The Impact of Document Segmentation on the Efficiency of RAG Systems

Marko Ćurković, Korina Jurić and Ivan Lisica*

Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

{marko.curkovic, korina.juric, ivan.lisica}@fer.hr

*Corresponding author

Abstract

Purpose – This research investigates the impact of document segmentation, or "chunking," on the efficiency and accuracy of Retrieval-Augmented Generation (RAG) systems, specifically within the context of technical and regulatory documentation. The study aims to identify the optimal balance between fragment granularity and semantic context to minimize hallucinations and improve retrieval precision in knowledge-intensive tasks.

Methodology – The primary corpus consists of technical specifications and regulatory acts related to the Republic of Croatia's Fiscalization standards. Using the LlamaIndex framework, seven different segmentation strategies were evaluated, including rigid fixed-size chunking (ranging from 128 to 1024 tokens) and context-aware semantic segmentation. System performance was assessed using the RAGAs framework, employing *gpt-4o-mini* as an automated "LLM-as-a-Judge" to score responses based on faithfulness, context relevancy, and answer correctness against a ground-truth dataset of 30 specialized questions.

Findings – Experimental results demonstrate that document segmentation is critical to RAG success, with the 512-token chunk size emerging as the optimal configuration for technical datasets. The strategy utilizing 512 tokens with a 100-token overlap achieved the highest composite score (0.550), while smaller 128-token segments improved speed at the cost of detail. Notably, complex Semantic Chunking underperformed compared to fixed-size methods, suggesting that simple, consistent segmentation is more reliable for highly structured regulatory PDF files

Originality/Value – This study provides empirical evidence on how specific chunking parameters affect the "lost-in-the-middle" phenomenon and knowledge resolution in specialized financial domains. It offers concrete architectural recommendations for designers of enterprise NLP systems, highlighting that for technical corpora, structural preservation via fixed-size overlapping windows is superior to standard semantic splitting.

Keywords – Retrieval-Augmented Generation (RAG); Document Segmentation; Natural Language Processing; Technical Documentation; LLM Evaluation.

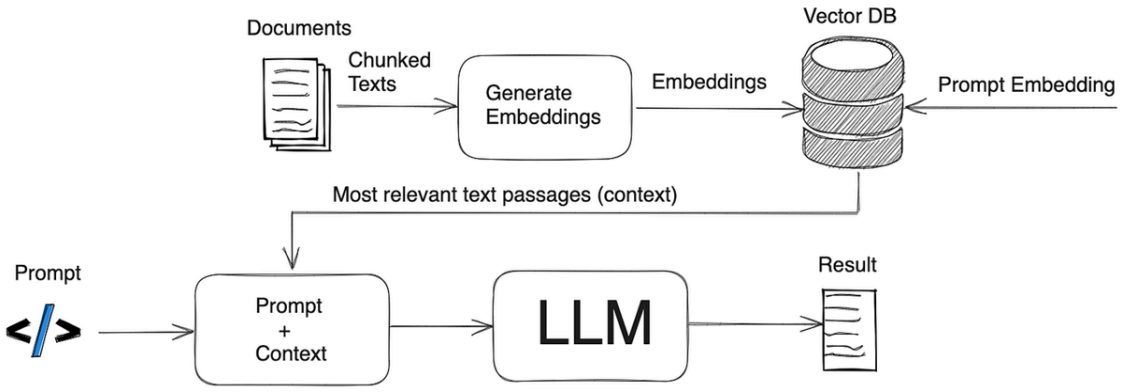
Paper Type – Research paper.

1. Introduction

The field of Natural Language Processing (NLP) has undergone a radical transformation over the past decade, moving from rule-based and statistical models to the era of deep learning. This evolution reached a turning point in 2017 with the introduction of the Transformer architecture, which replaced sequential processing with a self-attention mechanism. This innovation allowed models to analyze relationships between words simultaneously regardless of their distance in a text, laying the foundation for modern Large Language Models (LLMs). These models function primarily through next-token prediction, learning semantic nuances, factual knowledge, and logic from vast datasets during their pre-training phase.

The effectiveness of an LLM is generally defined by its parameters, the internal weights that store complex knowledge, and its context window, which represents the limited "working memory" available for a single prompt. Despite their impressive ability to generate human-like text, LLMs face significant intrinsic limitations. They are prone to hallucinations, where they generate factually incorrect but plausible-sounding information, and they suffer from a "knowledge cutoff," meaning their internal data is frozen at the time of training. Furthermore, they lack access to private or proprietary data, which limits their utility in specialized or secure environments. To address these challenges, Retrieval-Augmented Generation (RAG) was developed as a hybrid framework that integrates the generative power of LLMs with real-time access to external data.

Figure 1: Schematic representation of the RAG architecture pipeline

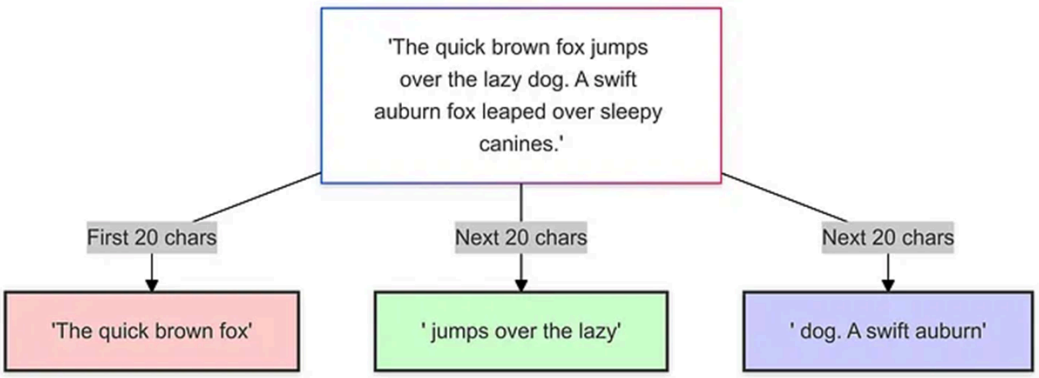


Source:https://learnmycourse.medium.com/retrieval-augmented-generation-rag-process-using-an-llm-339430ff0a05

