Rafael Sandrini Guaracho

Development of an Automatic Segmentation Model for Legal Documents

São Paulo, SP

2025

Rafael Sandrini Guaracho

Development of an Automatic Segmentation Model for Legal Documents

Course completion work submitted to the Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo for the award of the degree of Engineer.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Supervisor: Profª Tereza Cristina Melo de Brito Carvalho

São Paulo, SP

2025

Catalogação-na-publicação

Sandrini Guaracho, Rafael
Development of an Automatic Segmentation Model for Legal Documents / R. Sandrini Guaracho -- São Paulo, 2025.
54 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Legal contract analysis 2.Semantic segmentation of legal contracts 3.Document segmentation 4.Boundary detection 5.LLM-based data augmentation I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Acknowledgements

To my parents, Vagner Guaracho and Magda Fernandes Sandrini, for their support, trust, and for being my lifelong sources of inspiration.

To my sister, Maria Clara Sandrini Guaracho, also a graduate of the Universidade de São Paulo, whose example and guidance shaped much of my path through university.

To my family, and especially my grandparents, who have celebrated every milestone with me, whether here or in heaven.

To the professors and staff of Escola Politécnica da USP for their time and commitment to education and mentorship, and in particular to Professor Tereza Cristina Melo de Brito Carvalho for supervising this project.

To the friends, professors, and staff at Télécom Paris, Wiremind, Philips Cardiologs, Datadog, and across Paris, who have contributed to my professional training and personal growth.

To my friends at Escola Politécnica da USP for their partnership throughout the undergraduate years and for life.

To CAEA and Poli Plague for the countless shared moments and enduring camaraderie.

Abstract

This work tackles automatic segmentation and semantic labeling of legal contracts from raw text. The task is formulated as sequential sentence classification and target two outcomes: (i) precise detection of segment boundaries and (ii) assignment of labels from a pre-defined set, such as Contract Title, Preamble, Article, Introduction of Parties, Signature and Exhibit. The dataset couples a small, high-quality gold corpus of annotated contracts (English and French) with a larger unlabeled pool. We introduce two complementary components: a lightweight bi-encoder (pre-trained sentence embeddings + stacked BiLSTM layers with a classification head), and a LLM-based data-augmentation pipeline that generates "silver" annotations via retrieval-augmented prompting under strict I/O schemas. Evaluation spans classification metrics (precision, recall, F2, macro and weighted) and segmentation metrics (boundary F2, Pk, and WindowDiff).

Experiments show that the LLM workflow reliably produces high-quality silver labels. When these are merged with gold, the bi-encoder improves substantially over gold-only training. For example, segmentation weighted F2 rises from very low baselines to competitive values and both Pk and WindowDiff decrease, indicating cleaner boundaries. The result is a functional model and pipeline that (a) demonstrates LLMs as effective label generators and (b) distills their results into a smaller, deployable model. Future directions include scaling unlabeled data and refining retrieval and chunking for silver generation.

Keywords: legal contract analysis; semantic segmentation of legal contracts; document segmentation; boundary detection; LLM-based data augmentation; NLP; retrieval-augmented generation (RAG); F2-score; Pk; WindowDiff;

Resumo

Este trabalho aborda a segmentação automática e a rotulagem semântica de contratos jurídicos a partir de texto bruto. A tarefa é formulada como a classificação sequencial de sentenças e visa dois objetivos: (i) a detecção precisa das fronteiras de segmentos e (ii) a atribuição de rótulos de um conjunto predefinido, tais como Título do Contrato, Preâmbulo, Artigo, Identificação das Partes, Assinatura e Anexo. O conjunto de dados combina um pequeno corpus gold de alta qualidade de contratos anotados (inglês e francês) com um conjunto maior de contratos não anotados. Apresentamos dois componentes complementares: um bi-encoder leve (embeddings de sentenças pré-treinado + camadas BiLSTM empilhadas com uma cabeça de classificação) e um pipeline de aumento de dados baseado em LLM que gera anotações "silver" por meio de prompts com recuperação (retrieval-augmented) sob esquemas rigorosos de E/S. A avaliação abrange métricas de classificação (precisão, recall, F2, macro e ponderada) e métricas de segmentação (F2 de fronteira, Pk e WindowDiff).

Os experimentos mostram que o fluxo com LLM produz rótulos silver de alta qualidade de forma confiável. Quando esses são combinados com o gold, o bi-encoder melhora substancialmente em relação ao treinamento apenas com gold. Por exemplo, o F2 ponderado de segmentação sobe de patamares muito baixos para valores competitivos e tanto Pk quanto WindowDiff diminuem, indicando fronteiras mais limpas. O resultado é um modelo e um pipeline funcionais que (a) demonstram os LLMs como geradores de rótulos eficazes e (b) destilam seus resultados em um modelo menor e implantável. Direções futuras incluem ampliar os dados não anotados e refinar a recuperação e o chunking para a geração de silver.

Palavras-chave: análise de contratos jurídicos; segmentação semântica de contratos jurídicos; segmentação de documentos; detecção de fronteiras; aumento de dados baseado em LLM; NLP; geração aumentada por recuperação (RAG); F2-score; Pk; WindowDiff.

Résumé

Ce travail traite de la segmentation automatique et à l'étiquetage sémantique de contrats juridiques à partir de texte brut. La tâche est formulée comme une classification séquentielle de phrases et vise deux objectifs : (i) la détection précise des frontières de segments et (ii) l'attribution d'étiquettes issues d'un ensemble prédéfini, telles que Titre du contrat, Préambule, Article, Introduction des parties, Signature et Annexe. Le jeu de données associe un petit corpus « gold » de haute qualité de contrats annotés (anglais et français) à un plus grand ensemble non annoté. Nous introduisons deux composants complémentaires : un bi-encodeur léger (représentations de phrases préentraînées + couches BiLSTM empilées avec une tête de classification) et un pipeline d'augmentation de données basé sur un LLM qui génère des annotations « silver » via un prompt enrichi par récupération sous des schémas d'E/S stricts. L'évaluation couvre des métriques de classification (précision, rappel, F2, macro et pondérée) et de segmentation (F2 aux frontières, Pk et WindowDiff).

Les expériences montrent que le flux LLM produit de manière fiable des étiquettes « silver » de haute qualité. Lorsqu'elles sont fusionnées avec les annotations « gold », le bi-encodeur s'améliore sensiblement par rapport à un entraînement sur « gold » seul. Par exemple, le F2 pondéré de segmentation passe de niveaux très bas à des valeurs compétitives et Pk comme WindowDiff diminuent, indiquant des frontières plus nettes. Il en résulte un modèle et un pipeline fonctionnels qui (a) démontrent l'efficacité des LLM comme générateurs d'étiquettes et (b) distillent leur résultat dans un modèle plus petit, déployable. Les pistes futures incluent la mise à l'échelle des données non annotées et l'affinement de la récupération et du découpage en fenêtres pour la génération « silver ».

Mots-clés : analyse de contrats juridiques ; segmentation sémantique de contrats juridiques ; segmentation de documents ; détection de frontières ; augmentation de données basée sur un LLM ; NLP ; génération augmentée par recherche (RAG) ; F2-score ; Pk ; WindowDiff.

List of Figures

List of Tables

List of abbreviations and acronyms

LLM	*Large Language Model*
RAG	*Retrieval-Augmented Generation*
API	*Application Programming Interface*
JSON	*JavaScript Object Notation*
XML	*eXtensible Markup Language*
ETA	*Estimated Time of Arrival*
NFC	*Unicode Normalization Form C*
OCR	*Optical Character Recognition*
GPU	*Graphics Processing Unit*
CPU	*Central Processing Unit*
BERT	*Bidirectional Encoder Representations from Transformers*
XLM-R	*XLM-RoBERTa (cross-lingual RoBERTa)*
LSTM	*Long Short-Term Memory*
BiLSTM	*Bidirectional Long Short-Term Memory*
GRU	*Gated Recurrent Unit*
BiGRU	*Bidirectional Gated Recurrent Unit*
IOB	*Inside–Outside–Begin*
BCE	*Binary Cross-Entropy (loss)*
ICF	*Inverse Class Frequency (class weighting)*
F2	*F-measure with β=2*
Pk	*Beeferman's Pk segmentation error*
WindowDiff	*WindowDiff segmentation error*

List of symbols

Σ	Summation
TP	True positive
FP	False positive
FN	False negative

Glossary

Legal Document Structure

Article Numbered clause or section in a contract that defines rights, duties, prices, deadlines, penalties, and similar elements. Articles may contain sub-articles (for example, 3.1, 3.2).

Contract Title (Title / Heading) Main heading of the contract, usually on the first page, that names the type of agreement, such as "Service Agreement" or "Insurance Contract".

Exhibit Attachment at the end of a contract (appendix, schedule, annex) that contains supporting material, for example technical details, price tables, or forms.

Introduction of Parties Part of the preamble that explicitly identifies who is signing the contract (companies or individuals) and sometimes their roles, such as "Provider" and "Client".

Preamble Introductory section that explains the context, motivations, date, place, and often a short summary of the agreement before the detailed clauses.

Reference Number / Date Administrative identifiers printed on contracts (protocol numbers, internal reference codes) and the official date of issuance or signature.

Signature / Signature Block Final section where the parties sign the document. It usually includes names, roles, locations, and dates for each signatory.

Table of Contents List of sections, articles, and exhibits with their page numbers, usually at the beginning of longer contracts.

Core Task Concepts (Segmentation and Labels)

Segment Continuous block of lines in a contract that shares the same label or set of labels, for example one Article segment or one Exhibit segment.

Boundary / Segment Boundary Position in the document (for example, a specific line) where one segment ends and another begins.

Boundary Detection / Segmentation Task of deciding where each contract segment starts and ends.

Category Assignment / Classification Task of deciding what a given piece of text represents (for example, Article, Exhibit, Preamble) after or together with segmentation.

Begin Label Special marker that indicates the first line of a new segment for a given label, for example a line marked as the beginning of an Article or an Exhibit.

Over-segmentation Situation where the model predicts too many boundaries, producing smaller segments than necessary.

Machine Learning and NLP Concepts

BiGRU (Bidirectional Gated Recurrent Unit) Recurrent neural network similar to LSTM but with a different internal structure. It reads the sequence in both directions.

BiLSTM (Bidirectional Long Short-Term Memory) Recurrent neural network that reads sequences both forward and backward. It allows the model to use information from previous and following lines when classifying a given line.

BERT (Bidirectional Encoder Representations from Transformers) Transformer-based model widely used in natural language processing to create contextual word and sentence representations.

Binary Cross-Entropy (BCE) Loss Loss function used when each label is a yes or no decision. For each label and line, the model predicts a probability, and BCE measures how far this probability is from the correct answer (0 or 1).

Class Imbalance Situation where some labels appear very frequently (for example, Article) and others very rarely (for example, Reference Number / Date).

Embedding / Sentence Embedding Numeric vector that represents the meaning of a sentence. Sentences with similar meaning have embeddings that are close to each other in the embedding space.

Few-shot Learning Technique where a large language model is given only a few labeled examples of a task inside the prompt and is then asked to label new, similar data.

Gold Dataset / Gold Training Set Subset of contracts that were manually annotated with high quality.

Inverse Class Frequency (ICF) Weighting scheme where rare labels receive higher weight in the loss function so that the model pays more attention to them during training.

IOB Scheme (Inside–Outside–Begin) Labeling format where each line is tagged as B–Label (first line of a segment), I–Label (continuation of a segment), or O (outside any labeled segment).

Large Language Model (LLM) Very large neural network trained on massive amounts of text to understand and generate natural language.

Layer Normalization Operation applied inside neural network layers to normalize intermediate activations and help stabilize training.

Mean Pooling Operation that averages the token vectors of a sentence to form one single sentence embedding.

Multiclass Classification Scenario where each line receives exactly one label from a fixed set, such as only Article or only Exhibit.

Multilabel Classification Scenario where a line can receive several labels at the same time (for example, a line can be both Article and Exhibit).

`paraphrase-xlm-r-multilingual-v1` Pre-trained multilingual model, based on XLM-RoBERTa, used to generate sentence embeddings in many languages, including English and French.

Retrieval-Augmented Generation (RAG) Technique where a large language model receives not only the prompt but also retrieved examples or passages from a database, improving accuracy and reducing hallucinations.

Regular Expression (Regex) Text pattern used for simple automatic operations such as cleaning spaces, extracting tags, and validating structured text.

Sentence-Transformers Library built on top of Transformers that provides models specialized in generating sentence embeddings, such as `paraphrase-xlm-r-multilingual-v1`.

Sigmoid Activation Non-linear activation function that maps any real value to a number between 0 and 1.

Silver Dataset / Silver Annotations Dataset labeled automatically by a large language model.

Sliding Window / Window Size Technique where the document is processed in overlapping chunks of fixed length (for example, 128 sentences). The window "slides" over the document, often with overlap, to keep context and avoid cutting important structures.

Softmax Activation Function that converts a vector of scores into a probability distribution that sums to one.

Transformer Encoder Part of a transformer model that converts a sequence of tokens into contextualized vectors.

XLM-R / XLM-RoBERTa Multilingual transformer model trained on many languages. In this project, a variant of XLM-R is used to generate sentence embeddings.

Evaluation Metrics

Confusion Matrix Table that shows, for each class, how many times it was predicted correctly and how many times it was confused with other classes.

F_2-score Harmonic mean of precision and recall that gives more weight to recall. It penalizes false negatives more strongly than false positives, which is important when missing a boundary is worse than adding an extra one.

Macro Metric (Macro Precision, Macro Recall, Macro F_2) Metric where each class contributes equally, regardless of how many examples it has. Useful to evaluate the model on rare labels.

Precision Of all the predictions the model made for a class, the proportion that is correct. High precision means few false positives.

Recall Of all the true instances of a class, the proportion that the model correctly found. High recall means few false negatives.

