Legal Knowledge and Information Systems
J. Savelka et al. (Eds.)

doi:10.3233/FAIA241255

Legal Chunking: Evaluating Methods for Effective Legal Text Retrieval

Andrea Filippo FERRARIS [a,1], Davide AUDRITO [a,b], Giovanni SIRAGUSA [b] and Alessandro PIOVANO [c]
[a] *LAST-JD, Alma AI, Alma Mater Studiorum, University of Bologna*
[b] *University of Turin, Computer Science Department*
[c] *University of Turin, Law Department*
ORCiD ID: Andrea Filippo Ferraris https://orcid.org/0009-0004-7487-5560, Davide Audrito https://orcid.org/0009-0003-1832-8135, Giovanni Siragusa https://orcid.org/0000-0002-1797-7956

Abstract. Legal document retrieval is heavily influenced by how documents are segmented, or ”chunked,” for processing within Retrieval-Augmented Generation (RAG) systems. This paper investigates the effectiveness of three automated chunking techniques — Simple Text Splitting, Recursive Text Splitting using regular expressions (regex), and Semantic Chunking — within the legal domain, using the General Data Protection Regulation (GDPR) as a testbed. The chunking methods were evaluated based on their semantic relevance to a set of seventeen legal questions and their corresponding relevant sections, with performance measured using cosine similarity metrics. Results show that none of the methods consistently produced high semantic relevance on an individual chunk level: Git Hub link.

Keywords. Natural Language Processing (NLP), Legal RAG, Chunking Techniques, Legal Document Retrieval, Legal LLM

1. Introduction

The increasing complexity of the legal system, with its growing body of laws, regulations, and rulings, poses challenges for both legal professionals and citizens seeking to understand legal information [1, 2]. While Large Language Models (LLMs) hold promise for improving legal accessibility, they struggle with the unique characteristics of legal texts, including specialized terminology, complex sentence structures, and intricate contextual dependencies [3].

LLMs often face difficulties in accurately interpreting legal terms, managing long, complex sentences with nested clauses [4], and navigating the hierarchical structure of legal documents [5]. One of their main challenges is maintaining the appropriate context throughout [6].

Retrieval-Augmented Generation (RAG) offers a potential solution by combining generative models with retrieval techniques [7]. In RAG systems, how documents are

[1] Corresponding Author: Andrea Filippo Ferraris, Mail: andrea.ferraris3@unibo.it

chunked greatly impacts retrieval performance. Poorly defined chunks can break semantic coherence and hinder the system's ability to retrieve relevant information [8].

This issue is particularly critical in the legal domain, where it is essential to preserve both the logical structure and semantic integrity of texts. Despite this, there is limited research on chunking techniques specifically tailored for legal documents.

This paper aims to fill that gap by evaluating automated chunking methods for legal texts. It explores three approaches designed to improve the accuracy and relevance of legal text processing in RAG systems, contributing to more reliable AI-driven solutions for legal information retrieval.

2. Related Work

Retrieval-Augmented Generation (RAG) represents a breakthrough approach designed to improve Large Language Model (LLM) performance by integrating external knowledge sources, thus expanding their functional capabilities, following various setups and potential uses of this technique [9]. However, the research contributions exploring the application of Retrieval-Augmented Generation (RAG) in legal information retrieval remains limited.

Some research initiatives focus on enhancing case-based reasoning by utilizing datasets of judicial decisions as external knowledge sources [10, 11, 12]. Additional efforts include the development of LegalBench-RAG, a benchmark specifically designed for evaluating legal RAG systems [13], and LexDrafter, a framework aimed at improving legal drafting processes [14]. Within RAG architectures, chunking of documents, the process of dividing text data into smaller, manageable pieces or ”chunks”, plays a crucial role in enhancing the effectiveness of RAG systems by making the retrieval of content more relevant.