The RAG process is divided into two main stages: the indexing phase and the inference phase. During indexing, source documents are collected and processed into smaller, logical fragments known as "chunks." These chunks are converted into numerical embeddings and stored in a vector database. During the inference phase, a user's query is similarly embedded to retrieve the most relevant fragments from the database. This context is then provided to the LLM, ensuring that the generated response is grounded in verifiable, up-to-date evidence.

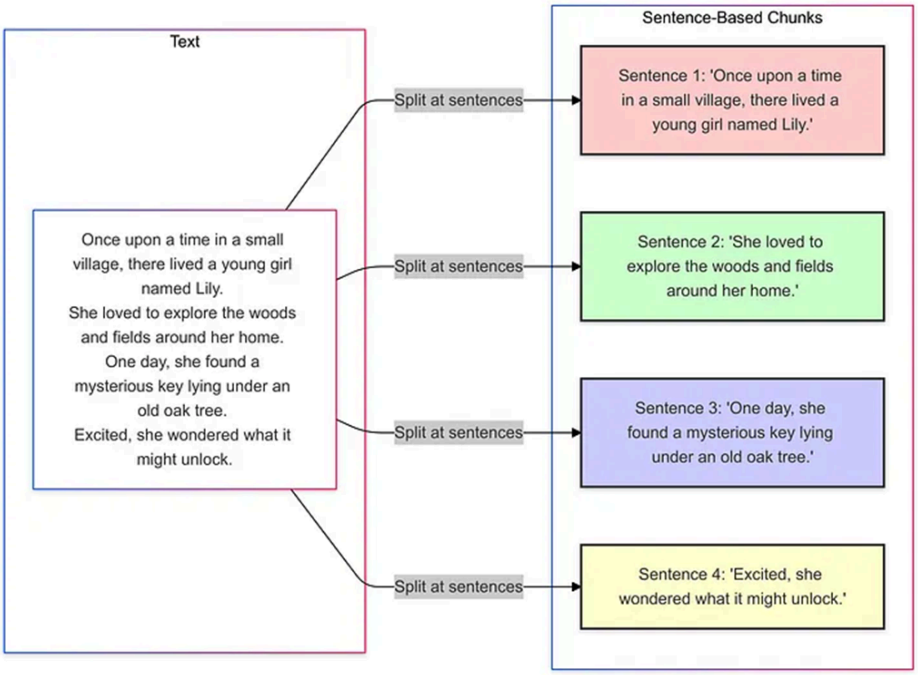
Document segmentation, or chunking, is the critical foundation of this architecture. The quality of the system's output depends directly on whether these fragments retain sufficient semantic meaning. If segments are too large, the meaning becomes diluted, leading to imprecise retrieval; if they are too small, essential context may be lost. Various strategies are employed to optimize this process, ranging from simple fixed-length splitting to more sophisticated structural approaches that respect sentence or paragraph boundaries.

Figure 2: Fixed-length chunking



Source: https://masteringllm.medium.com/11-chunking-strategies-for-rag-simplified-visualized-df0dbec8e373

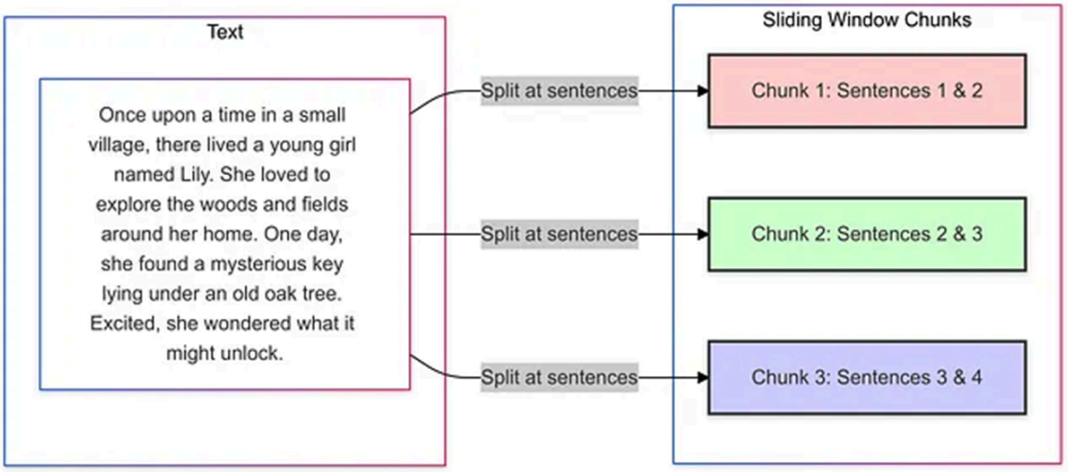
Figure 3: Structural (sentence/paragraph) chunking



Source: https://masteringllm.medium.com/11-chunking-strategies-for-rag-simplified-visualized-df0dbec8e373

To further refine retrieval accuracy, advanced methods like sliding window chunking introduce overlaps between segments to preserve continuity.