Support (Class Support) Number of true examples of a class in the dataset. Used when computing weighted metrics.

Weighted Metric (Weighted Precision, Weighted Recall, Weighted F_2) Metric where each class is weighted by its support (its frequency). Frequent classes have more influence on the final value.

Pk Segmentation error metric that uses a sliding window to check whether pairs of positions fall in the same segment in both the prediction and the ground truth. Lower Pk means better segmentation.

WindowDiff Segmentation metric similar to Pk, but it compares how many boundaries fall inside a window in the prediction versus the ground truth. Lower WindowDiff means better segmentation.

f

Contents

1 Introduction

1.1 Motivation

In the contemporary legal landscape, the large volume and complexity of legal documents, such as contracts, have increased severely. Legal professionals and companies face significant challenges in managing, analyzing and extracting meaningful information from these documents efficiently and accurately. Traditional document analysis methods are often laborious, time-consuming and susceptible to human error, highlighting the need for advanced technological solutions.

A critical aspect of legal document management is the segmentation and classification of documents. This process is based on splitting a long text into manageable and meaningful chunks, such as sections, articles and exhibits. The task is inherently complex due to the legal texts' hierarchical structure and specialized language. Hierarchical segmentation and classification, where documents are organized into multiple levels of categories, add another layer of difficulty. Models must be able to differentiate small changes and maintain contextual integrity across different chunks, ensuring that each segment is accurately identified and classified within the broader structure of the document.

1.2 Objectives

The objectives of this dissertation are two-fold. First, we aim to reliably detect the exact boundaries of every logical segment present in a contract using nothing more than its raw, layout-free text, so that no clause, exhibit, or header is omitted or split incorrectly. Second, once these segments are delimited, we seek to assign each one to a fixed set of semantic categories (e.g., Contract Title, Preamble, Article, Exhibit), enabling downstream queries and analytics. While robust performance is desired for both segmentation and classification, the work prioritizes the segmentation task: minimizing false or missing boundaries is essential to prevent cascading errors that could undermine any subsequent legal review, compliance check, or information-extraction step built on top of the produced chunks.

1.3 Justification

Automating the segmentation and labeling of contracts addresses a clear operational gap: legal teams spend substantial time identifying clauses, comparing versions, and checking compliance, yet manual review remains error-prone and expensive. A system

that reliably marks every boundary and assigns a consistent, audit-ready label to each segment can shorten review cycles, lower costs, and reduce the risk of overlooking critical provisions that might trigger financial or regulatory penalties. Because many organizations handle multilingual and stylistically heterogeneous documents, any practical solution must generalize across formats while keeping inference latency and hardware requirements within the limits of a typical corporate environment.

From a research perspective, the task is equally compelling. It combines hierarchical segmentation and multilabel classification under conditions of label scarcity and class imbalance, an ideal test bed for studying how automatically generated silver data can boost model generalization without inflating production costs. Prioritizing a lightweight bi-encoder over large language models at inference time ensures reproducibility and facilitates future extensions to new label sets or languages.

1.4 Structure of the Work

The work carried out and the studies performed to achieve the objectives outlined above are described in the following chapters. Chapter 2 - Conceptual Aspects presents the theoretical foundations, challenges and key concepts necessary to understand the problem; Chapter 3 - Methodology details the dataset preparation, labeling protocol, proposed bi-encoder model, data-augmentation strategy and the evaluation protocol; Chapter 4 - Requirements Specification lists functional and non-functional requirements and acceptance criteria; Chapter 5 - Development documents the implementation and engineering choices; Chapter 6 - Results reports quantitative and qualitative findings and analyses; and Chapter 7 - Conclusions and Future Work, summarizes the contributions, limitations and proposed next steps.

This work is the result of a bi-national effort: the core bi-encoder architecture and its initial training routines were conceived and prototyped at Télécom Paris during my double-degree program, while the LLM-based data-augmentation pipeline, which enabled the best bi-encoder results, was designed, implemented, and validated in Brazil. The final integration and large-scale experiments consolidate the contributions from both phases.

2 Conceptual Aspects

This chapter grounds the dissertation in the core concepts and challenges that shape the proposed model for legal documents segmentation and classification. It first surveys the structural properties of legal documents and clarifies the distinction between boundary detection and category assignment. It then examines the data constraints that complicate model training and reviews the metrics that best capture the task quality. The final section positions the work within the existing body of research.

2.1 Hierarchical Structure of Legal Contracts

Legal contracts are inherently layered documents. Figure 1 illustrates a typical hierarchical structure, beginning with a Contract Title and Preamble, followed by an Introduction of Parties, and then progressively nested Articles and Sub-Articles. After these core sections, contracts usually conclude with Signatures and Exhibits (appendices, schedules, annexes).

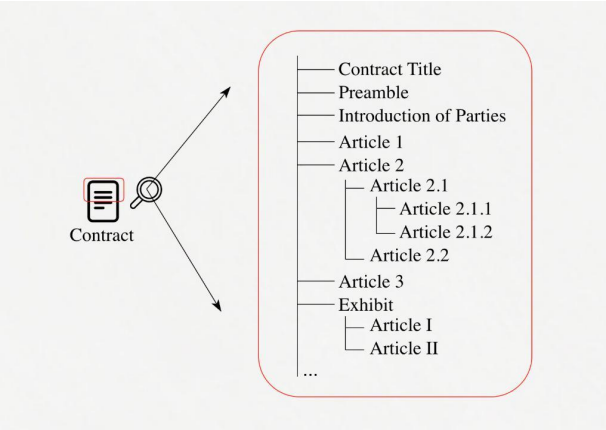


Figure 1 – Illustration of the hierarchical structure of a legal contract

The hierarchy is only partially signaled by typography and numbering. For example, a “Section 3.2(a)” may appear visually similar to “Article 4” in a different template, and an Exhibit can span dozens of pages without any explicit closing marker. Moreover, layouts vary across jurisdictions and even across departments within a single organization. These inconsistencies undermine rules that rely solely on regular expressions or fixed heading patterns. A robust system therefore needs to infer boundaries from a combination of lexical cues, numbering schemes, and contextual dependencies that persist across the entire document.

2.2 Segmentation and Classification

The contract-processing task splits naturally into two subtasks — identifying exact segment boundaries and assigning semantic labels to each chunk:

- Segmentation (boundary detection) aims to decide where each segment starts and ends. Errors at this stage propagate: if a boundary is missed, any downstream label is attached to the wrong text span.
- Classification then decides what a segment represents. A single span may carry multiple roles (e.g., a line can simultaneously start an Article and belong to a larger Exhibit), so the classifier must handle these cases.

Treating the two subtasks jointly is attractive because contextual signals useful for one can improve the other. Yet decoupling them conceptually clarifies evaluation priorities: this work gives precedence to boundary accuracy, since false or missing splits can invalidate all subsequent analysis irrespective of label correctness.

2.2.1 Multilabel

Multilabel classification means that each line (or segment) can receive more than one label at the same time. This is common in contracts: a single sentence may simultaneously begin an Article and belong to an Exhibit for example. This approach is implemented as independent binary predictions per label — sigmoid outputs with a BCE (Binary Cross-Entropy) loss, which allows the model to return sets like `Article, Exhibit` for the same text unit. Multilabel is appropriate when overlapping segments are frequent and the goal is to preserve all applicable assignments rather than force a single exclusive label.

2.2.2 Multiclass

Multiclass labeling assigns exactly one label per unit. When used for segmentation, this is often implemented with the IOB scheme (Inside–Outside–Begin)(RAMSHAW; MARCUS, 1995): each sentence/token receives a tag such as `B-Article`, `I-Article` or `O`. The IOB approach is implemented with a softmax head and cross-entropy loss and is suitable when the task can be linearized into a single label sequence. As noted earlier, the multilabel step handles overlapping roles, while the multiclass step with IOB marks segments explicitly (e.g., `B-Article`). When overlaps occur (for example, Articles inside Exhibits), the IOB scheme can be designed to preserve the innermost label, so the `Article` tags remain on sentences that are both inside an Exhibit and form an Article.

Figure 2 exemplifies the IOB multiclass and multilabel approach in a contract.

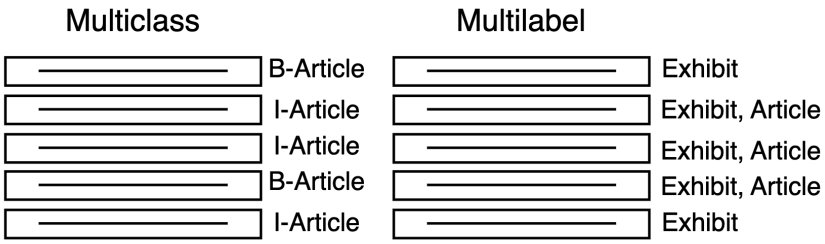


Figure 2 – IOB multiclass and multilabel contract example

2.3 Label Scarcity and Class Imbalance

High-quality line-level annotations are scarce because they require domain experts and are time-consuming, as annotating a long contract can take hours. Consequently, available datasets are typically small and heavily biased: frequent labels (e.g., Article) dominate while rare ones (e.g., Reference/Date, Signature) occur only sporadically — an expected pattern in legal corpora where far more lines are classified as Article than as Contract Title, for example. This imbalance biases models toward majority classes and degrades performance on underrepresented labels. Techniques such as oversampling, loss reweighting, and data augmentation can help, but they also risk introducing or amplifying noise when rare examples are inconsistent.

2.4 Evaluation Metrics

Evaluation will target two complementary aspects of the model: label quality and boundary quality. For classification (label quality) we will report standard measures such as precision and recall. Precision quantifies how many predicted labels are correct, while recall measures how many true labels are found, we then aggregate them per-class and with appropriate averages. For segmentation (boundary quality) we will rely on windowed metrics that capture disagreement in boundary placement across the document, most notably P_k and `WindowDiff`: both slide a fixed window across the text and compare reference vs. hypothesis about whether points belong to the same segment or how many boundaries lie inside the window, penalizing missed, and shifted boundaries in complementary ways. More implementation details and the complete evaluation protocol are given in Chapter 3.

2.5 Related Work

The segmentation and classification of legal documents have progressed significantly over time, transitioning from rule-based methods to deep learning approaches. Early methods relied on rule-based systems and traditional NLP libraries like NLTK, as seen in tools such as LexNLP (II; KATZ; DETTERMAN, 2018). These approaches focused

on explicit markers like titles, numbering, and headings for feature extraction and can be effective for structured legal document segmentation. However, these methods depend on predefined linguistic patterns and struggle with less structured texts, such as legal contracts that often do not follow a very specific format.

The introduction of Conditional Random Fields (CRFs) provided a step forward by modeling text as sequences of tokens or sentences. These models started using handcrafted features to identify segment boundaries and assign semantic labels such as rhetorical roles (e.g., facts, arguments, statutes) (MALIK et al., 2022) and later were combined with embedding models, such as sent2vec (MOGHADASI; ZHUANG, 2020) and BERT (DEVLIN et al., 2019) to improve phrase or token representations. Finally, CRF layers were combined with a Recurrent Neural Network (RNN) such as BiLSTM encoders to extract contextualized embeddings with local and global context. However, CRF-based methods are inherently ill-suited for highly imbalanced classes, often necessitating additional preprocessing techniques like oversampling, which may not always be practical. Moreover, they are restricted to multiclass classification, limiting their applicability in more complex scenarios.

Meanwhile, Large Language Models (LLMs) have shown remarkable ability to perform few-shot learning (BROWN et al., 2020), adapting to new tasks with minimal labeled examples provided at inference time. Such models can effectively "learn on the fly" by reading only a few examples prompts paired with expected outputs. This paradigm reduces the need for extensive task-specific datasets and is highly flexible across a wide range of language tasks.

Retrieval-Augmented Generation (RAG) (LEWIS et al., 2021) represents another major development, especially for knowledge-intensive NLP tasks. RAG couples a parametric model (e.g., a large language model) with an external retrieval mechanism. Instead of relying solely on pre-trained parameters, RAG accesses detailed background information from a non-parametric memory and then conditions the model on these retrieved passages. By splitting the system into a retriever (which fetches relevant content) and a generator (which synthesizes a response), RAG can provide accurate and interpretable outputs even under heavy domain-specific demands.

Recently, fine-tuning LLMs and employing retrieval-based augmentation have become popular approaches, as they enhance model performance on specialized tasks. This is the case in the multi-label sequential sentence classification task (LAN et al., 2024) where an LLM can be fine-tuned with context-augmented prompts to classify sequences of sentences, exploiting the extensive pre-training of LLMs and avoiding the costly annotation of large datasets. Some limitations of LLM-based approaches include a restricted context window and high memory requirements when deploying models locally, as memory usage scales exponentially with token count, even with distilled or medium-sized models (LAN

et al., 2024). Additionally, fine-tuning LLMs via APIs can be financially expensive.

Our approach addresses specific challenges encountered in our legal document segmentation task, such as managing highly imbalanced classes, working with limited GPU resources for both training and inference, and handling the inherent complexity of loosely structured legal texts. Given these constraints, we formulated our task as a sequential sentence classification problem within a multilabel framework, which naturally accommodates overlapping classes. To effectively address these issues, we introduced a novel LLM-based data-augmentation pipeline alongside a bi-encoder classification pipeline optimized for computational efficiency on limited GPU hardware.

2.6 Chapter Considerations

This chapter introduced the main characteristics of legal contracts, the challenges involved in identifying their structure and related work to this subject. This discussion helps explain why segmentation is difficult and why a learned model is needed. These points naturally lead into Chapter 3, where the dataset and the model used to address these challenges are presented.

3 Methodology

This chapter describes the concrete artifacts and procedures used to implement and evaluate this work: the dataset and its preparation, the labeling rules that map segment annotations into line-level targets, the conceptual design of the proposed bi-encoder, the data-augmentation workflow based on retrieval + LLM, and the evaluation protocol that makes experiments reproducible and comparable.

