters, or special symbols. In such cases, even though the retrieval

has correctly identified the relevant chunk, the corrupted text within the chunk has led to imprecise or incomplete answers. This limitation is inherent to the use of scanned input data and represents one of the main challenges that appear when deploying RAG architectures on legacy engineering documentation [3].

5. Conclusions

This paper has presented the implementation and evaluation of a RAG architecture applied to the power engineering domain. The system has been tested using a scanned Romanian textbook on the reliability of energy installations as input data, which has required the integration of an OCR preprocessing step into the pipeline. The experimental results have shown that the proposed approach achieves a mean Faithfulness of 0.784, a mean Answer Relevance of 0.836, and a mean Context Precision of 0.728 across 40 test queries distributed over 10 thematic categories.

The embedding space analysis has confirmed that the sentence-transformer model is capable of organizing the document chunks into semantically coherent clusters, even when the input text contains OCR-induced noise.

Future work will focus on three main directions. First, the integration of a reranking mechanism to improve the Context Precision of the retrieval step. Second, the exploration of multimodal foundation models that can process the scanned pages directly, without requiring explicit OCR. Third, the extension of the evaluation framework to include more diverse source materials, such as technical standards, equipment datasheets, and real-time SCADA data, in order to assess the scalability and generalizability of the proposed architecture for broader power engineering applications.

Author Contributions: Conceptualization, V.A., A.-V.B. and M.-S.M.; methodology, V.A., A.-V.B. and M.-S.M.; formal analysis, V.A., A.-V.B. and M.-S.M.; investigation, A.-V.B.; resources, M.-S.M.; data curation, V.A., A.-V.B. and M.-S.M.; writing—original draft preparation, A.-V.B.; writing—review and editing, V.A., A.-V.B. and M.-S.M.; supervision, M.-S.M.; project administration, M.-S.M.; funding acquisition, M.-S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the data used is available within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
BM25	Best Match 25
GPT	Generative Pre-trained Transformer
IR	Information Retrieval
LLM	Large Language Model
OCR	Optical Character Recognition
RAG	Retrieval Augmented Generation
RAGAS	Retrieval Augmented Generation Assessment
SCADA	Supervisory Control and Data Acquisition
t-SNE	t-distributed Stochastic Neighbor Embedding

References

1. Hindi, M.; Mohammed, L.; Maaz, O.; Alwarafy, A. Enhancing the Precision and Interpretability of Retrieval-Augmented Generation (RAG) in Legal Technology: A Survey. *IEEE Access* **2025**, *13*, 46171–46189. [[CrossRef](#)]
2. Nitu, V. *Fiabilitatea Instalațiilor Energetice*; Editura Tehnică: Bucharest, Romania, 1979.
3. Zhang, J.; Zhang, Q.; Wang, B.; Ouyang, L.; Wen, Z.; Li, Y.; Chow, K.-H.; He, C.; Zhang, W. OCR Hinders RAG: Evaluating the Cascading Impact of OCR on Retrieval-Augmented Generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Honolulu, HI, USA, 19–20 October 2025.
4. Reimers, N.; Gurevych, I. 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), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
5. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2019**, *7*, 535–547. [[CrossRef](#)]
6. Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; Jégou, H. The Faiss library. *arXiv* **2024**, arXiv:2401.08281.
7. van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
8. Es, S.; James, J.; Espinosa-Anke, L.; Schockaert, S. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL), St. Julians, Malta, 17–22 March 2024; pp. 150–158.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.