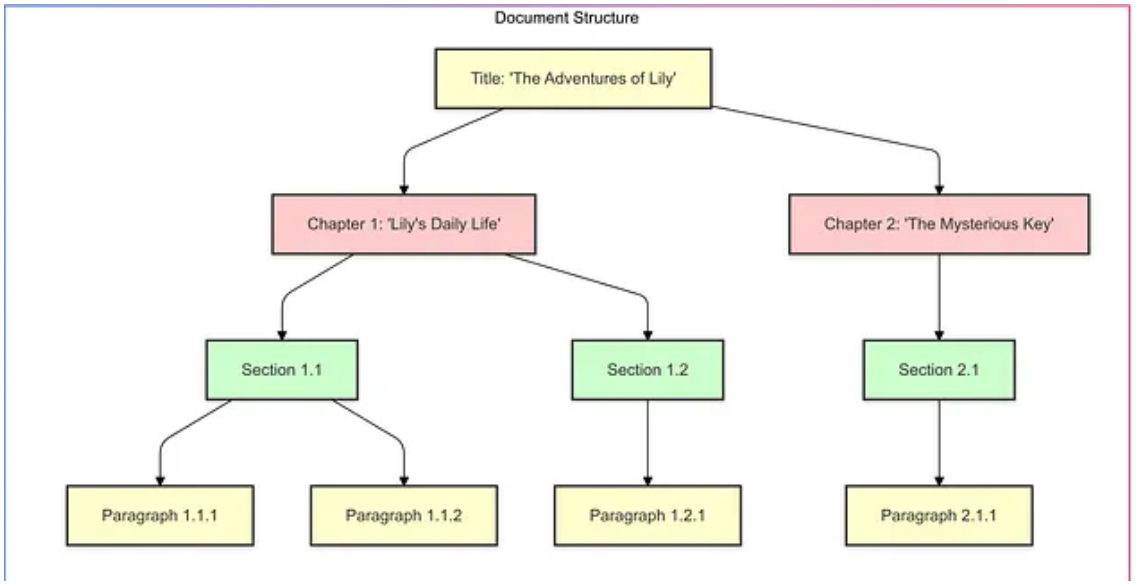
Figure 4: Visualization of the sliding window technique with overlapping text segments



Source: https://masteringllm.medium.com/11-chunking-strategies-for-rag-simplified-visualized-df0dbec8e373

Recursive chunking, often considered the industry standard, utilizes a hierarchy of separators to keep related content together.

Figure 5: Recursive segmentation process using hierarchical separators



Source: https://masteringllm.medium.com/11-chunking-strategies-for-rag-simplified-visualized-df0dbec8e373

Ultimately, document segmentation serves as the invisible gatekeeper of system performance; the choice of chunking strategy determines the semantic resolution of the entire retrieval space. By utilizing a specialized corpus of technical and regulatory documentation regarding the Republic of Croatia's Fiscalization standards, this study provides an empirical evaluation of the trade-offs between fragment granularity, semantic coherence, and retrieval precision. To ground these experiments in current academic discourse, the following section examines existing literature regarding the synergy between parametric and non-parametric components and the architectural limitations of long-context attention.

2. Related work

The analysis of existing literature provides a robust theoretical foundation for evaluating document segmentation strategies within RAG systems, particularly when applied to complex technical and regulatory corpora such as the Republic of Croatia's Fiscalization standards. Central to this field is the work of Lewis et al. (2021), which establishes the synergy between parametric and non-parametric components as a cornerstone for high-performance generative models in knowledge-intensive tasks. However, the efficiency of this synergy is frequently compromised by the "lost in the middle" phenomenon (Liu et al., 2024), where LLMs exhibit significantly degraded performance when relevant information is positioned in the center of long input contexts as opposed to the beginning or end. This architectural limitation underscores the critical importance of effective document segmentation to ensure that relevant technical specifications remain within the model's high-attention zones.

The trade-off between segment granularity and semantic context is a primary focus of current research. Empirical investigations (Bhat et al., 2025) have demonstrated that the optimal chunk size is highly task-dependent, where smaller blocks (64–128 tokens) favor high-precision factoid retrieval, while larger segments (512–1024 tokens) are necessary for capturing broad contextual understanding in descriptive or technical response generation. This finding supports the multi-tiered chunking approach utilized in this study, which employs micro (128), standard (512), and macro (1024) token blocks to address various retrieval requirements within fiscalization datasets. Furthermore, rigid fixed-size chunking can often lead to fragmented semantic content, a methodological issue that Qu et al. (2024) identify as a significant bottleneck. They suggest that while fixed-size methods are computationally efficient, they may be less reliable for topic-diverse data compared to semantic chunking, which uses embedding similarity to determine breakpoints.

To improve retrieval in specialized domains, advanced strategies such as semantic and structure-aware chunking have been developed. Aytar et al. (2024) demonstrated that transitioning from recursive to semantic chunking improves the coherence of vector databases and retrieval accuracy in technical literature. This aligns with the methodology of this research, which employs a semantic breakpoint percentile threshold of 95 to ensure that only significant semantic shifts trigger new segments. Additionally, structural element-based chunking, which prioritizes components like tables and titles, has been shown to significantly improve RAG results in financial and technical reporting (Jimeno Yepes et al., 2024). This structural integrity is further enhanced by techniques like Summary-Augmented Chunking (SAC), which prepends document-level synthetic summaries to each chunk to mitigate document-level retrieval mismatch in structurally similar datasets (Reuter et al., 2025). Similarly, Wang et al. (2025) proposed the use of pseudo-instructions to guide segmentation, ensuring that chunks align with the document's key themes and thereby reducing model hallucinations during generation.