3.1 Dataset and preparation

This section describes the dataset used in this work and the steps applied to prepare the documents for training. It summarizes where the contracts come from, how the text was extracted and cleaned, and how the documents were split into lines. The goal is to show how raw, heterogeneous contracts were transformed into a consistent line-level format that the model can use.

3.1.1 Sources, languages and formats

The dataset combines a small, high-quality gold set of human-annotated contracts with a larger set of unlabeled documents. The gold set contains 51 contracts (23 French, 28 English), and the unlabeled pool contains 200 documents (100 French, 100 English), where each document is a text file (.txt) and each line is a sentence extracted from the original contract files. The annotation of the dataset was done using JSON, and each object has three attributes that define a segment to be predicted: the starting line of the segment, the end line of the segment, and the label name assigned to that segment. The segments can be overlapped in such a way that each line of the contract can belong to multiple segments at the same time. Figures 3 and 4 show a gold set contract example.

```
1  CONTRAT DE CONDITIONS DE LIVRAISON STANDARD
2  Numéro de Contrat :
3  Le présent Contrat, constitué des conditions générales et des conditions particulières,
4  LE GESTIONNAIRE DU RESEAU DE DISTRIBUTION- GRD
5  Nom ou raison sociale : Veolia Eau - Compagnie Générale des Eaux
6  Adresse : 00 Xxxx xx Xxxxx XX 000 - 00000 XXXXXXXX XXXXX
```

Figure 3 – Plain text legal document example

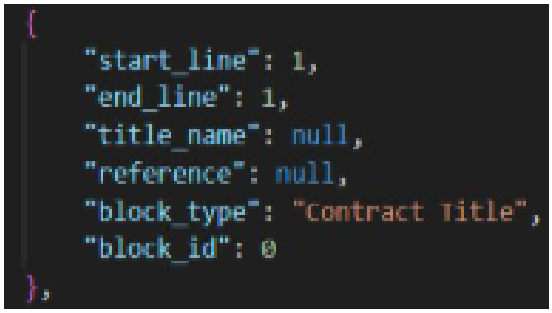


Figure 4 – Label block legal document example

Figure 5 shows the frequency of labels in the annotated dataset, and it is possible to notice a strong class imbalance, which poses an additional challenge to learn good representations for all classes.

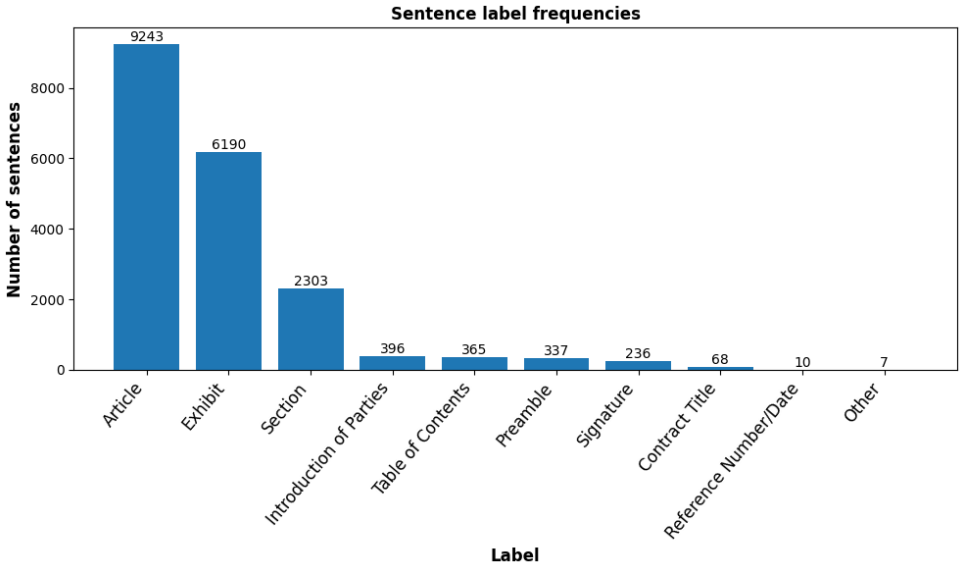
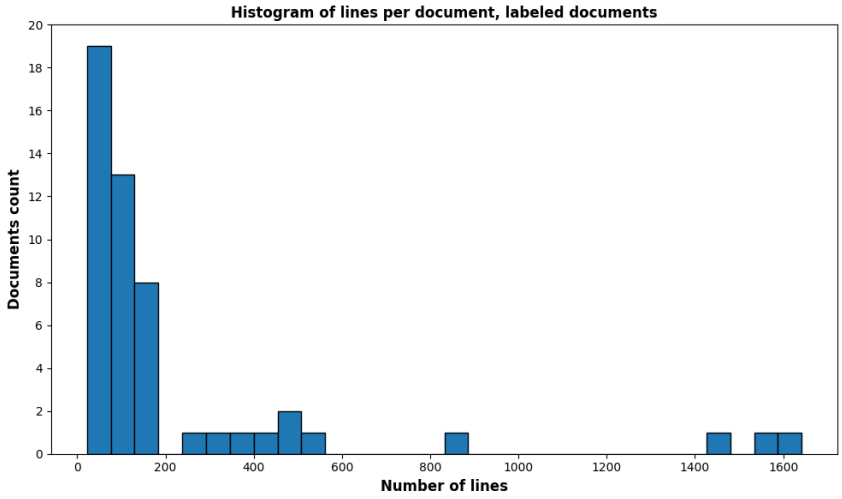


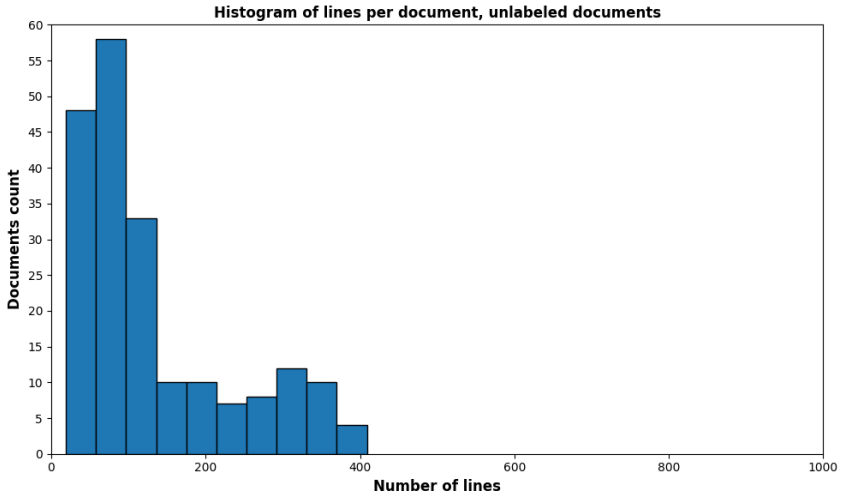
Figure 5 – Label frequencies across labeled documents, ordered from most frequent to less frequent

3.1.2 Statistics and distribution

Most of contract documents have less than 200 lines, as can be seen in Figure 6, and this poses an additional challenge in learning long-term dependencies between labels to segment long contracts. In addition to that, some labels can be very frequent in specific documents and absent in others, which makes it difficult to split the dataset into training, validation, and testing sets with similar label distributions.



(a) Histogram for labeled documents



(b) Histogram for unlabeled documents

Figure 6 – Histograms of lines per document for labeled and unlabeled document sets.

3.1.3 Cleaning and normalization

The raw corpus arrived largely pre-cleaned, so the pre-processing stage was intentionally light. The principal cleaning step performed was the removal of blank lines and normalization of line endings. Unicode was normalized (NFC) and a small set of common OCR artifacts (ligatures, repeated punctuation) were corrected where obviously noisy, but broader character-stripping was avoided: experiments showed that removing special characters degraded contextual cues. Punctuation such as colons, periods, commas and hyphens often signal structural information (e.g., headers, key–value pairs, list items), so these tokens were preserved to help the model infer boundaries and label roles.

3.1.4 Split (train / validation / test)

We call our manually annotated dataset the "gold dataset" and we split it into training (80 %), validation (10 %) and testing sets (10 %) using a manually set random seed. The distribution of labels in each set is highly dependent on single contracts, which makes it difficult to implement a stratified split. In Figure 7, we can see that the distributions of splits in the gold dataset are significantly different, which is taken into consideration when evaluating performance.

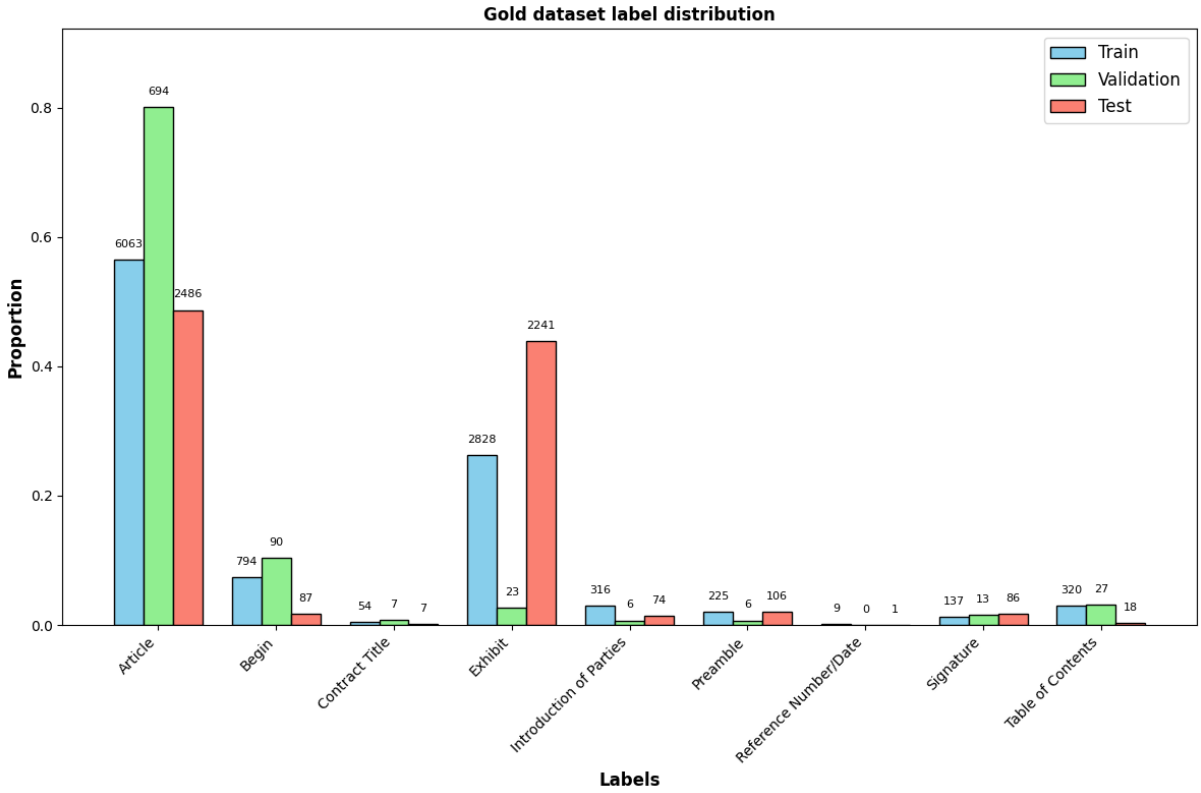


Figure 7 – Distribution of labels in manually annotated training, validation and testing sets

3.2 Label scheme and boundary policies

The annotated contracts were originally labeled by segments, where each segment is defined by a starting line, an ending line and the assigned class. To perform multilabel sequential sentence classification, we implemented an annotation procedure where we assign one set of labels to each contract line (sentence) as described below:

- For each contract line, find all segments that contain the respective line.
- Add each of the segment labels to the line set of labels.
- If the line corresponds to the beginning of a segment, we add an extra label `Begin` to its set of labels to distinguish contiguous segments.

In Figure 8, we can see the multilabel mapping example, from the original label blocks to the multilabel classification for each sentence.

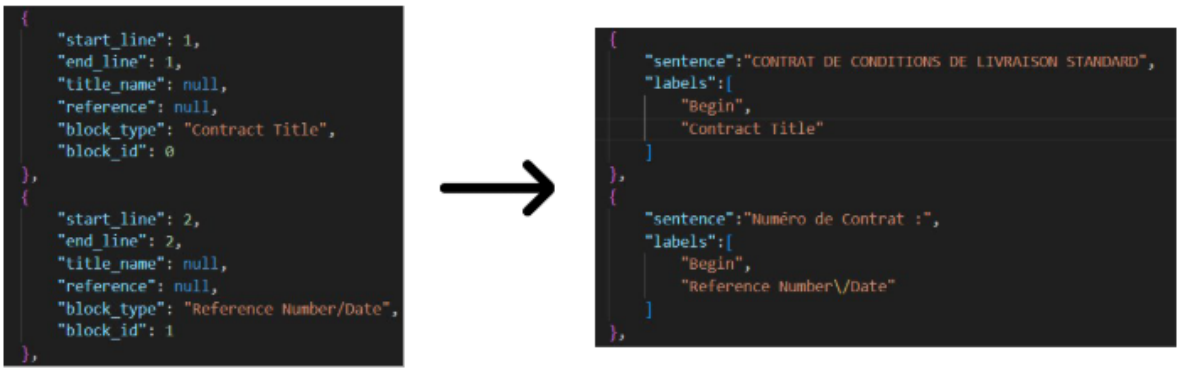


Figure 8 – Multilabel mapping example

As mentioned in Chapter 2, we also explored the multiclass labeling scheme, assigning labels using Inside–Outside–Begin (IOB) markers (e.g., `B-Article`) to each sentence (RAMSHAW; MARCUS, 1995), ensuring that the innermost label is maintained within the nested structure if any segment overlapping is present (for example, we keep Article labels if we have Articles inside Exhibits segments).