2.1. Automated Chunking Techniques

Text chunking is essential for document retrieval, particularly in processing large, complex texts. In the legal domain, where dense terminology, intricate sentence structures, and cross-references are common, chunking becomes even more challenging [15]. Even semi-structured legal texts like EU regulations reflect this complexity, highlighting the need for effective chunking methods. This paper focuses on three key techniques: `Simple Text Splitting`, `Recursive Text Splitting using Regex`, and `Semantic Chunking`.

Simple Text Splitting divides text into uniform chunks based on criteria like word count or character limits, such as splitting the GDPR into 500-word sections. This method is easy to implement and efficient for large-scale processing but lacks semantic awareness, which can fragment legal provisions and disrupt contextual integrity and logical structure. Pak and Lee [16] note that simple splitting struggles with capturing the complexity of legal texts.

Recursive Text Splitting is more adaptable, using punctuation or regular expressions (regex) to segment text based on legal structures like section titles or articles. For instance, regex can split the GDPR at each ”Article \d+” marker, which should ensure that each chunk preserves legal provisions. While this method enhances structural integrity, developing regex patterns is time-consuming and risks overfitting, limiting generalizability.

Semantic Chunking uses natural language processing (NLP) to split text based on meaning rather than length or structure. NLP models, like transformers, should be able to identify logical breaks, creating chunks that preserves legal concept. However, this method is computationally intensive and depends on the quality of the underlying NLP model. Poorly trained models may fail to capture legal context.

3. Methodology

This study examines the semantic produced by three automated chunking techniques in the context of EU regulatory documents, using the General Data Protection Regulation (GDPR, Reg. 679/2016) as a test case. Seventeen questions, selected for their relevance to key legal principles such as data subject rights, lawful processing, and controller obligations, were used to represent a wide range of legal complexities. The chunking methods were tested within a simulated RAG pipeline, with a focus on semantic relevance. To independently evaluate each technique, we applied cosine similarity [17], a widely-used metric in RAG architectures. The evaluation specifically targeted chunking methods, excluding other RAG components. We utilized state-of-the-art metrics [18] and expert reviews [19] to assess the semantic relevance of each chunk in relation to the questions. Future work will expand the query set to cover additional legal frameworks for a more comprehensive analysis.

3.1. Chunking Method Parameters

To assess the performance of each chunking technique, we varied the chunk length and overlap for the Simple Text Splitting and Recursive Text Splitting with Regex methods, while Semantic Chunking determined these parameters automatically. For each question, the K most relevant chunks (defined in $K = 3$) were retrieved, and the semantic proximity between the query and each chunk was noted.

For **Simple Text Splitting**, we used variable chunk lengths of 128, 256, and 512 tokens, with overlap settings of 8 and 16 tokens for each chunk length. This variation allowed us to explore how chunk size and overlap influence the retrieval process, ensuring a balance between capturing sufficient context and avoiding redundant information.

Similarly, for **Recursive Text Splitting with regex**, we applied the same chunk lengths (128, 256, and 512 tokens) and overlaps (8 and 16 tokens). The splitting was performed based on typical punctuation marks found in legal texts, specifically using periods (".") and paragraph breaks as delimiters.

For **Semantic Chunking**, the chunking process was handled automatically by the NLP model, which determined the appropriate chunk length and overlap based on the semantic structure of the text. This method dynamically adjusts these parameters to ensure that each chunk maintains semantic coherence without manual intervention.