The evaluation of these complex segmentation strategies requires advanced metrics that do not rely solely on human-annotated ground truth. The RAGAs framework (Es et al., 2024) provides a suite of reference-free metrics, including faithfulness, answer relevance, and context relevance, which are essential for tuning the retrieval-generation loop in dynamic environments. While advanced techniques like contextual retrieval enriches segments with document-level summaries to preserve coherence, research indicates they often incur higher computational overhead compared to embedding-time optimizations like late chunking (Merola & Singh, 2025). Collectively, these findings suggest that for the specialized technical

specifications of the Croatian Fiscalization system, a segmentation strategy must balance the structural preservation of PDF data with the semantic demands of retrieval to achieve peak efficiency.

3. Methodology

3.1 Dataset Description

The primary corpus utilized in this study consists of technical specifications and regulatory documentation pertaining to the Republic of Croatia's fiscalization system, with a particular emphasis on the modernized Fiscalization 2.0 standards. These documents, provided in PDF format, encompass a wide range of legislative acts and technical guides, including the Zakon o fiskalizaciji u prometu gotovinom (Fiscalization in Cash Transactions Act) and the Pravilnik o fiskalizaciji (Ordinance on Fiscalization). The corpus includes complex technical instructions for API communication via the Central Information System (CIS), structural data requirements for the Unique Invoice Identifier (JIR) and Security Code (ZKI), and detailed protocols for digital certificate procurement through Public Key Infrastructure (PKI) systems. To evaluate the system’s performance on specialized reporting like non-cash payment fiscalization and "FiskAplikacija" reporting, a structured evaluation dataset was developed in JSON format. This dataset contains 30 "ground-truth" questions and expected answers across categories such as "eRačun basic concepts," "reporting obligations," and "application functionalities," facilitating a quantitative assessment of the RAG pipeline's accuracy.

3.2 Dataset Preprocessing

The preprocessing pipeline transforms raw PDF documents into searchable vector representations using the LlamaIndex framework. Document ingestion is performed via the *SimpleDirectoryReader*, configured to recursively parse input directories and process only .pdf files. Extracted text is subsequently segmented into nodes according to the selected chunking strategy and embedded using the *text-embedding-3-large* model to ensure high-quality semantic representations. The *gpt-4o-mini* model is employed as the primary large language model for both response generation and evaluation tasks.

3.3 Theoretical background

Within the technological stack of the proposed system, RAG is treated as a modular architecture in which retrieval, representation, and generation operate as loosely coupled components. This separation enables independent optimization of document preprocessing, embedding construction, retrieval logic, and response generation, while preserving end-to-end semantic grounding (Lewis et al., 2021). Consequently, system performance depends not only on model capacity but also on how external knowledge is structured prior to retrieval.

A central theoretical assumption of this study is that document segmentation defines the effective knowledge resolution of a RAG system. During preprocessing, raw documents are transformed into discrete semantic units that serve as the atomic elements of retrieval. The size and internal coherence of these units determine how information is distributed within the embedding space and how precisely relevant context can be retrieved at inference time. Overly large segments may dilute semantic relevance, while excessively small segments risk fragmenting logically connected information across multiple retrieval candidates.

This trade-off is further shaped by positional effects inherent to transformer-based language models. Empirical studies show that attention allocation across long contexts is uneven, often prioritizing tokens near the beginning and end of the input (Liu et al., 2024). In this context, overlap mechanisms act as redundancy strategies that increase the likelihood that critical regulatory information remains accessible within high-attention regions of the context window.

Semantic segmentation introduces an alternative theoretical framing in which documents are modeled as trajectories through embedding space rather than linear token sequences. Segment boundaries correspond to statistically significant shifts in semantic similarity, enabling the construction of retrieval units that preserve conceptual coherence. Such approaches aim to reduce semantic fragmentation and improve neighborhood structure within the vector index (Qu et al., 2024), aligning with findings that semantic-aware chunking improves retrieval stability in technical corpora (Aytar et al., 2024).

Finally, evaluation in RAG systems is conceptualized as a semantic alignment problem rather than a surface-form matching task. Since generated outputs are conditioned on retrieved context, correctness depends on the consistency between retrieved evidence, generated responses, and reference knowledge. LLM-based evaluation frameworks operationalize this alignment by modeling evaluation itself as a reasoning task, enabling automated assessment of grounding, relevance, and factual correctness (Es et al., 2024).

3.4 System Implementation and Experimental Setup

The system was implemented using a modular configuration approach to support systematic comparison across different architectural settings. A centralized *RAGConfig* class defines shared parameters such as model selection, directory paths, and environment variables, ensuring reproducibility across all experimental runs. Data orchestration and retrieval are managed through the LlamaIndex framework, with the *VectorStoreIndex* used to maintain document-to-node mappings and similarity-based retrieval.

The experimental setup employs two OpenAI models: *text-embedding-3-large* for high-dimensional vector representations and *gpt-4o-mini* as the primary generator. To mitigate API rate limitations and ensure stable inference behavior, a fixed delay of 0.5 seconds was enforced between successive queries. All generated outputs, including response latency and evaluation scores, were aggregated into structured CSV files for downstream statistical analysis.