3.3 Proposed model — Bi-encoder

This section explains the proposed bi-encoder model used for segmentation and classification. It introduces how the model encodes each line together with its surrounding context, how the two embeddings are combined, and how the segmentation and multilabel heads operate.

3.3.1 Overview

The proposed architecture is a bi-encoder with two stages: (1) a pre-trained transformer encoder that extracts sentence embeddings with in-phrase context; (2) a BiLSTM encoder with a classification head that models contracts as sequences of sentences, extracting contextualized document-level embeddings.

3.3.2 Components

- Encoder (sentence context): pre-trained multilingual sentence encoder produces dense embeddings per line.
- Encoder (document context): 3-layer bidirectional LSTM with layer normalization and dropout consumes the line embeddings and yields contextualized hidden states.

- Classification head: a fully connected layer that outputs a logit for each label, followed by an activation to produce probabilities — sigmoid for multilabel (independent binary probabilities) or a softmax for multiclass (mutually exclusive labels).

3.4 LLM-based data-augmentation

Considering the limited availability of annotated legal data, LLM-based data-augmentation provides a scalable way to increase both the volume and diversity of training examples without extensive manual labeling. Specifically, a large language model can be used to classify the unlabeled documents provided, which can then be integrated into the training set to help downstream models better learn domain-specific patterns, reduce overfitting, and improve generalization.

We then generate labels for our unlabeled dataset using a GPT-4o LLM-based classification pipeline that will be described in Chapter 5. We call the LLM labeled dataset the "silver dataset" and we can see its label distribution in Figure 9. When optimizing the LLM classification pipeline, we observed that the model was frequently replacing "Article" label with "Section", so we decided to remove "Section" label from our label set, prioritizing "Article" labeling.

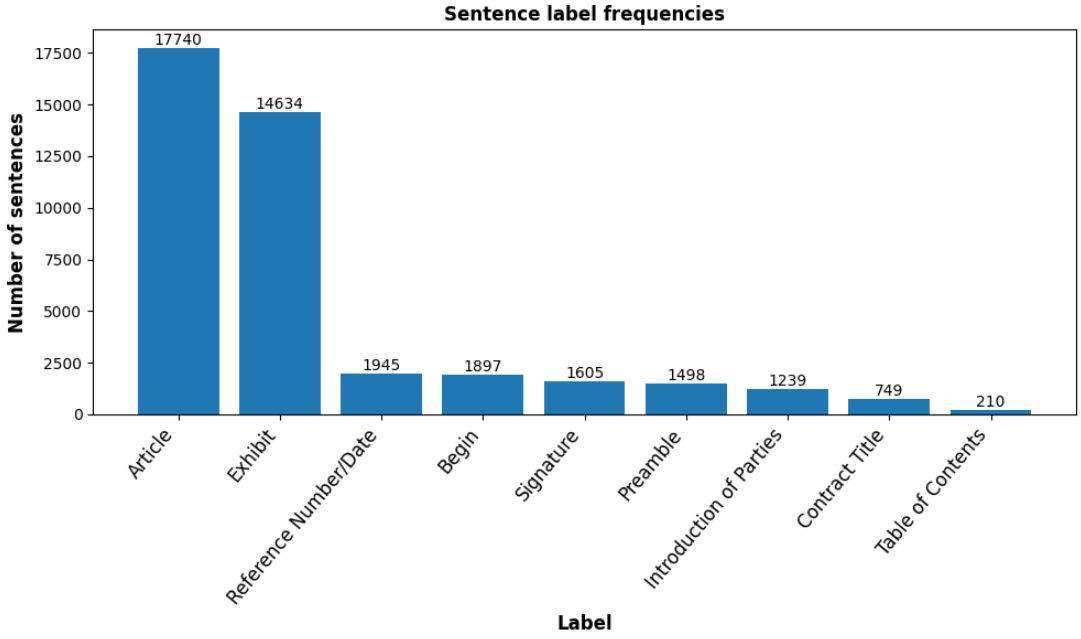


Figure 9 – Distribution of labels in the LLM-labeled silver dataset of contracts

3.5 Evaluation protocol

As classification metrics, we compute macro and weighted versions of precision, recall and F2-score.

$$\text{Precision: } \frac{TP}{TP + FP} \tag{3.1}$$

$$\text{Recall: } \frac{TP}{TP + FN} \tag{3.2}$$

$$\text{F2-score: } \frac{5 \text{ Precision} * \text{Recall}}{4 \text{ Precision} + \text{Recall}} \tag{3.3}$$

F2-score penalizes false negatives (missed labels/boundaries) more strongly than false positives. Because a missed segment start can compromise all subsequent analysis of a contract, this work prioritizes recall: the system is tuned to prefer over-segmentation (extra predicted boundaries) rather than risk omitting or merging true beginnings.

- Macro: computes the metric separately for each class and then averages the per-class scores without weighting. In other words, every class counts equally, regardless of how many examples it has.
- Weighted: computes the metric per class and then takes a support-weighted average, where each class is weighted by its number of true instances. Frequent classes therefore have greater influence.

$$\text{Macro: } \frac{1}{C} \sum_{i=k}^{C} Metric_k \tag{3.4}$$

$$\text{Weighted: } \sum_{i=k}^{C} \frac{n_i}{N} Metric_k \tag{3.5}$$

We also compute the confusion matrices. We evaluate the model in the segmentation task by comparing the ground truth and predicted boundaries, where a boundary is defined as the line index of the start or end of a segment. We extract the boundaries separately for each class and we compute as segmentation metrics macro and weighted F2-scores. As F2-score punishes too much the small changes in boundary predictions, we also compute windowed segmentation error metrics such as weighted Pk (equation 3.6) and weighted window difference score (equation 3.7), which are less strict to exact boundary positions.

$$\text{Pk} = \sum_{k=1}^{C} W_k \left(\frac{1}{N - DL_k} \sum_{d=1}^{D} \sum_{i=1}^{N_d - L_k} \mathbf{1}\{b_{d,k}(i) \neq \hat{b}_{d,k}(i + L)\} \right) \tag{3.6}$$

$$\text{WindowDiff} = \sum_{k=1}^{C} W_k \left(\frac{1}{N - DL_k} \sum_{d=1}^{D} \sum_{i=1}^{N_d - L_k} \mathbf{1} \left\{ \left| \sum_{j=i+1}^{i+L_k} b_{d,k}(j) - \sum_{j=i+1}^{i+L_k} \hat{b}_{d,k}(j) \right| > 0 \right\} \right) \tag{3.7}$$

where

- N is the total number of document lines.
- C is the number of classes excluding the "Begin" class.
- D is the total number of documents.
- N_d is the number of document lines in document d.
- L_k is the window size for class k, set to half the average segment length.
- W_k is the weight for class k. For the macro score,$W_k = \frac{1}{C}$, for the weighted score $W_k = \frac{S_k}{\sum_{k=1}^{C} S_k}$, where $S_k = \sum_{d=1}^{D} \sum_{i=1}^{N_d} b_{d,k}(i)$ is the segmentation support of class k, representing the total number of document lines where a true boundary for class k is present.
- $b_{d,k}(j)$ is the true segmentation indicator at position j for document d and class k (usually 1 if there is a boundary, 0 otherwise).
- $\hat{b}_{d,k}(j)$ is the predicted segmentation indicator at position j for document d and class k.

3.6 Chapter Considerations

This chapter detailed the dataset, normalization steps, labeling rules, and the design of both the bi-encoder and the LLM-based augmentation pipeline. As these components define how the model learns and how data flows through the system, they also set the basis for specifying what the system must deliver. Consequently, Chapter 4 formalizes these expectations into functional and non-functional requirements that reflect the capabilities, constraints, and design decisions introduced here.

4 Requirements Specification

This chapter lists the functional and non-functional requirements that guided the design and implementation of the contract segmentation and classification pipeline. The division between functional (FR) and non-functional (RNF) requirements allows us to distinguish the essential capabilities of the system from the expected quality properties: functional items describe what the system must do, non-functional items constrain how it must behave.

4.1 Functional requirements

The functional requirements of the system are:

- **RF1 – Document ingestion:** Accept raw contracts in common format of (plain text .txt, text extracted from .pdf/OCR) and produce a normalized plain-text representation broken into lines for downstream processing.
- **RF2 – Line-level segmentation:** The model should be able to identify segment boundaries at line granularity and mark line(s) that begin segments (Begin tag or B-marker).
- **RF3 – Line-level classification:** The model should be able to assign one or more semantic labels to each line from a predefined label vocabulary.
- **RF4 – LLM-based data-augmentation:** The system should support a controlled workflow to generate silver labels from unlabeled documents using a LLM-based pipeline.
- **RF5 – Model training and inference:** The system should provide a training pipeline for the proposed bi-encoder trained using gold and silver data, exposing standard controls for hyperparameters and able to run inference over documents and output final segments and labels in standard formats.
- **RF6 – Post-processing and heuristics:** The system should provide configurable post-processing rules with switchable options or configuration flags.
- **RF7 – Metrics:** The system should compute evaluation metrics for both segmentation and classification, producing standard reports for model validation and comparison.

4.2 Non-functional requirements

The non-functional requirements of the system are:

- **RNF1 – Resource constraints:** Training and inference should be feasible on a single GPU (NVIDIA L4 GPU for example).
- **RNF2 – Latency:** The model should be able to segment and classify a contract in less than 10 seconds.

4.3 Chapter Considerations

This chapter presented the functional and non-functional requirements that guide the system's development. These requirements describe what the system must do and the constraints under which it must operate. With these definitions in place, Chapter 5 follows by describing how the system was constructed and implemented in practice, connecting each requirement to concrete tools, architectural choices, and development steps.

5 Development

This chapter presents the development of the system, covering how the different components were implemented and integrated. It describes the overall workflow: data ingestion, preprocessing, model training, inference, and evaluation and explains the main engineering choices made during the project.

5.1 Tools

This section describes the main technologies, libraries, and tools used in the bi-encoder model and in the LLM-based data-augmentation pipeline.

5.1.1 Bi-encoder

5.1.1.1 Python

The bi-encoder model is written in Python because it strikes the right balance between productivity and performance for machine learning. Python glues together the full pipeline: data ingestion, preprocessing, embedding generation, model training, evaluation, and plotting.

5.1.1.2 PyTorch

PyTorch supports the model and training loop, providing tensors, automatic differentiation, GPU acceleration, and a clean module system for composing networks. This makes it a natural fit for sequence modeling over sentence embeddings: the framework handles batching, masking, and backprop cleanly while keeping model components easy to swap.

5.1.1.3 NumPy

NumPy supports fast vectorized operations. It is ideal to hold precomputed sentence embeddings, to compute statistics such as label supports and class weights, and to perform lightweight preprocessing (e.g., reshaping, masking, concatenating arrays) before handing tensors to PyTorch.

5.1.1.4 Transformers and Sentence-Transformers

The Hugging Face stack (Transformers / sentence-transformers) supplies pre-trained, multilingual encoders that map sentences to fixed-size vectors. Advantages include

standardized tokenization, GPU support, and easy model swapping.

5.1.1.5 TorchMetrics

TorchMetrics offers batched metric computation with consistent definitions across epochs. It integrates with PyTorch tensors, supports masking and simplifies code for tracking precision, recall, and F-scores during training and validation.

5.1.1.6 TQDM

Long epochs over many windows benefit from lightweight progress reporting. TQDM wraps data loaders and training loops to provide responsive progress bars, iteration rates, and ETA, reducing friction when monitoring experiments and helping catch stalls or data issues early.

5.1.1.7 Optimizer

Optimization uses AdamW, which decouples weight decay from the gradient update and tends to be stable for sequence models.

5.1.1.8 Scikit-learn (metrics)

For the final test-time metrics, scikit-learn's metrics module is used alongside the training-time TorchMetrics. It offers rich, CPU-friendly summaries such as per-class precision/recall/F2-score, weighted and macro aggregates, confusion matrices, and classification reports that are easy to serialize into tables.

5.1.2 LLM-based data-augmentation

5.1.2.1 Python

The LLM-based data-augmentation pipeline is also implemented in Python, which offers an expressive syntax and a mature ecosystem for text processing and scientific computing. Python's standard library covers file I/O, JSON, regular expressions, and lightweight concurrency, allowing the augmentation workflow to remain compact and readable while delegating numerically intensive steps to specialized libraries.

5.1.2.2 OpenAI API

The official client exposes a stable, well-documented API for LLMs. It supports structured prompts, deterministic settings (e.g., temperature, seed when available), streaming, and token accounting.

5.1.2.3 Tiktoken

Tiktoken is a fast tokenizer aligned with OpenAI models. It provides precise token counts for prompts and responses, enabling budget-aware windowing and preventing context overflows. Because counts reflect the actual model tokenizer, we can size prompts confidently and forecast costs and latency more accurately.

5.1.2.4 Pandas

Pandas supplies high-level data frames for tabular manipulation. It simplifies consolidating raw text, model outputs, and metadata into consistent records. Export to JSON is straightforward, which helps preserve intermediate artifacts for inspection and reproducibility.

5.1.2.5 Regular Expressions and Fuzzy Matching

Regex remains the most direct tool for input sanitation and output parsing when formats are semi-structured. It handles whitespace normalization, tag extraction, and format validation. Light fuzzy matching (e.g., token-level diffs) can be layered on top to align near-matches without imposing a heavyweight alignment library.

5.1.2.6 TQDM

Again, TQDM provides responsive progress bars with iteration rate and ETA. It works in terminals and notebooks and adds immediate visibility to long-running stages such as retrieval and API calls, aiding monitoring and early failure detection.

5.1.2.7 Threading

Python's built-in thread is sufficient when throughput is limited by network I/O rather than CPU. Threading was used to improve the time for batched API requests.

5.2 Project and implementation

This section details how the project was structured and implemented in practice. It outlines the organization of the modules, how data flows through the system, and how configuration, training, and inference were designed.