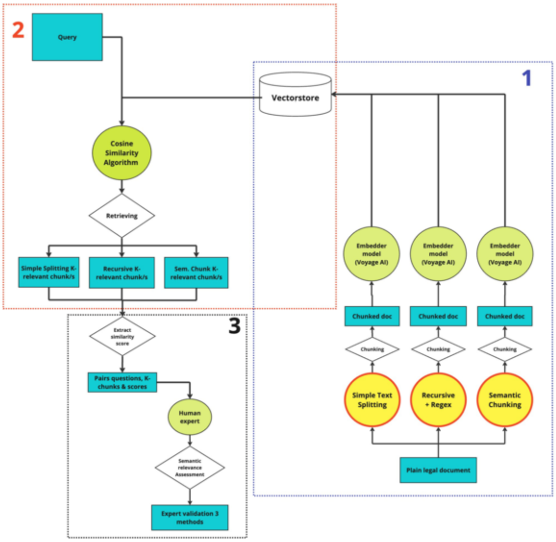
2
Query
Vectorstore
1
Cosine Similarity Algorithm
Retrieving
Simple Splitting K-relevant chunk/s
Recursive K-relevant chunk/s
Sem. Chunk K-relevant chunk/s
Embedder model (Voyage AI)
Embedder model (Voyage AI)
Embedder model (Voyage AI)
Chunked doc
Chunked doc
Chunked doc
3
Extract similarity score
Chunking
Chunking
Chunking
Pairs questions, K-chunks & scores
Human expert
Simple Text Splitting
Recursive + Regex
Semantic Chunking
Semantic relevance Assessment
Plain legal document
Expert validation 3 methods

Figure 1. Visual representation of the workflow

4. Experimental Architecture Setup

This section outlines the experimental setup used to evaluate the effectiveness of different chunking techniques for legal texts within a Retrieval-Augmented Generation (RAG) architecture. In this experiment, we relied on the architecture presented in Figure 1, which is composed by 3 different modules, each marked with a coloured box and pairing number:

- **Module 1 (blue): Chunking and Vector Store creation**
  This module, executed once for each chunking technique, generates three chunked versions of the GDPR, one for each method. We then initialized the `VoyageAIEmbeddings` model,[2], which transforms the text into high-dimensional vectors within a dense vector space to capture semantic meaning. Specifically, we used the `Voyage-2-Law` pre-trained embedding model,[3] designed for legal texts, as it is better suited to handle legal language nuances than general-purpose models. The generated embeddings were stored in the open-source vector database Chroma DB,[4], which the retriever uses in Module 2 to calculate semantic similarity.
- **Module 2 (red): Retriever and Similarity Computation**
  This module tests the effectiveness of the three chunking methods by retrieving the most relevant chunks based on semantic similarity. Using the Chroma vector database, a retriever with a cosine similarity algorithm calculates the closeness between query vectors and the chunked GDPR documents. For each of the 17 questions in the gold standard, the retriever processes all chunked versions, retrieves the top 3 most relevant chunks, and computes similarity scores. The $K = 3$

[2] See: `https://www.voyageai.com/`
[3] See: Voyage-2-Law overview
[4] See: `https://github.com/chroma-core/chroma`

choice balances relevance and evaluation. The process is repeated for all chunking methods, and the results are saved in a CSV file for further analysis.

- **Module 3 (black): Chunk Semantic Relevance Evaluation and Validation** This module evaluates the semantic relevance of the chunks retrieved for each of the 17 questions using cosine similarity from the ChromaDB retriever. The similarity scores provide a quantitative measure of how closely each chunk aligns with the query, and the process is repeated for all three chunking methods. Legal experts then conduct a qualitative assessment, reviewing the overall relevance of the top 3 retrieved chunks per question, and individually evaluating the semantic accuracy of each chunk. This combined approach ensures a thorough evaluation of each method's effectiveness.

5. Results and Discussion

This section presents the results of the chunking method evaluations (the dataset can be accessed here **Git Hub**). Interestingly, none of the tested chunking techniques produced chunks with notably high semantic closeness scores, although in both the simple and recursive approaches, the shorter the chunk size the higher the score (Simple δ = 0.04 % & Recursive δ = 0.03 %).

This suggests that none of the methods, in their current form, are particularly well-suited for legal text retrieval. Simple text splitting produces slightly higher result, but in general the variance in scores between the methods was minimal (max δ = 0.09%), indicating that no single approach demonstrated a significant advantage over the others in terms of semantic relevance. Most interestingly, even presenting a high computation cost, the semantic chunking technique produces the worst results in terms of chunk relevance.