3.4.1 Chunking Strategy Logic

The core experimental focus of this study is the impact of document segmentation on retrieval accuracy and response quality. Segmentation strategies are implemented through the *build_index* function, which dynamically selects between rigid fixed-size segmentation using *SentenceSplitter* and context-aware segmentation via *SemanticSplitterNodeParser*.

For fixed-size strategies, three granularities are evaluated: micro (128 tokens), standard (512 tokens), and macro (1024 tokens). This selection directly follows empirical findings that smaller chunks are more effective for precise, fact-based queries, while larger segments better support responses requiring broader technical context (Bhat et al., 2025). To mitigate information loss at segment boundaries and address the lost in the middle phenomenon (Liu et

al., 2024), overlapping windows ranging from 20 to 200 tokens are applied depending on segment size.

The semantic strategy employs a dynamic boundary detection mechanism based on embedding distance between consecutive sentences. A *breakpoint_percentile_threshold* of 95 and a *buffer_size* of 1 ensure that new segments are created only when statistically significant semantic shifts occur. This approach directly addresses fragmentation issues associated with rigid segmentation and aligns with prior evidence supporting semantic-aware chunking in technical documentation (Qu et al., 2024; Aytar et al., 2024).

3.4.2 Evaluation via LLM-as-a-Judge

Following the RAGAs evaluation paradigm proposed by Es et al. (2024), system performance was assessed using an automated LLM-as-a-Judge framework designed to capture grounding, relevance, and factual correctness beyond lexical overlap. The evaluator was implemented as a separate OpenAI client and configured to return structured JSON outputs to ensure deterministic parsing and scoring consistency across runs.

Evaluation was conducted using the custom dataset of 30 ground-truth questions related to Fiscalization 2.0 and eRačun standards. Generated responses were scored on a continuous scale from 0.0 to 1.0 across three semantic dimensions derived from the RAGAs framework: 1) faithfulness - assessing whether responses are grounded exclusively in the retrieved context, 2) context relevancy - measuring the focus and sufficiency of retrieved segments for answering the query, and 3) answer correctness - evaluating factual alignment between generated outputs and expected regulatory answers.

This evaluation methodology ensures that specialized fiscal terminology, including JIR, ZKI, and AMS, is evaluated based on semantic grounding rather than surface-form overlap, reflecting findings that context-aware retrieval and evaluation are essential for accurate handling of domain-specific financial terminology in RAG systems (Jimeno Yepes et al., 2024).

4. Results

This section explains the results of our experiments on the RAG pipeline. Different ways of cutting documents into pieces were tested, focusing on how these methods affect the quality of answers about the fiscalization system. By looking at the performance of seven different strategies, we can see which settings work best for technical and legal documents.

4.1 Analysis of Accuracy and Strategy Rankings

The first part of our analysis focuses on how well the system finds information and creates correct answers. According to the Metrics Comparison (Figure 6), the strategies that used a medium chunk size of 512 tokens performed much better than the very small or very large ones. For example, the strategy with 512 tokens and a 100-token overlap performed very well at finding the right context and staying "faithful" to the source documents. Interestingly, the Semantic Chunking method, which uses AI to decide where to cut the text, had the lowest scores in almost every category. This shows that for technical documents such as the ones used in this study, fixed sizes are often more reliable than "smart" cutting methods. These results are confirmed in the Overall Ranking (Figure 7), which combines all scores into one

final grade. The 512-token strategies clearly hold the top positions, with the *512 Size, 100 Overlap strategy* being the overall winner with a score of 0.550.

Table 1: Performance metrics across document segmentation strategies

Strategy	***Faithfulness***	***Context Relevancy***	***Answer Correctness***	***Latency (s)***	***Average Score***
512 Size, 100 Overlap	*0.515*	*0.657*	*0.478*	*3.41*	*0.550*
512 Size, 50 Overlap	*0.517*	*0.627*	*0.500*	*3.22*	*0.548*
1024 Size, 200 Overlap	*0.473*	*0.610*	*0.493*	*3.16*	*0.526*
1024 Size, 100 Overlap	*0.457*	*0.583*	*0.447*	*3.82*	*0.496*
128 Size, 20 Overlap	*0.440*	*0.540*	*0.433*	*2.63*	*0.471*
128 Size, 40 Overlap	*0.423*	*0.567*	*0.403*	*2.62*	*0.464*
Semantic Chunking	*0.393*	*0.583*	*0.343*	*3.33*	*0.440*