5.2.1 Bi-encoder

5.2.1.1 Architecture

The bi-encoder architecture uses two encoders: one for the target line and one for its context. Each encoder produces an embedding, and the two vectors are combined before passing through the segmentation and classification heads. This allows the model to use both local information (the line itself) and structural information from nearby lines when making predictions.

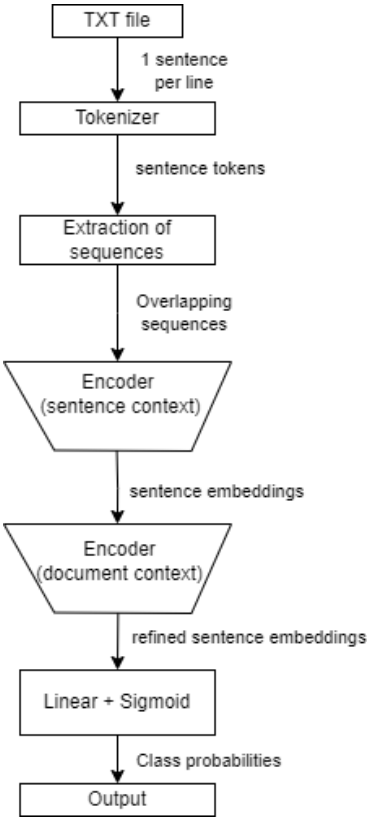


Figure 10 – Proposed bi-encoder model architecture

5.2.1.2 Implementation

The final chosen pre-trained transformer encoder is the variant "paraphrase-xlm-r-multilingual-v1" (REIMERS; GUREVYCH, 2019) of the XLM-RoBERTa base model (CONNEAU et al., 2020), which is pre-trained in 100 different languages (including English and French) to predict masked tokens and fine-tuned on a large-scale multilingual paraphrase identification task. This fine-tuning enables the model to learn semantically meaningful sentence embeddings that reflect sentence-level similarity. The model has around 278 million parameters, making it suitable for inference on small GPUs, and we can extract sentence embeddings via mean pooling over the token embeddings. We have also tested other popular embedding models on Hugging Face Hub such as "paraphrase-multilingual-

MiniLM-L12-v2", "multilingual-e5-large-instruct" and "bilingual-embedding-large", but the chosen model showed superior performance in our task.

The BiLSTM encoder is composed of 3 stacked BiLSTM layers with Layer Normalization to reduce internal covariate shift and stabilize training and dropout layers ($p = 0.2$) to avoid overfitting. The chosen model's hidden size is 768 (same as the transformer encoder) for the first layer and 2×768 for the subsequent layers since the first layer outputs bidirectional concatenated embeddings. In addition to that, we add a final fully connected layer with sigmoid activation function for classification. This encoder has around 38 million parameters, which makes it suitable for small GPUs. We've also experimented with BiGRU (CHUNG et al., 2014) and BERT (DEVLIN et al., 2019) architectures as document-level encoders. However, BiGRU performed worse than BiLSTM, and we could not fine-tune a BERT-based encoder using XLM-RoBERTa embeddings as input; thus, training the model from zero became inviable with our limited document corpus. Figure 10 shows the complete designed bi-encoder model.

5.2.1.3 Training and hyperparameters

We generate text sequences using a sliding window approach, where each line is treated as an individual unit and the surrounding lines are added as contextual input. Consecutive windows overlap by 50%, and we experimented with different window sizes to find a balance between providing enough structural information and keeping the computation efficient. Larger windows help the model capture broader contract structure, while smaller ones reduce processing time, making the window size an important hyperparameter in the overall system.

For training the bi-encoder model, we combine our gold training set split with the silver dataset into a final training set, and the final distribution can be seen in Figure 11. We use as objective the Binary Cross Entropy (BCE) loss with inverse class frequency (ICF) as weights, which is computed in the training set as shown in equation 5.1. The training strategy chosen was to freeze the weights of the first encoder and train only the second encoder, monitoring BCE loss and weighted F2-score in the gold validation set for each epoch, saving the model weights for the best F2-score. This saving strategy was used to optimize performance for the most prevalent classes (such as "Article") while minimizing false negatives.

We trained the model with AdamW optimizer in a single NVIDIA L4 GPU. Figures 12 and 13 show stable training for our best model.

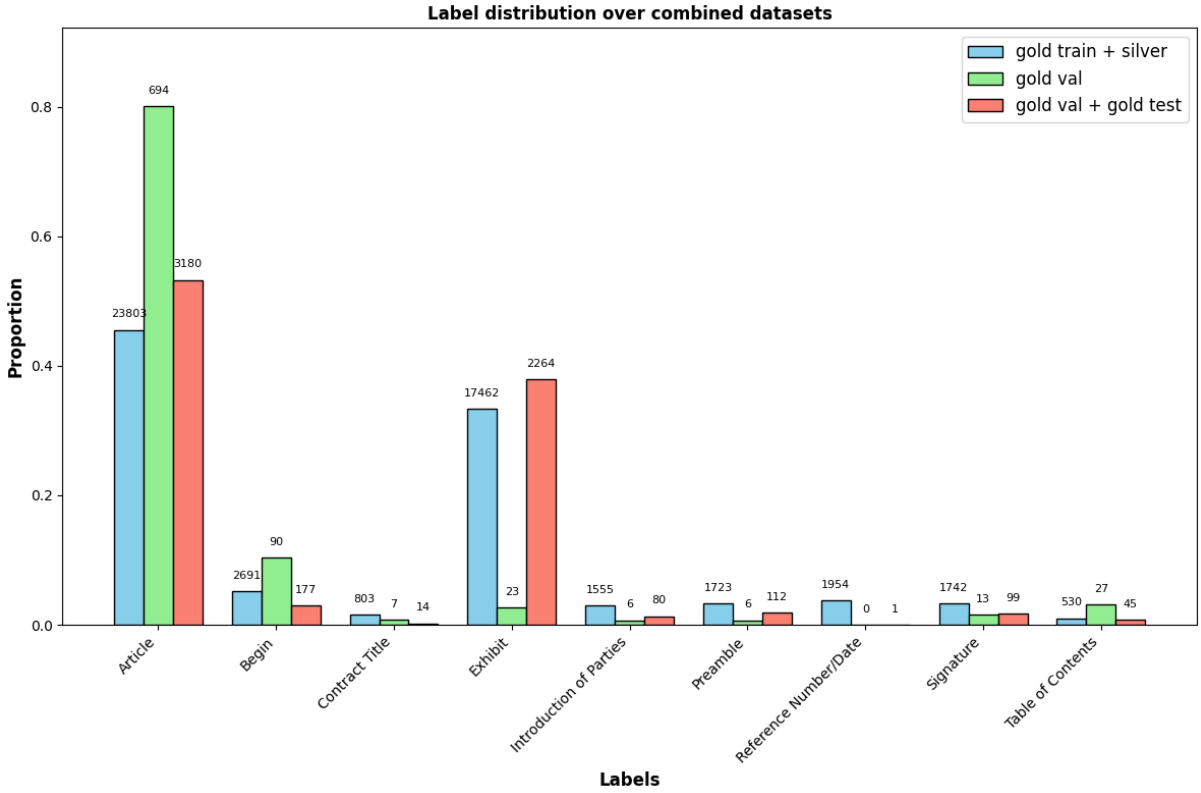


Figure 11 – Distribution of labels in combined datasets

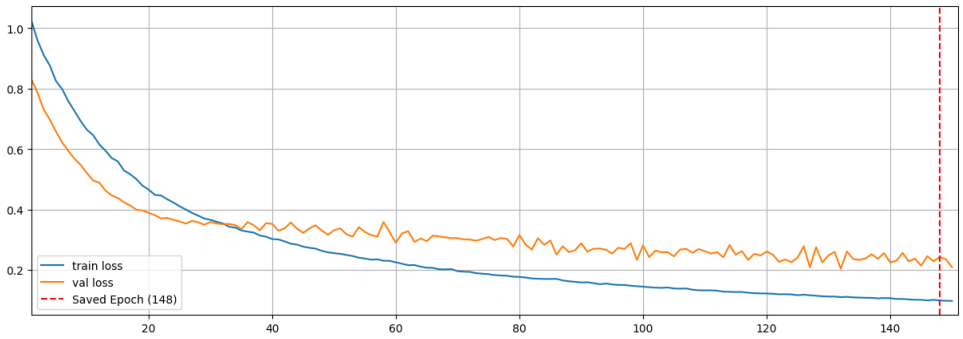


Figure 12 – Training and validation losses over epochs, window size = 128, batch size = 64, 150 epochs, learning rate = 10^{-5}, weight decay = 0.05

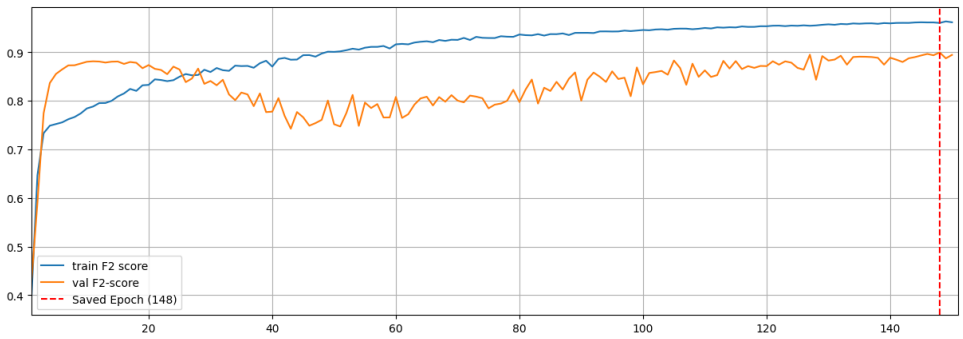


Figure 13 – Training and validation weighted F2-scores over epochs, window size = 128, batch size = 64, 150 epochs, learning rate = 10^{-5}, weight decay = 0.05

$$\text{ICF}_i = \frac{N - n_i}{n_i} \quad \text{where} \quad \begin{array}{l} N = \text{total number of samples,} \\ n_i = \text{number of samples where class } i \text{ is present.} \end{array} \tag{5.1}$$

To demonstrate the effectiveness of our data augmentation method, we also trained our classification bi-encoder using only the gold training set.

5.2.2 LLM-based data-augmentation

5.2.2.1 Architecture

The LLM-based augmentation pipeline takes unlabeled contracts and uses structured prompts to obtain segment boundaries and labels from an LLM. The outputs are cleaned and converted into the same line-level format as the gold dataset. These silver labels are then combined with gold data to increase the amount of training material and improve model robustness.

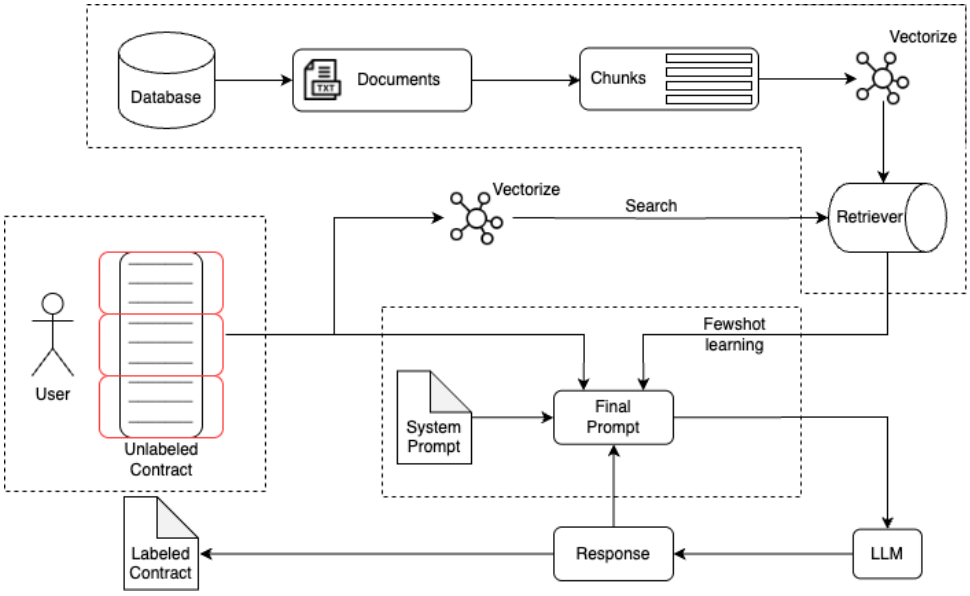


Figure 14 – LLM-based few-shot learning system design

5.2.2.2 Implementation

The LLM-based data-augmentation pipeline is implemented as a small set of Python modules, using the following steps and features:

a) **Labeled Contracts Chunking:** We split the gold training set of annotated contracts into smaller, overlapping chunks using a sliding-window approach over lines, and we attach the per-line labels to each chunk. Window length is constrained by a maximum number of lines so that retrieval can focus on local, representative contexts.

To reduce token waste, long lines that add little to local semantics are cropped since they are less critical for context.

b) **Vectorization:** Each labeled chunk is embedded with the "paraphrase-multilingual-MiniLM-L12-v2" embedding model, which generates one embedding per line; we apply mean pooling across lines to form a single vector that summarizes the chunk. The pooled embeddings, together with the raw text and labels, are stored as an external knowledge base for retrieval.

c) **Input chunking:** For every contract to be augmented, we construct non-overlapping chunks based on a strict token budget. Token counts are computed with the tiktoken library (OpenAI's tokenizer). Each input chunk is then passed through the same vectorization step from (b), producing a query vector. We keep an explicit mapping from chunk indices back to original line numbers so that any output can be realigned precisely.

d) **Retrieval:** For each input chunk, we compute cosine similarity between its vector and the knowledge-base vectors from (b), returning the top-k most similar labeled chunks as domain examples context to our LLM.

e) **LLM prompt:** The prompt is assembled from four parts:

- *System prompt*: A set of predefined roles and rules to better instruct model behaviour and the required output schema.
- *Few-shot learning context*: The top-k retrieved chunks, with raw text as input and labeled text as expected output.
- *Response*: Key information from the most recent sliding window request, ensuring the model retains context within the same document when we change windows.
- *Query*: The current input chunk to be labeled.

f) **API handling:** We then use the OpenAI Chat Completions API to label each window and run calls concurrently with a small Python thread pool because windows are independent and can be processed in parallel. This keeps the code simple while improving throughput: each thread builds its prompt, sends the request, and writes the response to disk.

g) **Output**: The final product of the pipeline is a set of *silver* annotations aligned at line level for each augmented contract.