Method	Chunk Size	Chunk Overlap	Score
Simple	128	8	0.4106
	128	16	**0.4267**
	256	8	0.4024
	256	16	0.4031
	512	8	0.3830
	512	16	0.3849
Recursive	128	8	0.4106
	128	16	**0.4122**
	256	8	0.3954
	256	16	0.3971
	512	8	0.3771
	512	16	0.3761
Semantic	-	-	**0.3317**

Table 1. Average scores with chunk size and chunk overlap for different splitting methods.

The low semantic relevance scores likely stem from the complexity of legal texts. Simple and Recursive Splitting may miss provisions spanning sections, while Semantic Chunking, despite its computational cost, struggles with nested clauses, resulting in either too broad chunks or at the opposite, too fragmented chunks. These issues highlight the need for more advanced techniques that capture the hierarchical structure of legal texts.

To improve results, hybrid approaches, such as those explored by Yepes et al. [20], and proposition-based knowledge base reformulation [21] could enhance chunking efficacy. Additionally, a graph- and vector-based approach, known as hybrid RAG [22], should be tested in the legal domain as they represent promising directions for future research.

References

[1] Danièle Bourcier and Pierre Mazzega. Toward measures of complexity in legal systems. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, page 211–215, New York, NY, USA, 2007. Association for Computing Machinery.

[2] John B Ruhl and Daniel Martin Katz. Measuring, monitoring, and managing legal complexity. *Iowa L. Rev.*, 101:191, 2015.

[3] Zhongxiang Sun. A short survey of viewing large language models in legal aspect. *arXiv preprint arXiv:2303.09136*, 2023.

[4] Antonino Rotolo and Giovanni Sartor. Argumentation and explanation in the law. *Frontiers in Artificial Intelligence*, 6, 2023.

[5] Weicong Qin and Zhongxiang Sun. Exploring the nexus of large language models and legal systems: A short survey, 2024.

[6] Mario Ferrer-Benítez. Online dispute resolution: can we leave the initial decision to large language models (llm)? *Metaverse Basic and Applied Research*, 1:23–23, 2022.

[7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[8] Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400, 2024.

[9] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[10] Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. Cbr-rag: case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer, 2024.

[11] Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. Clerc: A dataset for legal case retrieval and retrieval-augmented analysis generation. *arXiv preprint arXiv:2406.17186*, 2024.

[12] Rui Yang. Casegpt: A case reasoning framework based on language models and retrieval-augmented generation. *arXiv preprint arXiv:2407.07913*, 2024.

[13] Nicholas Pipitone and Ghita Houir Alami. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*, 2024.

[14] Ashish Chouhan and Michael Gertz. Lexdrafter: Terminology drafting for legislative documents using retrieval augmented generation. *arXiv preprint arXiv:2403.16295*, 2024.

[15] Peter Butt. *Modern legal drafting: A guide to using clearer language*. Cambridge University Press, 2013.

[16] Irina Pak and Phoey Lee Teh. Text segmentation techniques: a critical review. *Innovative Computing, Optimization and Its Applications: Modelling and Simulations*, pages 167–181, 2018.

[17] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6, 2016.

[18] Brandon Smith and Anton Troynikov. Evaluating chunking strategies for retrieval. Technical report, Chroma, July 2024.

[19] Kehua Feng, Keyan Ding, Kede Ma, Zhihua Wang, Qiang Zhang, and Huajun Chen. Sample-efficient human evaluation of large language models via maximum discrepancy competition, 2024.

[20] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Leah Li. Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131*, 2024.

[21] Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*, 2023.

[22] Bhaskarjit Sarmah, Benika Hall, Rohan Rao, Sunil Patel, Stefano Pasquali, and Dhagash Mehta. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction, 2024.