Figure 6: Metrics Comparison

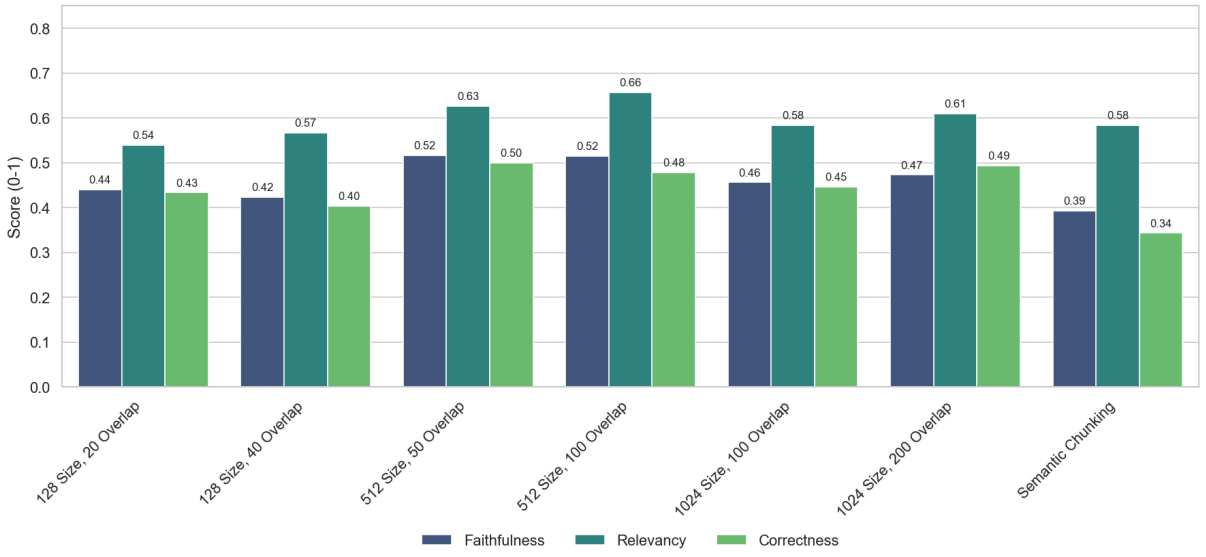
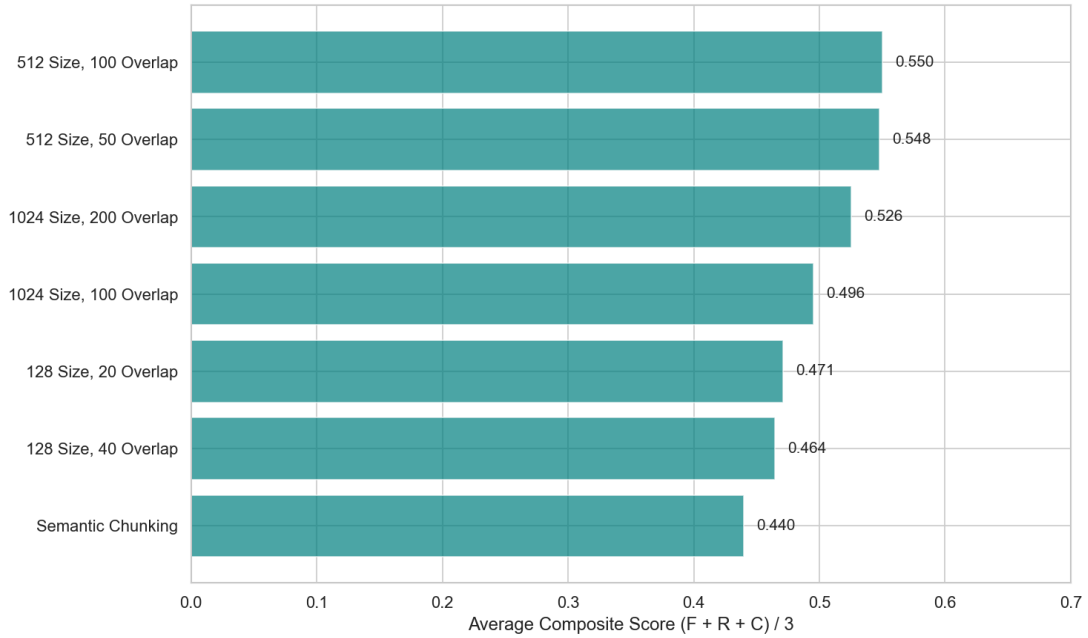


Figure 7: Overall Ranking



4.2 Speed versus Quality and Context Retrieval

In a real-world application, the system has to be fast as well as accurate. The Latency vs. Accuracy (Figure 8) chart shows the trade-off between the time it takes to answer and the quality of that answer. It can be seen that smaller text pieces make the system faster, but they often lack the details needed for a correct answer. The strategies with 1024 tokens were the slowest, taking nearly 4 seconds to answer, but they did not provide better results than the 512-token versions. This makes the *512 Size, 50 Overlap strategy* the "sweet spot" for this system, as it provides high accuracy in about 3.2 seconds. Finally, the connection between finding the right text and giving a correct answer was analyzed. In the Relevancy vs. Correctness (Figure 6) chart, a clear trend can be seen: when the system finds more relevant text, the final answer is usually more accurate. This is especially true for the 512-token strategies, which successfully find and use the specific technical information required for fiscalization reports.