5.2.2.3 Input and output formats

Input and output design matters as much as model choice in LLM workflows. The prompt and the expected response follow a pattern: the more rigorous and predictable

this pattern, the lower the variance and the lower the risk of hallucination. Two forces act in opposite directions: richer instructions and fewer context steps generally improve performance, but they also inflate token counts and, beyond a certain point, increase error rates. The general rule is to keep instructions concise and reserve tokens for the smallest scheme that still captures exactly what is needed.

In this project, various input and output formats were tested. XML and JSON are the most common choices. XML is human-readable and naturally expresses hierarchical structure with *<tags>*. . . *</tags>*, but sliding-window processing makes it hard to preserve that hierarchy: an opening tag may appear in one window and its closing tag in the next, creating ambiguity. Worse, tiny hallucinations, like a missing closing tag or an invented attribute, can invalidate the whole parsing tree. JSON is more compact, and straightforward to validate programmatically, which lowers token cost and reduces the chance of format drift, but the main challenge is preserving a one-to-one mapping between input lines and outputs: if the model echoes or reformats text freely, it can reorder, merge, or drop items.

The best-behaved variant follows a strict rule, using a line-indexed approach, and carrying the label(s) of this line inside a set. Each line is labelled using the following structure with $n = 1$ in the multiclass approach:

$$\{line_{id}\}\ \{label\}_n\ (Begin\ label)_n : line_{text}.$$

This structure represents the output and context format, while the input follows the same structure without $\{label\}_n$ and $(Begin\ label)_n$ elements. This way, we help the model to preserve a one-to-one mapping between input lines and outputs and can easily detect when a label starts by the Begin tag. An example is shown in Figure 15.

```
{1} {Contract Title} (Begin Contract Title) : AGENCY AGREEMENT
{2} {Reference Number/Date}{Introduction of Parties} (Begin Introduction of Parties)(Begin Reference Number/Date) : THIS AG
{3} {Introduction of Parties} : between the ADVISORS' INNER CIRCLE FUND II, a business trust existing under the
{4} {Introduction of Parties} : laws of the Commonwealth of Massachusetts, having its principal place of
{5} {Introduction of Parties} : business at Xxx Xxxxxxx Xxxxxx Xxxx, Xxxx, Xxxxxxxxxxx 00000 (the "Trust") on
{6} {Introduction of Parties} : behalf of each separate series of the Trust (each a "Fund") and each separate
{7} {Introduction of Parties} : series of certain Funds (each a "Portfolio"), and DST SYSTEMS, INC., a
{8} {Introduction of Parties} : corporation existing under the laws of the State of Delaware, having its
{9} {Introduction of Parties} : principal place of business at 000 Xxxx 00(xx) Xxxxxx, 0(xx) Xxxxx, Xxxxxx
{10} {Introduction of Parties} : Xxxx, Xxxxxxxx 00000 ("DST"):
{11} {Preamble} (Begin Preamble) : WITNESSETH:
{12} {Preamble} : WHEREAS, the Trust desires to appoint DST as Transfer Agent and
{13} {Preamble} : Dividend Disbursing Agent, and DST desires to accept such appointment;
{14} {Preamble} : NOW,THEREFORE, in consideration of the mutual covenants herein contained, the
{15} {Preamble} : parties hereto agree as follows:
{16} {Article} (Begin Article) : 1. DOCUMENTS TO BE FILED WITH APPOINTMENT.
{17} {Article} : In connection with the appointment of DST as Transfer Agent and Dividend
{18} {Article} : Disbursing
{19} {Article} : Agent for the Trust, there will be filed with DST the following documents:
```

Figure 15 – LLM-based model output and context format example for the multilabel framework

A complete example of a classification prompt is provided in appendix A, but we do not include the retrieved context due to its excessive size.

5.2.2.4 Hyperparameters

In terms of hyperparameters, the best results were found with:

- Overlap percent for labeled documents chunking: 70 %.
- Window size for labeled documents chunking: 200 sentences ≈ 10000 tokens.
- Maximum number of tokens for the input chunking: 6000 tokens.
- Number of retrieved chunks (k): 2.

5.2.2.5 Post-processing

Even with a carefully designed prompt and I/O formats, the pipeline relies on a robust post-processor to validate outputs. The response is first checked against a strict schema, then each predicted record is aligned to its input line using the line ID. To tolerate minor hallucinations, such as extra whitespace, order changing or slight rewrites, the validator applies conservative fuzzy matching, but it never overwrites the source text: the original input lines remain the source of truth. Labels are then extracted with regular expressions rules and only records that pass validation are serialized to JSON and merged into the training corpus as “silver” annotations, producing the augmented dataset used for the bi-encoder model training.

We observed that the model struggled to classify complete "Exhibit" segments, even though it correctly identified the start of each exhibit. Based on this observation, we implemented a heuristic: if exhibits extend to the end of the document, once an "Exhibit Begin" is detected, all subsequent lines are post-processed to receive the "Exhibit" label.

5.3 Chapter Considerations

This chapter described the development of the system, explaining how the components were implemented and integrated into a complete workflow. The chapter shows how the design decisions were driven by the requirements defined earlier. This leads into Chapter 6, where the system is evaluated and the impact of the methodological and implementation choices is reflected in the experimental results.

6 Results

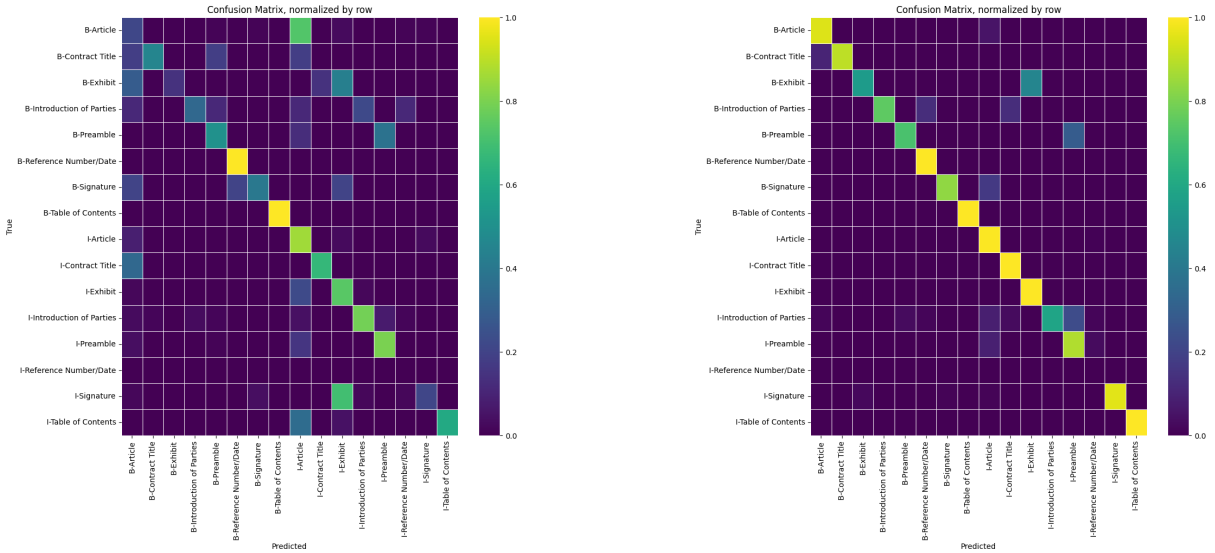
In Table 1, we can observe the performance of our LLM-based classification pipeline with the OpenAI models GPT-4o and GPT-4o-mini, computed over our combined testing set, which is composed of gold test and validation documents. We can see that GPT-4o performs better than GPT-4o-mini for all tasks with a considerable margin, especially in the segmentation metrics. We observe that the model performs well for both multilabel and multiclass classification, however, it's important to note that the multilabel performance depends significantly on the post-processing steps applied specifically to the "Exhibit" label.

We notice a considerable difference between macro and weighted metrics, which indicates that the LLM performs better for the most populated classes, and we confirm this by looking at performance for different classes in Figures 17 and 18. This performance gap is expected because the LLM classification strongly depends on context, which is very limited for less populated classes such as "Reference Number/Date".

For the multiclass results, we can see in figure 16 a higher tendency of GPT-4o-mini to bias the results toward the "I-Article" label, which may be explained by its stronger dependence on the context than GPT-4o and the fact that our gold training set naturally contains more "I-Article" labels. Finally, in figure 16, we can see that GPT-4o frequently confuses inside "I-" labels with begin "B-" labels of the same class.

Metric	**Multiclass**		**Multilabel**	
	4o	**4o-mini**	**4o**	**4o-mini**
Classification				
Weighted-F2	**0.9789**	0.8187	**0.9838**	0.8534
Macro-F2	**0.7898**	0.5086	**0.8221**	0.6998
Segmentation				
Weighted-F2	**0.8818**	0.2839	**0.8542**	0.5447
Macro-F2	**0.8721**	0.3401	**0.7523**	0.5478
*Weighted Pk	**0.0637**	0.3673	**0.0423**	0.2363
*Weighted Windowdiff	**0.0786**	0.4028	**0.0549**	0.2948

Table 1 – Results for the LLM-based classification, evaluated on both multiclass and multilabel classification with the combined testing set (gold testing and gold validation). *Smaller Pk and WindowDiff scores represent better results



(a) GPT-4o-mini (b) GPT-4o

Figure 16 – Confusion matrix for LLM-based multiclass classification, computed with the combined testing set (gold testing + gold validation)

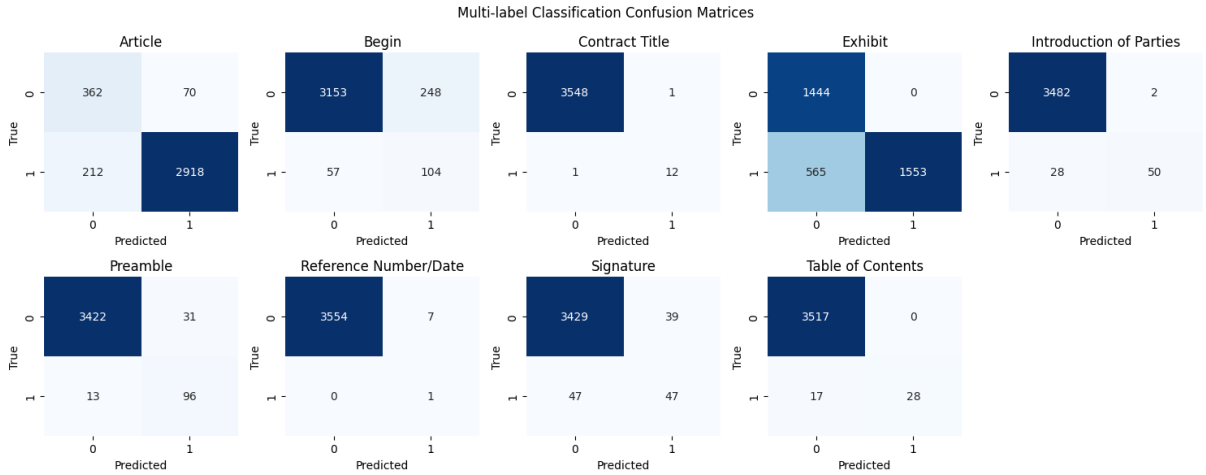


Figure 17 – Binary confusion matrices computed with the combined testing set (gold testing + gold validation) for GPT-4o-mini classifications

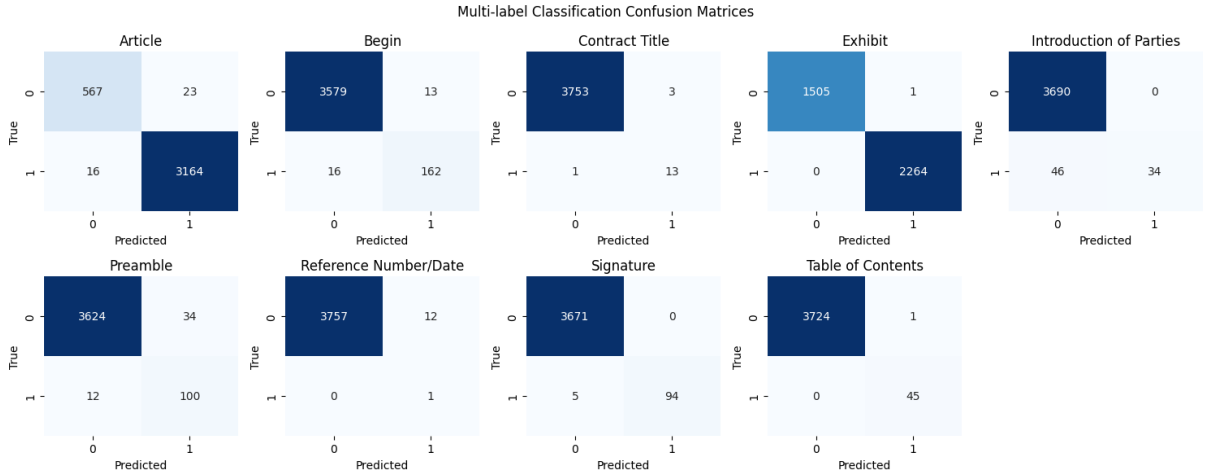


Figure 18 – Binary confusion matrices computed with the combined testing set (gold testing + gold validation) for GPT-4o classifications

In Table 2 we compare the classification and segmentation results using different window sizes in the bi-encoder pipeline. It's possible to see that we achieved the best segmentation performance with 128 sentences per window and the best classification performance with 64 sentences per window. Since our main goal is segmentation, we choose as our best bi-encoder model the one with the top segmentation F2-score.

Window size	**Classification weighted-F2**	**Segmentation weighted-F2**
64	**0.9545**	0.5034
128	0.9333	**0.5140**
256	0.9553	0.4877

Table 2 – Bi-encoder weighted F2-scores for Classification and Segmentation at different window sizes, evaluated on the combined testing set

In Table 3 we can see the classification and segmentation results for the best bi-encoder model and in Figure 19 we can see the binary confusion matrices for each class. The improved results across all metrics with the augmented training set suggest the effectiveness of the LLM-based data augmentation method. The weighted metrics are better than macro metrics which indicates better performance for the most populated classes, and we confirm this when comparing the confusion matrices for "Article" and "Introduction of Parties", for example. We observed that the "Exhibit" class is as prevalent in the testing set as the "Article" class, yet its general performance is worse. This may be due to the inclusion of a large, complete "Exhibit" document in the gold testing set. In general, the classification metrics are higher than the segmentation versions which is expected since the classification is measured across the entire segments whereas segmentation considers mainly the segment boundaries. Finally, we observe a substantial number of false positives for the "Begin" label, indicating that the model tends to produce smaller chunks. However, this behavior is anticipated, as our training approach prioritizes higher recall.

Metric	Evaluation method		Training set
	test	test + val	
Classification			
Weighted-F2	0.8896	0.8662	gold only
Weighted-F2	**0.9541**	**0.9333**	augmented
Macro-F2	0.4192	0.4440	gold only
Macro-F2	**0.7116**	**0.7287**	augmented
Segmentation			
Weighted-F2	0.1311	0.1557	gold only
Weighted-F2	**0.4244**	**0.5140**	augmented
Macro-F2	0.1124	0.1124	gold only
Macro-F2	**0.3390**	**0.3787**	augmented
*Weighted Pk	0.2882	0.2668	gold only
*Weighted Pk	**0.1871**	**0.1909**	augmented
*Weighted Windowdiff	0.3591	0.3239	gold only
*Weighted Windowdiff	**0.3150**	**0.3009**	augmented

Table 3 – Results for the bi-encoder model with gold training only vs. augmented set. Best results in both the test and test + val columns are highlighted in bold. *Smaller Pk and WindowDiff scores represent better results

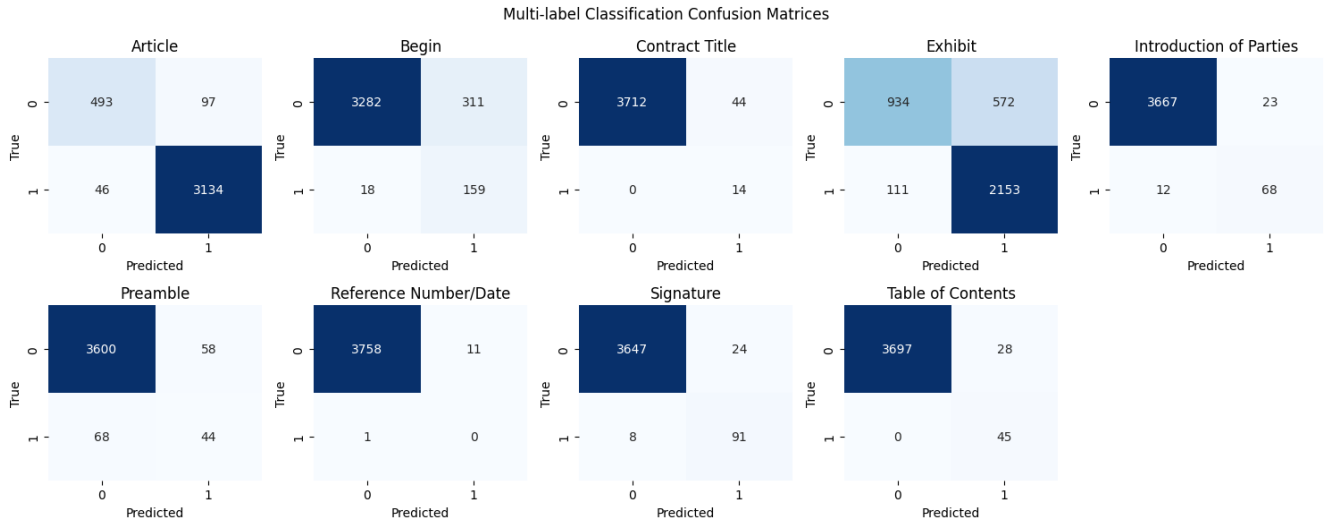


Figure 19 – Binary confusion matrices computed with the combined testing set (gold testing + gold validation) for the bi-encoder model classifications

7 Conclusion and Future Work

In this work, we developed an automatic segmentation model for legal documents and introduced an LLM-based classification pipeline to generate additional labeled data, which is intended to improve the training of a smaller encoder model.

7.1 Main Contributions

Our experiments show that this pipeline reliably produces high-quality silver annotations, as indicated by higher weighed and macro F2-scores, and that its robust outputs, when merged with the gold set, strengthen the training signal of the bi-encoder model and translate into consistent downstream gains (higher weighted and macro F2-scores, improved boundary metrics) relative to gold-only training. These results support LLMs as an effective label generator whose outputs can be distilled into a smaller, deployable model, without relying on the LLM at inference time.

However, expanding the size of the unlabeled dataset will likely be necessary to enhance the bi-encoder model's generalization capabilities, especially for underrepresented classes. Additionally, our observations regarding the "Exhibit" label suggest that using a more focused label, such as "Exhibit Title", might yield more consistent segmentation boundaries, especially given that exhibits often span entire segments of the contract. A similar solution can potentially be applied to the "Section" label, which was ultimately removed from our final label set because of the confusion with "Article" label.

Beyond the numerical results, the project also contributes a practical, end-to-end pipeline for legal document segmentation. Starting from raw contracts, the system normalizes text, segments it into lines, applies the bi-encoder for segmentation and classification, and then refines outputs using simple, configurable heuristics. This design shows that it is possible to combine modern encoder architectures, LLM-assisted labeling, and lightweight post-processing into a workflow that remains compatible with realistic constraints such as single-GPU execution and sub–10-second latency for typical contracts.

Another contribution lies in the systematic analysis of the label scheme itself. By comparing different label granularities and monitoring how errors cluster around specific roles (such as "Exhibit" and "Section"), the work highlights how the choice of labels directly impacts boundary quality and metric stability.

7.2 Future Work

Future work may concentrate on expanding the unlabeled dataset to improve distillation and also improving the labeling scheme to enhance performance on the critical mentioned classes. Two simple schema adjustments already suggested by our analysis, splitting Exhibit into Exhibit Title and clarifying the Section/Article boundary, would reduce ambiguity. We also expect that fine-tuning a small, open-weight LLM on gold plus high-precision silver would further raise silver quality.

A natural continuation is to refine the process by which silver data is selected and incorporated. Instead of treating all LLM-generated annotations equally, future work could explore confidence-aware filtering, disagreement-based selection (for instance, focusing human review on lines where the bi-encoder and the LLM disagree), or iterative re-labeling rounds where the model's own predictions are fed back into the prompt. Such strategies would help allocate human effort where it brings the most benefit and might reduce the noise introduced by LLM outputs on borderline cases.

A major avenue is chunking and retrieval. Instead of fixed and token-count windows, a tag-aware approach could be explored, with variable-size windows that expand around headings, enumerations, and other structural anchors, such as matching definition blocks. Retrieval itself should go beyond cosine similarity, a combination of dense and lexical evidence, such as cosine and BM25, with different weight, could improve a lot the retrieval quality and the results, as we noticed a substantial improvement with the use of Retrieval Augmented Generation (RAG) in our LLM pipeline.

Finally, from a systems perspective, there is room to investigate further optimizations and deployment aspects. Techniques such as model pruning, quantization, or distilled variants of the encoder could be evaluated to reduce latency and memory usage while preserving segmentation quality. In parallel, integrating the model into an interactive review tool, where users can visualize segments, correct labels, and feed edits back into the training loop, would close the gap between experimental results and real-world adoption. Such integration would also open the door to continuous learning, where the system gradually adapts to new document types and organizational practices over time.

Bibliography

BROWN, T. B. et al. *Language Models are Few-Shot Learners*. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Citado na página 23.

CHUNG, J. et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. Disponível em: <https://arxiv.org/abs/1412.3555>. Citado na página 39.

CONNEAU, A. et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. Disponível em: <https://arxiv.org/abs/1911.02116>. Citado na página 38.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Disponível em: <https://arxiv.org/abs/1810.04805>. Citado 2 vezes nas páginas 23 and 39.

II, M. J. B.; KATZ, D. M.; DETTERMAN, E. M. *LexNLP: Natural language processing and information extraction for legal and regulatory texts*. 2018. Disponível em: <https://arxiv.org/abs/1806.03688>. Citado na página 22.

LAN, M. et al. Multi-label sequential sentence classification via large language model. In: AL-ONAIZAN, Y.; BANSAL, M.; CHEN, Y.-N. (Ed.). *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, 2024. p. 16086–16104. Disponível em: <https://aclanthology.org/2024.findings-emnlp.944/>. Citado 2 vezes nas páginas 23 and 24.

LEWIS, P. et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. Disponível em: <https://arxiv.org/abs/2005.11401>. Citado na página 23.

MALIK, V. et al. *Semantic Segmentation of Legal Documents via Rhetorical Roles*. 2022. Disponível em: <https://arxiv.org/abs/2112.01836>. Citado na página 23.

MOGHADASI, M. N.; ZHUANG, Y. Sent2vec: A new sentence embedding representation with sentimental semantic. In: *2020 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2020. p. 4672–4680. Citado na página 23.

RAMSHAW, L. A.; MARCUS, M. P. *Text Chunking using Transformation-Based Learning*. 1995. Disponível em: <https://arxiv.org/abs/cmp-lg/9505040>. Citado 2 vezes nas páginas 21 and 29.

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*. Association for Computational Linguistics, 2019. Disponível em: <http://arxiv.org/abs/1908.10084>. Citado na página 38.

A PROMPT EXAMPLE

System Prompt

You are a legal text classifier. Your task is to:
- Read each line of the provided text.
- Assign one or more of the following labels to each line: [Contract Title, Article, Introduction of Parties, Table of Contents, Preamble, Signature, Exhibit, Reference Number/Date].
- For every label that applies to a line, include it in the output.
- If a line marks the start of a 'new top-level section' for any label (even if nested within another label), prepend that line with '(Begin label)'.
- A 'new top-level section' refers to the change of a label from the previous the line
- Do not add 'Begin Article' for sub-articles (e.g., 1.1, 1.2, etc.). Treat these lines as continuations of the same Article.
- Continue applying all the 'top-level sections' labels to subsequent lines.
- Output in the specified format.

K-pairs of Input and Expected output (Few-shot learning), $K = 2$

Query Prompt

Now classify the following lines from the contract.
Remember to follow the exact format:
{line_number} {label1}{label2}{ (Begin label) if new}: {original_text}
Make sure to include all applicable labels for nested structures.
Note that the last Begin Articles you classified were at:
{81} {Article} (Begin Article): 3. INTERVENTION PROCEDURES
{96} {Article} (Begin Article): 4. VEHICLE ASSISTANCE SERVICES
So, only start new Article if it has the same marker level.
Also, the last line you classified was:
{154} {Article}: In France , if the Vehicle has been towed under the conditions of paragraph
Thus, you might keep this label (do not Begin it again).
{155} : Breakdown/towing and has been immobilized for more than 24 hours as a result of:
{156} : - of an Accident,
{157} : - of a Fire,
{158} : - of a Breakdown,
{159} : - of an Attempted Theft,
{160} : - of the Theft of the Vehicle,
{161} : We make available to you, or to a person of your choice residing in France, a 1st

class train ticket or economy class plane ticket to go and collect your repaired Vehicle.
{162} : This service cannot be combined with the services:
{163} : - Waiting for repair
{164} : Abroad, if the Vehicle has been towed under the conditions of paragraph
{165} : Breakdown/towing and has been immobilized for more than 72 hours as a result of:
{166} : - of an Accident,
{167} : - of a Fire,
{168} : - of a Breakdown,
{169} : - of an Attempted Theft,
{170} : - of the Theft of the Vehicle
{171} : We make available to you, or to a person of your choice residing in France, a 1st class train ticket or economy class plane ticket to go and collect your repaired Vehicle.
{172} : This service cannot be combined with the services:
{173} : - Waiting for repair
{174} : - Repatriation of the Vehicle (from Abroad only)
{175} : 4.5. STORAGE COSTS (France and Abroad)
{176} : Storage costs following a breakdown/towing.
{177} : During a trip in France or Abroad, if your Vehicle, towed under the conditions of paragraph 4.1 Breakdown/Towing , is immobilized in a garage as a result of:
{178} : - of an Accident,
{179} : - of a Fire,
{180} : - of a Breakdown,
{181} : - of an Attempted Theft,
{182} : - of the Theft of the Vehicle, if the Vehicle is found damaged within a period of 6 months from the declaration of Theft to the authorities.
{183} : We cover the storage costs of the Vehicle for up to a maximum of 48H and up to a limit of 55 TTC maximum.
{184} : 4.6. COSTS OF ABANDONING THE VEHICLE (Abroad only)
{185} : Abroad, if the market value or the experts valuation of the Vehicle before the Accident, Fire, Breakdown, Attempted Theft, or Theft of the Vehicle which caused the immobilization, is lower than the cost of repairs, We may, at your express request, arrange for your Vehicle to be abandoned on site. In this case, the abandonment costs are payable by you. You must then provide Us, within 1 month at the latest from the date of your return to France, with the documents required for the abandonment, as requested by the customs authorities of the country concerned. Failing this, You will be responsible for the abandonment of the Vehicle on site.
{186} : 4.7. STORAGE COSTS FOLLOWING THE ABANDONMENT OF THE VEHICLE (Abroad only)
{187} : Your Vehicle will be abandoned under the service costs of abandoning the Vehicle