Figure 8: Latency vs. Accuracy

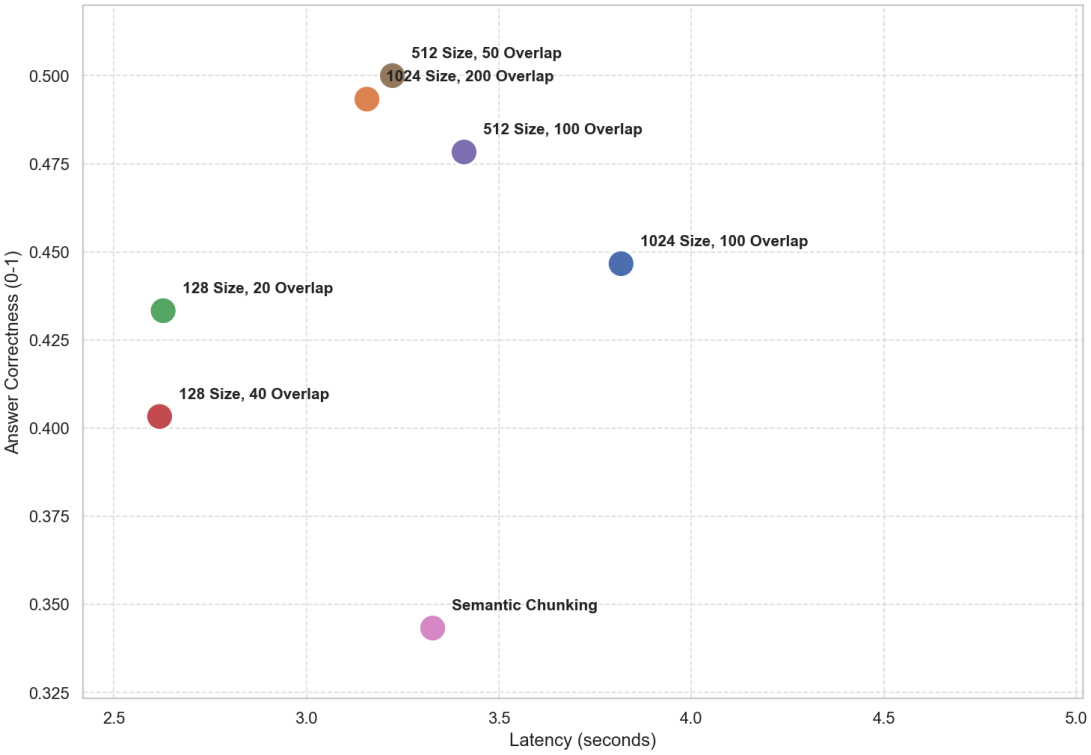
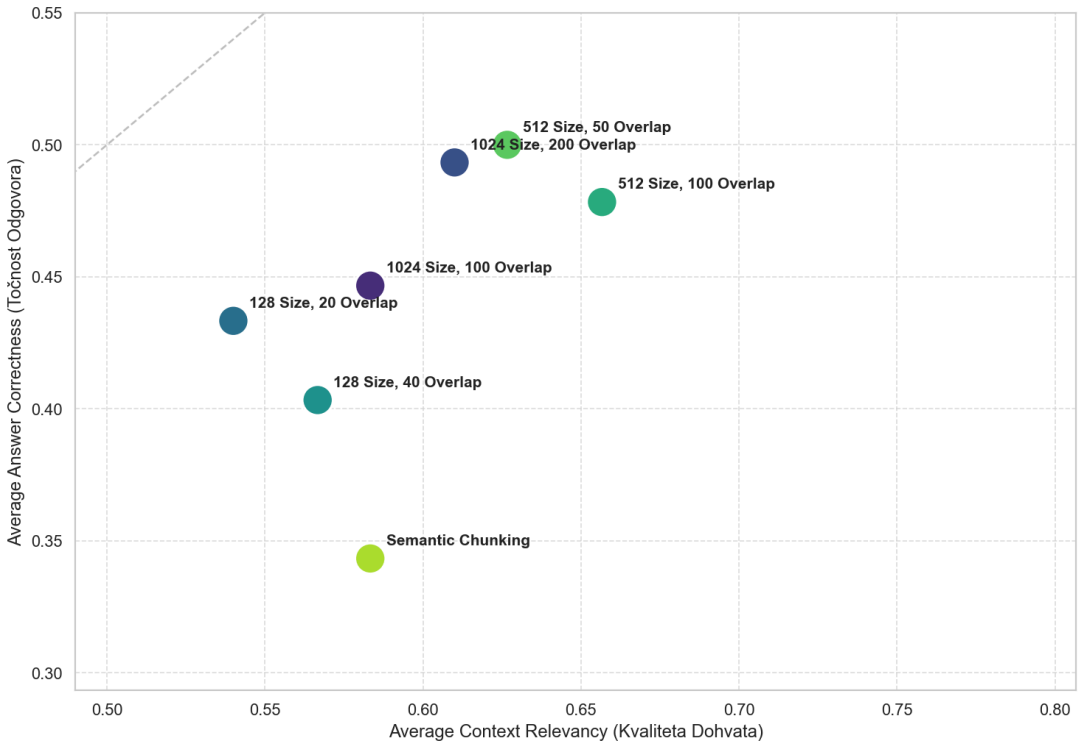


Figure 9: Relevancy vs. Correctness



5. Discussion

5.1 Conclusions

The primary objective of this research was to investigate the impact of document segmentation strategies on the efficiency and accuracy of RAG systems when processing the complex technical and regulatory documentation of the Croatian fiscalization system. Based on the experimental results, document segmentation serves as the critical foundation for RAG architecture because the quality of the system's output depends directly on whether these fragments retain sufficient semantic meaning. The findings indicate that the 512-token chunk size is the optimal configuration for this specific technical corpus. Specifically, the strategy using a 512-token size with a 100-token overlap achieved the highest overall ranking with a composite score of 0.550.

Contrary to expectations that advanced methods would perform better, the Semantic Chunking approach, which uses AI to decide where to cut text based on embedding similarity, yielded the lowest scores in almost every category. This suggests that for highly structured technical and legal PDF files, simple fixed-size strategies are more reliable and effective than complex semantic methods. Furthermore, a clear trade-off between speed and quality exists; smaller text segments make the system faster but often lack the details needed for a correct answer, while segments larger than 512 tokens increase latency without improving accuracy. Consequently, the 512-token size with a 50-token overlap represents the "sweet spot" for this system, providing high accuracy in approximately 3.2 seconds. These findings are particularly relevant for regulatory environments where maintaining semantic grounding is essential for accurately handling specialized terminology.