, We cover the storage costs up to a limit of 115 TTC starting from the receipt of the documents necessary for the legal abandonment of the Vehicle.
{188} : 5. ASSISTANCE SERVICES FOR PERSONS
{189} : 5.1. SOME ADVICE FOR YOUR TRIP
{190} : BEFORE DEPARTURE
{191} : • Check that your contract covers You for the country concerned and for the duration of your trip.
{192} : • Remember to take with You forms appropriate to the duration and nature of your trip as well as to the country in which You are travelling (there is specific legislation for the European Economic Area). These various forms are issued by the Primary Health Insurance Fund to which You are affiliated so that You may, where applicable, in case of illness or accident, benefit from direct coverage of your medical expenses by this body.
{193} : • If You travel to a country that is not part of the European Union and the European Economic Area (EEA), You must inquire, before your departure, to check whether this country has concluded a social security agreement with France. To do so, You must consult your Health Insurance Fund to find out whether You fall within the scope of the said agreement and whether You have any formalities to complete (obtaining a form, etc.).
{194} : To obtain these documents, You must contact, before your departure, the competent institution and, in France, your Health Insurance Fund.
{195} : • If You are undergoing treatment, do not forget to take your medication and carry it in your hand luggage to avoid an interruption of treatment in case of delay or loss of luggage; indeed, some countries (United States, Israel, etc.) do not allow shipments of this type of product.
{196} : ON SITE
{197} : • If You engage in a risky physical or motor activity or travel in an isolated area as part of your trip, We advise You beforehand to make sure that an emergency rescue system has been put in place by the competent authorities of the country concerned to respond to any possible request for assistance.
{198} : • In case of loss or theft of your keys, it may be important to know their numbers. Take the precaution of noting these references.
{199} : • Likewise, in case of loss or theft of your identity papers or your means of payment, it is easier to replace these documents if You have taken the trouble to make photocopies and to note the numbers of your passport, identity card and bank card, which You will keep separately.
{200} : • On entering certain countries, the characteristics of the vehicle are recorded on your passport or on an official document; if Xxxx leave the country by
{201} : leaving your vehicle, you must complete certain formalities with the customs authorities (passport to be cleared, temporary import, etc.).

{202} : • If You are ill or injured, contact us as soon as possible, after having called the emergency services (SAMU, fire brigade, etc.) which We cannot replace.
{203} : • In case of breakdown or accident on a motorway or expressway, use the nearest emergency telephone. You will be directly connected to a person authorized to call the emergency services.
{204} : WARNING
{205} : Certain medical conditions may constitute a limit to the application conditions of the contract. We advise You to read this assistance agreement carefully.
{206} : 5.2. TRANSPORT/REPATRIATION
{207} : Following an Injury resulting from a road accident, or an Illness, in France or Abroad, our doctors contact the local doctor who has treated You following the event.
{208} : The information gathered from the local doctor, and possibly from your usual general practitioner, allows Us, after a decision by our doctors, to initiate and organize, according to purely medical requirements:
{209} : - either your return to your Home,
{210} : - or your transport, if necessary under medical supervision, to an appropriate hospital service close to your Home,
{211} : by light medical vehicle, ambulance, sleeper car, 1st class train (berth or seat), economy class plane or air ambulance.
{212} : Similarly, depending solely on medical requirements and on the decision of our doctors, We may in some cases arrange and organize an initial transport to a nearby care centre, before envisaging a return to a facility close to your Home.
{213} : Only your medical condition and compliance with the health regulations in force are taken into account in deciding on the transport, the choice of means used for this transport and the choice of any hospitalization facility.
{214} : IMPORTANT
{215} : In this respect, it is expressly agreed that the final decision to be implemented rests ultimately with our doctors, in order to avoid any conflict of medical authority.
{216} : Furthermore, if Xxxx refuse to follow the decision considered the most appropriate by our doctors, your refusal releases us from all liability, notably in case of return by your own means, or in case of deterioration of your state of health.
{217} : 5.3. ADVANCE ON HOSPITALIZATION COSTS (Abroad only)
{218} : Following an Illness or an Injury resulting from a road accident occurring on board the covered Vehicle, during a trip Abroad and while You are hospitalized, We may advance hospitalization costs up to 7 620 TTC per Beneficiary and per year. This advance is subject to the following cumulative conditions: for care prescribed in agreement with our doctors, as long as they consider You non-transportable after obtaining information from the local doctor.
{219} : No advance is granted from the day on which We are able to carry out the transport,

even if Xxxx decide to remain on site.
{220} : In all cases, You undertake to reimburse Us this advance no later than 30 days after receipt of our invoice.
{221} : For You yourself to be reimbursed, You must then carry out the necessary procedures to recover your medical expenses from the relevant bodies.
{222} : 5.4. PRESENCE OF A RELATIVE
{223} : When You are hospitalized at the place of your Illness or your Injury
{224} : following a road Accident occurring on board the covered Vehicle and when
{225} : our doctors consider, based on the information provided by the local doctors, that your return cannot take place before 10 days (for a child under 16, the period is reduced to 48 hours), We arrange and cover the round trip from France by 1st class train or economy class plane for a person of your choice so that they may come to your bedside.
{226} : We also cover this persons hotel costs (room and breakfast) for a maximum of 10 nights, up to a maximum of 46 TTC per night.
{227} : 5.5. EARLY RETURN FOLLOWING A DEATH
{228} : During your trip, You learn of the death, occurring in France, during your trip on board the covered Vehicle:
{229} : - of a Member of your family,
{230} : So that Xxxx can attend the funeral of the deceased in France,
{231} : We arrange and cover your one-way trip by 1st class train or economy class plane to France within the 8 days following the date of death.
{232} : If supporting documents (death certificate, proof of family relationship) are not submitted within 30 days, We reserve the right to charge You the full cost of the service.
{233} : This service is granted provided that the date of the funeral is earlier than the date initially planned for your return.
{234} : 5.6. TRANSPORT OF THE BODY IN CASE OF DEATH OF A BENEFICIARY
{235} : In case of the death of a Beneficiary during their trip following a road Accident occurring on board the covered Vehicle.
{236} : We arrange and cover the transport of the deceased Beneficiary to the place of the funeral in France.
{237} : We also cover all costs required for the preparation and the specific arrangements for the transport only.
{238} : 5.7. COFFIN COSTS IN CASE OF DEATH OF A BENEFICIARY
{239} : In case of the death of a Beneficiary following an Accident occurring on board the covered Vehicle, We contribute to coffin or urn costs, which the family obtains from the funeral provider of its choice, up to a maximum of 762 TTC. The other costs (notably ceremony, local transport, burial) remain payable by the family.
{240} : 5.8. TRANSMISSION OF URGENT MESSAGES
{241} : During your trip, if You are unable to contact a person who is in France, We trans-

mit, at the time and on the day You have chosen, the message that You have previously communicated to Us by telephone. NOTE:
{242} : This service does not allow the use of collect calls. The content of your messages cannot, moreover, in any case incur our responsibility, and remains subject to French legislation, in particular criminal and administrative law. Failure to comply with this legislation may lead to refusal to transmit the message.
{243} : 5.9. REPLACEMENT DRIVER
{244} : Following:
{245} : - an Injury resulting from a road Accident occurring on board the covered Vehicle, during your trip, if your medical condition no longer allows You to drive your private Vehicle and none of the passengers can replace You, We provide:
{246} : - either a driver to drive the Vehicle to your Home, by the most direct route. We cover the drivers travel expenses and salary. The latter operates in accordance with the regulations in force applicable to his or her profession. This cover is granted to You if your Vehicle is duly insured, in perfect working order, compliant with the standards of the
{247} : national and international Highway Code and meets the compulsory roadworthiness test standards. Otherwise, We reserve the right not to send a driver,
{248} : - or a 1st class train ticket or economy class plane ticket, so that You or a person of your choice can bring the Vehicle back.
{249} : Road expenses (fuel, any tolls, ferry crossings, hotel and restaurant costs for any passengers) remain payable by you.
{250} : 5.10. ADVANCE OF CRIMINAL BAIL
{251} : You are travelling Abroad and are subject to legal proceedings as a result of a traffic accident and this to the exclusion of any other cause. We advance the criminal bail up to a maximum of 11 450 TTC subject to the prior submission of an indictment andor any document issued by the local judicial authorities proving that legal proceedings have been initiated against you.
{252} : You undertake to reimburse Us this advance within a period of 30 days after receipt of our invoice or as soon as the criminal bail has been returned to you by the authorities if the return occurs before the expiry of this period.
{253} : 5.11. COVER OF LAWYERS FEES
{254} : You are travelling Abroad and are subject to legal proceedings as a result of a traffic accident and this to the exclusion of any other cause, We cover the lawyers fees that You have had to incur on site up to a maximum of 1525 TTC, provided that the alleged facts are not, under the countrys legislation, punishable by criminal sanctions.
{255} : Your request for coverage must necessarily be accompanied by the final court decision that has become enforceable.
{256} : This service does not cover legal proceedings brought in France, following a road Accident occurring Abroad.

{257} : 6. EXCLUSIONS
{258} : 6.1. EXCLUSIONS COMMON TO ALL SERVICES
{259} : Excluded are claims arising:
{260} : - from civil or foreign war, riots, popular movements, acts of terrorism, natural disasters,
{261} : - from your voluntary participation in riots or strikes, brawls or acts of violence,
{262} : - from the disintegration of the atomic nucleus or any irradiation from an energy source of a radioactive nature,
{263} : - from the use of medication, drugs, narcotics and similar products not medically prescribed, and from the abusive use of alcohol,
{264} : - from an intentional act on your part or a fraudulent act, an attempted suicide or suicide,
{265} : - from an incident occurring during motor trials, races or competitions (or their trials), which under current regulations require prior authorization from the public authorities, when You take part as a competitor, or during trials on circuits requiring prior approval from the public authorities, even if You use your own vehicle.
{266} : - from a loss occurring in one of the countries excluded from the guarantee of the assistance agreement or outside the guarantee validity dates, in particular beyond the planned duration of the trip Abroad.
{267} : Also excluded are:
{268} : - requests falling within the competence of local emergency services or primary transport such as SAMU, the fire brigade, and the related costs,
{269} : - costs incurred without our agreement, or not expressly provided for by this assistance agreement,
{270} : - costs not supported by original documents,
{271} : - non-waivable excess costs in case of vehicle rental,
{272} : - fuel and toll costs,
{273} : - customs duties,
{274} : - meal costs.
{275} : 6.2. EXCLUSIONS SPECIFIC TO ASSISTANCE FOR PERSONS
{276} : We cannot under any circumstances replace local emergency services. In addition to the common Exclusions to all services listed in section 6.1.1, the following are excluded:
{277} : - consequences of situations involving infectious risks in an epidemic context, exposure to infectious biological agents, exposure to chemical agents such as warfare gases, exposure to incapacitating agents, exposure to neurotoxic agents or agents with persistent neurotoxic effects, which are subject to quarantine or specific preventive or monitoring measures by international andor local health authorities of the country where You are staying andor national authorities of your country of domicile,
{278} : - pre-existing Illnesses andor Injuries diagnosed andor treated that have required

continuous hospitalization, day hospitalization or outpatient hospitalization in the 6 months preceding any request, whether relating to the manifestation or worsening of said condition,
{279} : - trips undertaken for diagnostic andor medical treatment purposes or for cosmetic surgery, their consequences and the costs resulting therefrom,
{280} : - organization and coverage of the transport referred to in section
{281} : Transport Repatriation for minor conditions that can be treated on site and that do not prevent You from continuing your trip or your stay,
{282} : - assistance requests relating to medically assisted reproduction and its consequences or to voluntary termination of pregnancy and its consequences,
{283} : - requests relating to reproduction or gestation for others (surrogacy), and its consequences,
{284} : - medical equipment and prostheses (dental, hearing, medical),
{285} : - non-urgent dental care, its consequences and the costs resulting therefrom
{286} : - spa treatments and the costs resulting therefrom,
{287} : - medical expenses incurred in your country of Home,
{288} : - planned hospitalizations, their consequences and the costs resulting therefrom,
{289} : - optical expenses (glasses and contact lenses for example),
{290} : - vaccines and vaccination costs,
{291} : - medical check-up visits and the costs related thereto, and their consequences,
{292} : - cosmetic procedures, the costs resulting therefrom as well as their consequences,
{293} : - stays in a convalescent home and the costs resulting therefrom,
{294} : - rehabilitation, physiotherapies, chiropracties, osteopathies, the costs resulting therefrom, and their consequences,
{295} : - medical or paramedical services and the purchase of products whose therapeutic nature is not recognized by French legislation, and the costs related thereto,
{296} : - health check-ups for preventive screening, regular treatments or tests, and the costs related thereto,
{297} : - search and rescue of a person in the mountains, at sea or in the desert, and the costs related thereto,
{298} : - costs related to excess baggage weight during transport by plane and the costs of forwarding baggage when it cannot be transported with you,
{299} : - travel cancellation costs,
{300} : - off-piste ski rescue costs.
{301} : 6.3. EXCLUSIONS SPECIFIC TO ASSISTANCE FOR VEHICLES
{302} : In addition to the common Exclusions to all services listed in section 6.1.1, the following are excluded:
{303} : - consequences of immobilizing the Vehicle to carry out maintenance operations,
{304} : - immobilization of the Vehicle resulting from scheduled interventions (maintenance,

inspection, servicing) or resulting from a lack of maintenance, as well as their consequences