5.2 Theoretical implications

This study provides empirical support for several core theoretical frameworks in Natural Language Processing. First, the results validate the assumption that document segmentation defines the effective knowledge resolution of a RAG system by transforming raw documents into discrete semantic units that serve as the atomic elements of retrieval. Second, the superior performance of configurations with significant overlap reinforces the "lost-in-the-middle" theory (Liu et al., 2024), which posits that LLMs exhibit degraded performance when relevant information is positioned in the center of long input contexts. Overlap mechanisms act as a redundancy strategy to increase the likelihood that critical information remains accessible within high-attention regions of the context window. Third, the findings regarding semantic chunking contribute to the theoretical debate over representational coherence. While semantic segmentation aims to preserve conceptual coherence by modeling documents as trajectories through embedding space (Qu et al., 2024), the poor results in this study suggest there may be diminishing returns for such methods when applied to technical corpora where semantic shifts are less pronounced. Finally, the use of the RAGAs framework (Es et al., 2024) reinforces the conceptualization of RAG evaluation as a semantic alignment problem rather than a surface-form matching task. By utilizing an LLM-as-a-Judge, the system captured nuances in faithfulness, relevance, and correctness that standard lexical metrics would likely miss.

5.2 Practical implications

For practitioners designing RAG systems for enterprise or regulatory use, this research offers several concrete recommendations. When building systems for technical reporting or government APIs, engineers should prioritize standard 512-token chunks over micro or macro

sizes to ensure the best balance of speed and accuracy. Overlap should be considered more critical than raw chunk size, as it directly mitigates information loss at segment boundaries and addresses positional attention biases.

For organizations focused on financial reporting or compliance, adopting fixed-size segmentation can reduce hallucinations and improve grounding by providing consistent contextual fragments to the LLM. In scenarios where latency is a primary constraint, micro-chunks (128 tokens) may be used to increase speed, but developers must accept a corresponding decrease in answer correctness. Conversely, if high accuracy is the priority, the computational overhead of standard chunk sizes is fully justified by the significant gains in context relevancy and faithfulness.

5.3 Limitations and future research

Several limitations should be considered when interpreting these results. The study's focus on Croatian fiscalization documents limits the generalization of the findings to more informal or less structured corpora. Additionally, the results are dependent on the specific models used, *gpt-4o-mini* and *text-embedding-3-large*, and other model architectures might exhibit different sensitivities to chunking strategies. The evaluation scope was also limited to 30 ground-truth questions centered on factual and procedural queries rather than open-ended reasoning or creative tasks.

Future research should explore the effectiveness of these segmentation strategies on multilingual datasets and less structured document types. Investigating adaptive or query-aware segmentation could lead to more dynamic RAG architectures that adjust chunking based on the complexity of the user query. Furthermore, future studies could extend the evaluation framework to include human-in-the-loop validation and investigate the interaction between document segmentation and advanced techniques like late chunking or contextual retrieval methods.

References

1. Aytar, A. Y., Kaya, K. and Kilic, K. (2024). A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science. Sabanci University, Istanbul, Turkey. [Online]. Available: https://arxiv.org/abs/2412.15404v1

2. Bhat, S. R., Spiekermann, J., Rudat, M. and Flores-Herr, N. (2025). Rethinking Chunk Size for Long-Document Retrieval: A Multi-Dataset Analysis. Fraunhofer IAIS, Germany. [Online]. Available: https://arxiv.org/abs/2505.21700v2

3. Es, S., James, J., Espinosa-Anke, L. and Schockaert, S. (2024). RAGAs: Automated Evaluation of Retrieval Augmented Generation. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 150-158. [Online]. Available: https://aclanthology.org/2024.eacl-demo.16/

4. Jimeno Yepes, A., You, Y., Milczek, J., Laverde, S. and Li, L. (2024). Financial Report Chunking for Effective Retrieval Augmented Generation. Unstructured Technologies, Sacramento, CA, USA. [Online]. Available: https://arxiv.org/abs/2402.05131v3

5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. and Kiela, D. (2021). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv preprint arXiv:2005.11401v4. [Online]. Available: https://arxiv.org/abs/2005.11401v4

6. Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F. and Liang, P. (2024). Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, vol. 12, pp. 157-173. [Online]. Available: https://doi.org/10.1162/tacl_a_00638

7. Merola, C. and Singh, J. (2025). Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation. Proceedings of the Second Workshop on Knowledge-Enhanced Information Retrieval, ECIR 2025. [Online]. Available: https://arxiv.org/abs/2504.19754v1

8. Qu, R., Tu, R. and Bao, F. (2024). Is Semantic Chunking Worth the Computational Cost? Vectara, Inc. & University of Wisconsin-Madison. [Online]. Available: https://arxiv.org/abs/2410.13070v1

9. Reuter, M., Lingenberg, T., Liepiņa, R., Lagioia, F., Lippi, M., Sartor, G., Passerini, A. and Sayin, B. (2025). Towards Reliable Retrieval in RAG Systems for Large Legal Datasets. arXiv preprint arXiv:2510.06999v1. [Online]. Available: https://arxiv.org/abs/2510.06999v1

10. Wang, Z., Gao, C., Xiao, C., Huang, Y., Si, S., Luo, K., Bai, Y., Li, W., Duan, T., Lv, C., Lu, G., Chen, G., Qi, F. and Sun, M. (2025). Document Segmentation Matters for Retrieval-Augmented Generation. Findings of the Association for Computational Linguistics: ACL 2025, pages 8063-8075. [Online]. Available: https://aclanthology.org/2025.findings-acl.422